**Федеральное государственное казенное военное
образовательное учреждение высшего образования
«Академия Федеральной службы охраны Российской Федерации»**

На правах рукописи

МИХАЛЕВ Павел Андреевич

УПРАВЛЕНИЕ ДЕЦЕНТРАЛИЗОВАННЫМ ОБМЕНОМ ИНФОРМАЦИЕЙ ДЛЯ ПРИНЯТИЯ РЕШЕНИЙ В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ МНОГОЭЛЕМЕНТНОЙ КЛАССИФИКАЦИИ С НЕПОЛНЫМИ ДАННЫМИ

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

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель:
к.т.н.
Куцакин Максим Алексеевич

Москва – 2026

Оглавление

ВВЕДЕНИЕ 6

Глава 1. Разработка модели классификатора распределенной системы многоэлементной классификации, учитывающей неполноту классов подмножества ее локальных классификаторов 13

1.1 Анализ предметной области распределенных систем с федеративным машинным обучением 13

1.2. Классификация систем федеративного машинного обучения 19

1.3. Исследование проблем неоднородности наборов данных в системах федеративного машинного обучения 22

1.3.1. Схемы неоднородного разделения наборов данных в системах с федеративным машинным обучением 23

1.3.2. Исследование подходов к обеспечению конфиденциальности наборов данных и параметров моделей в системах с федеративным машинным обучением 24

1.3.3. Исследование основ дифференцированной конфиденциальности и подходов к ее реализации 26

1.4. Исследование подходов к решению задачи многоэлементной классификации в системах с федеративным машинным обучением 30

1.4.1. Исследование способов бинаризации в задачах многоэлементной классификации 34

1.4.1.1. Исследование стратегии бинаризации OVO 35

1.4.1.2. Исследование стратегии бинаризации OVA 36

1.4.1.3. Исследование стратегии бинаризации AVA 37

1.5. Проблемы не наблюдаемости и неполноты классов в системах многоэлементной классификации с ФМО и подходы к их решению 38

1.6. Подходы к оцениванию производительности многоэлементной классификации в системах с федеративным машинным 41

1.7. Постановка задачи исследования 44

1.8 Выбор и обоснование функции потерь модели локальных классификаторов 45

1.8.1 Обоснование выбора категориальной кросс-энтропийной функции потерь модели локальных классификаторов 47

1.9 Моделирование многоэлементного классификатора системы с ФМО в условиях полноты классов 51

1.10 Моделирование многоэлементного классификатора для условий неполноты классов 52

1.11. Выбор метода классификации для многоэлементного классификатора в условиях неполноты классов локальных классификаторов 54

1.12. Выводы по главе..... 57

2.1 Исследование особенностей многоэлементного классификатора на основе модели гауссовой смеси распределений..... 59

2.2. Конкретизация модели GMM-классификатора для решения задачи многоэлементной классификации изображений..... 60

2.3 Исследование особенностей методов нахождения оценок максимального правдоподобия параметров модели гауссовой смеси распределений..... 64

2.3.1 Оценка параметров максимального правдоподобия 64

2.3.2 Алгоритм «Ожидание-максимизация» - EM-алгоритм..... 66

2.3.3 Метод оценивания апостериорного максимума (MAP - Maximum A posteriori Probability)..... 68

2.4 Разработка алгоритма получения значений оценок вероятностной функции ненаблюдаемых классов для многоэлементного классификатора,

функционирующего в условиях неполноты классов локальных классификаторов..... 70

2.4 Выводы по главе..... 73

Глава 3. Разработка алгоритма децентрализованного управления обменом данными системы многоэлементной классификации, функционирующей в условиях неполноты классов локальных классификаторов..... 74

3.1 Исследование парадигм обмена данными в системах с федеративным машинным обучением..... 74

3.2. Формальное представление централизованной схемы федеративного машинного обучения..... 78

3.3 Исследование и выбор метода агрегирования, применяемого в задачах бинаризации задач многоэлементной классификации и модельно-независимого машинного обучения..... 82

3.4 Разработка функциональной схемы децентрализованного управления обменом данными классификатора системы многоэлементной классификации 86

3.5 Разработка этапов алгоритма децентрализованного обмена данными классификатора системы многоэлементной классификации..... 88

3.6 Выводы по главе..... 92

ГЛАВА 4. РАЗРАБОТКА АРХИТЕКТУРЫ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ МНОГОЭЛЕМЕНТНОЙ КЛАССИФИКАЦИИ, ФУНКЦИОНИРУЮЩЕЙ В УСЛОВИЯХ НЕПОЛНОТЫ КЛАССОВ ЛОКАЛЬНЫХ КЛАССИФИКАТОРОВ93

4.1 Разработка структуры программного комплекса системы многоэлементной классификации, функционирующей в условиях неполноты классов локальных классификаторов..... 93

4.1.1 Выбор и обоснование фреймворка федеративного машинного обучения…..……93

4.1.1.2 TensorFlow Federated (TFF) 96

4.1.1.3 Фреймворк PySyft..... 97

4.1.1.4 Фреймворк PaddleFL..... 97

4.1.1.5 FedML 98

4.2 Выбор и обоснование фреймворков GMM-классификатора и обеспечения дифференцированной конфиденциальности APL 99

4.3 Разработка структуры программного комплекса распределенной системы многоэлементной классификации, функционирующей в условиях неполноты классов локальных классификаторов..... 100

4.4 Выбор и обоснование среды имитационного моделирования программного комплекса системы федеративного машинного обучения для задачи многоэлементной классификации в условиях неполноты классов локальных классификаторов 102

4.5 Разработка структуры имитационной модели распределенной системы многоэлементной классификации, функционирующей в условиях неполноты классов локальных классификаторов..... 107

4.6 Разработка моделирующего алгоритма предложенной имитационной модели разрабатываемой системы 113

4.7 Методика расчета числа имитационных прогонов..... 117

4.8 Выполнение имитационного эксперимента и получение сравнительной оценки предложенного решения..... 118

4.9 Выводы по главе..... 123

ЗАКЛЮЧЕНИЕ 124

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 127

Приложение 1 Сравнительный анализ существующих проектов…….....137

Приложение 2 Значения показателя фиксированных классов …….138

Приложение 3 Табличное представление значений показателей Accuracy.140

ВВЕДЕНИЕ

Актуальность темы

Развитие систем интеллектуальной обработки данных (ИОД), которые в настоящее время являются основой множества информационных, управляющих и контролирующих систем в различных предметных областях, неразрывно связано с совершенствованием методов, математического, программного, аппаратного обеспечения, технологий машинного обучения (МО), и моделей машинного обучения (ММО). В частности, вопросы МО имеют высокую актуальность для совершенствования систем, реализующих процесс многоклассовой (многоэлементной) классификации и имеющих распределенную архитектуру.

В традиционных системах классификации процесс МО опирается на централизованный сбор, хранение и распределение данных, являющихся обучающими и тестовыми выборками. Однако, постоянно возрастающая сложность этих моделей требует в процессе их обучения все больших объемов анализируемых данных, что существенно усложняет схемы организации центров обработки данных, на базе которых они развертываются. Другим аспектом, усложняющим использование централизованных схем, является потенциальная невозможность консолидации всей совокупности данных обучающих выборок в единой системе хранения. Это может быть связано, как с распределенным характером хранения данных, так и вопросами безопасности их использования в силу конфиденциальности некоторого их подмножества.

Одним из способов решения этой проблемы является использование моделей и методов федеративного МО (ФМО), исследование и разработка которых в последнее время получили активное развитие. В основе функционирования такого вида распределенных систем на основе ФМО лежит взаимодействие множества распределенных вычислительных узлов (worker-nodes), каждый из которых поддерживает локальную модель обучения (ЛМО). Таким образом, сложный процесс обучения декомпозируется на множество простых процессов (реализация стратегии «слабый ученик»), а в дальнейшем производится их агрегирование, в результате которого формируется итоговая (глобальная) модель обучения (ГМО).

Она используется в качестве финальной и применяется для решения прикладных задач. Такой подход наиболее актуален для решения задач многоэлементной классификации, когда множество ЛМО реализуют классификацию для двух и более классов, а ансамблированная ГМО поддерживает задачи классификации по всему множеству классов.

Основными исследовательскими проблемами в предметной области распределенной многоэлементной классификации являются проблемы управления процессом обмена данными выборок локальных классификаторов в нормальных и специальных условиях, который обеспечивает формирование итогового классификатора.

Тематика диссертационной работы соответствует научному направлению ФГКВОУ ВО «Академия Федеральной службы охраны Российской Федерации» «Повышение эффективности функционирования распределенных вычислительных систем».

Степень разработанности темы Существенный вклад в развитие предметной области методов и алгоритмов распределенных систем классификации, и в частности многоклассовых классификаторов на их основе, внесли: Мак-Махан Б. (Brendan McMahan), Ремейдж Д. (Daniel Ramage), Такурта А. (Abhradeep Thakurta), Бэй С. (Stephen D. Bay), Бишоп К. (Christopher M. Bishop)., Вольф Д.А., Агафонова Ю.Д, Соломин А.А.

При этом существующие исследования в основном ориентируются на архитектурные аспекты организации таких систем и разработку протоколов взаимного информационного согласования данных локальных классификаторов в контролируемых условиях, и не в полной мере рассматривают вопросы, связанные со специальными условиями, такими, как неполнота локальных элементных матриц классов, возникающих, если используются, как обучающие выборки на основе общедоступных данных, так и выборки на основе конфиденциальных (применительно к конкретному локальному классификатору) данных.

Таким образом, актуальность темы диссертационного исследования связана с необходимостью разработки специальных средств математического и программного обеспечения управления обменом данными для принятия решения в распределенных системах многоэлементной классификации, функционирующих в условиях неполноты локальных элементных матриц, с целью повышения эффективности процесса классификации.

Объектом исследования является распределенная система многоэлементной классификации.

Предметом исследования являются модели и методы формирования элементных матриц классов в процессе обучения распределенной системы многоклассовой классификации.

Цель и задачи исследования Целью диссертационного исследования является повышение эффективности процесса классификации в распределенной системе многоэлементной классификации в условиях неполных данных для принятия решения, за счет разработки модели и алгоритмов управления обменом данными об оценках ненаблюдаемых классов.

Для достижения поставленной цели необходимо решить следующие частные научные задачи:

1) Разработать модель классификатора распределенной системы многоэлементной классификации, учитывающую неполноту локальных элементных матриц и основанную на представлении меток ненаблюдаемых классов распределением плотности вероятности.

2) Создать алгоритм получения значений оценок вероятностной функции ненаблюдаемых классов локальных классификаторов, основанный на методе расчета параметров статистической вероятностной модели со смешанными распределениями.

3) Разработать алгоритм децентрализованного взаимодействия узлов распределенной системы многоэлементной классификации, функционирующей в условиях неполноты локальных элементных матриц классов, обеспечивающий

получение полной элементной матрицы с учетом потенциально ненаблюдаемых классов.

4) Модифицировать существующую архитектуру распределенной системы многоэлементной классификации и реализующий ее программный комплекс, обеспечивающие формирование итоговой модели обучения классификатора для условий неполноты локальных элементных матриц классов.

Методология и методы исследования При решении поставленных в диссертации задач использовались: методы классификации данных, методы машинного обучения, статистические методы оценки параметров вероятностных моделей, методы математической статистики и планирования экспериментов.

Научная новизна В диссертации получены следующие результаты, характеризующиеся научной новизной:

– модель многоэлементного классификатора, отличающаяся от известных учетом условий неполноты локальных элементных матриц и обеспечивающая представление значений оценки вероятностной функции ненаблюдаемых классов параметрами статистической вероятностной модели;

– алгоритм получения значений оценок вероятностной функции не наблюдаемых классов, отличающийся от известных итерационным оцениванием параметров модели методом максимального правдоподобия и обеспечивающий выбор таких оценок, которые наиболее полно представляют пространство признаков локальных классов;

– алгоритм децентрализованного управления обменом данными системы многоэлементной классификации, отличающийся гибридной схемой взаимодействия узлов в условиях неполноты классов локальных классификаторов и обеспечивающий дополнение их элементных матриц оценками вероятностных функций ненаблюдаемых классов;

– архитектура распределенной системы многоэлементной классификации, отличающаяся от известных реализацией трехэтапной процедуры получения вероятностной функции ненаблюдаемых классов и обеспечивающая формирование

итогового классификатора на основе динамически получаемых локальных элементных матриц.

Теоретическая значимость исследования заключается в том, что предлагаемый новый подход к формированию модели классификатора распределенной системы многоэлементной классификации в условиях неполных данных для принятия решения может быть использован в совершенствовании теоретических и экспериментальных перспективных систем интеллектуальной обработки данных.

Практическая значимость работы заключается в повышении эффективности процесса многоэлементной классификации, применимого в различных областях человеческой деятельности. Разработано специальное программное обеспечение компонентов распределенной системы многоэлементной классификации объектов на цифровых изображениях в условиях конфиденциальности части данных обучающей выборки. Предложены рекомендации для существующих вариантов систем интеллектуальной обработки данных по реализации процесса распределенной классификации в условиях неполных данных локальных элементных матриц классов.

Достоверность результатов подтверждается использованием при разработке моделей известных математических методов и результатами имитационных экспериментов.

Положения, выносимые на защиту:

1) Модель многоэлементного классификатора обеспечивает представление значений оценки вероятностной функции ненаблюдаемых классов параметрами статистической вероятностной модели.

2) Алгоритм получения значений оценок вероятностной функции не наблюдаемых классов обеспечивает выбор оценок вероятностной функции, наиболее полно представляющих пространство признаков локальных классов.

3) Алгоритм децентрализованного управления обменом данными системы многоэлементной классификации обеспечивает дополнение локальных элементных оценками вероятностных функций ненаблюдаемых классов.

4) Архитектура распределенной системы многоэлементной классификации обеспечивает формирование итогового классификатора на основе динамически получаемых локальных элементных матриц.

Апробация работы Основные положения диссертационной работы докладывались и обсуждались на следующих конференциях: Сборнике статей научно-исследовательского института систем связи и управления (НИИССУ), XVI Всероссийской научно-практической конференции «Территориально распределенные системы охраны» (Калининград, 2023), XXVIII-th International Open Science Conference «Modern informatization problems in the technological and telecommunication systems analysis and synthesis (MIP-2023'SCT)» (Yelm, WA, USA, 2023), XXIX-th International Open Science Conference «Modern informatization problems in simulation and social technologies (MIP 2024'SCT)» (Yelm, WA, USA, 2024), XIV Всероссийской межведомственной научной конференции «Актуальные направления развития систем обеспечения безопасности объектов государственной охраны и защиты охраняемых объектов, специальной связи для нужд органов государственной власти и специального информационного обеспечения государственных органов» (Орел, 2025), а также на научных семинарах кафедры информатики и вычислительной техники Академии ФСО России (2022–2025 гг.).

Реализация и внедрение результатов работы Результаты диссертации внедрены в практическую деятельность ООО «Айти Интегра Системс» (г. Москва), а также в образовательный процесс Академии ФСО России (дисциплина – «Компьютерные сети»).

Соответствие паспорту специальности Содержание диссертации соответствует п. 3 «Разработка методов и алгоритмов решения задач системного анализа, оптимизации, управления, принятия решений, обработки информации и искусственного интеллекта», п. 5 «Разработка специального математического и алгоритмического обеспечения систем анализа, оптимизации, управления, принятия решений, обработки информации и искусственного интеллекта», п. 15 «Теоретический анализ и экспериментальное исследование функционирования элементов систем управления в нормальных и специальных условиях с целью

улучшения технико-экономических и эксплуатационных характеристик» паспорта специальности 2.3.1. Системный анализ, управление и обработка информации, статистика.

Публикации По результатам диссертационного исследования опубликовано 11 научных работ, в том числе 3 – в изданиях, рекомендованных ВАК РФ, 1 – в издании Scopus и 1 свидетельство о регистрации программы для ЭВМ. В работах, опубликованных в соавторстве, лично автором получены следующие результаты: [1, 5, 7, 9] – модель классификатора многоэлементной системы классификации, функционирующей в условиях неполноты классов локальных классификаторов, [2, 4, 11] – получения значений оценок вероятностной функции не наблюдаемых классов, [3, 4] – алгоритм получения значений оценки вероятностной функции ненаблюдаемых классов локальных классификаторов, [3, 7, 8, 9, 10] – архитектура системы многоэлементной классификации с федеративным машинным обучением, поддерживающая формирование итогового классификатора на основе динамически полученных матриц классов локальных классификаторов.

Структура и объем работы Диссертация состоит из введения, четырех глав, заключения и 3 приложений. Работа изложена на 146 страницах машинописного текста, включая 38 рисунков, 14 таблиц и список литературы из 106 наименований.

Глава 1. Разработка модели классификатора распределенной системы многоэлементной классификации, учитывающей неполноту классов подмножества ее локальных классификаторов

1.1 Анализ предметной области распределенных систем с федеративным машинным обучением

В настоящее время неоспоримым фактом является то, что прогресс в исследованиях предметной области искусственного интеллекта (ИИ), и в частности, развитие методов и технологий в предметной области машинного обучения (МО), в том числе глубокого МО (deep learning) [1,2] привели к прогрессу в различных областях человеческой деятельности. Современные модели машинного обучения (ММО) обеспечивают эффективную поддержку решения сложных многофакторных задач, лежат в основе систем распознавания, классификации и кластеризации, а также обеспечивают реализацию логического вывода для систем поддержки принятия решения и управления технологическими процессами. Существенное влияние использование ММО оказывает на такие предметные области как медицинская диагностика [3], экономическая статистика [4], компьютерное моделирование сложных объектов [5] и ряд других.

Современные модели МО поддерживают сотни миллионов (в ряде исключительных случаев – миллиарды) параметров. В их основе лежит принцип использования больших наборов данных, требуемых для достижения точности, необходимой для той или иной предметной области. Важность этапа МО систем ИИ обуславливается необходимостью получения результатов, обеспечивающих возможность их практического применения, что для таких областей как клиническая диагностика заболеваний, поиск и распознавание дефектов и аномалий в различных областях машиностроения, обеспечение управления технологическими процессами объектов критической инфраструктуры является чрезвычайно важным.

Эволюционным подходом к решению задач МО является подход на основе централизованного обучения ММО. В его основе лежит принцип консолидации наборов данных, которые используются в качестве обучающей (training dataset)

и тестовой (testing dataset) выборок в рамках единого хранилища данных, и предоставления доступа к нему соответствующей ММО. При этом указанные наборы данных проходят предварительный этап нормализации, для решения проблем связанных, например, с переобучением ММО, а также обеспечением их безопасности [6, 7].

Централизованное обучение ММО реализуется на базе масштабируемых центров обработки данных (ЦОД), вычислительные узлы которых специализированы для решения задач МО и практического применения обученных ММО. В их основе, как правило, лежат специализированные вычислители – ИИ-ускорители (AI-accelerator), такие как NMPU (Neural-Morphing Processing Unit), TPU (Tensor Processor Unit), VPU (Vision Processor Unit) и др. [8]. При этом масштаб вычислительной мощности наиболее развитых из подобных ЦОД достигает сотен тысяч вычислительных узлов [9], а система хранения данных (СХД) составляет десятки петабайт.

Однако, в ряде случает реализация парадигмы централизованного МО является или неэффективной, или невозможной. В первую очередь это связано с отсутствием практической возможности консолидации данных для формирования соответствующих выборок для ММО.

Так, например, для обучения ММО, детектирующих класс опухолей, требуется огромный набор обучающих данных – компьютерной томографии (КТ) или магнитно-резонансной томографии (МРТ), который будет охватывать полный спектр возможных патологий. Однако, подобные данные фактически невозможно консолидировать в единой СХД, поскольку они относятся к категории персональных данных и цензурируются на законодательном уровне большинства стран [10]. Подобное ограничение имеет под собой практическую почву. Так, в [11] рассматривается возможность реконструкции лица пациента по предварительно анонимизированным данным его МРТ.

Другой не менее важной проблемой препятствующей эффективной консолидации данных для обучения ММО является политика безопасности, поддерживаемая компаниями и организациями – источниками данных. Одной

из причин введения подобных политик является учет временных и ресурсных показателей требуемых для сбора, обработки и хранения этих данных – ценных бизнес-активов компаний.

Существуют также и иные причины, препятствующие консолидации данных. Обзор некоторых из них приведен в [12].

Альтернативой централизованной схеме МО, рассмотренной выше, является парадигма федеративного МО (ФМО) – Federative Machine Learning (FML) [13, 14].

ФМО – это парадигма обучения, направленная на решение проблемы управления данными и их конфиденциальности путем совместного обучения ММО без обмена самими данными. В основе ее формирования стало решение проблем МО приложений, реализованных на множестве пользовательских терминальных устройств, подключенных к единой сетевой инфраструктуре и имеющих доступ к сервисам, реализованным централизованно на базе ЦОД. В первую очередь исследования велись по реализации процесса МО для обучения таких приложений как экранная клавиатура Gboard (задача предиктивного ввода текста) и голосовой ассистент Google Assistant (задача распознавания речи пользователя) для операционной системы Android. Таким образом, предметная область ФМО на начальном этапе своего формирования находилась на пересечении следующих предметных областей МО (рисунок 1.1):

- локальное МО (или самообучение (Self Learning): процесс МО, при котором ни наборы данных, ни параметры ММО не покидают узел, на котором выполняется МО;

- централизованное МО: наборы данных передаются в централизованную вычислительную инфраструктуру, которая отвечает за формирование ММО на их основе;

- граничные вычисления (Edge Computing) [15], реализующие процесс совместного МО (Collaborative Learning): все варианты распределенного взаимодействия узлов (обмен наборами данных и параметрами ММО) с целью получения каждым из узлов выгоды от процесса совместного МО. Особенностью

такого подхода является децентрализованный характер процесса агрегации ММО, например, на основе peer-to-peer взаимодействия узлов.

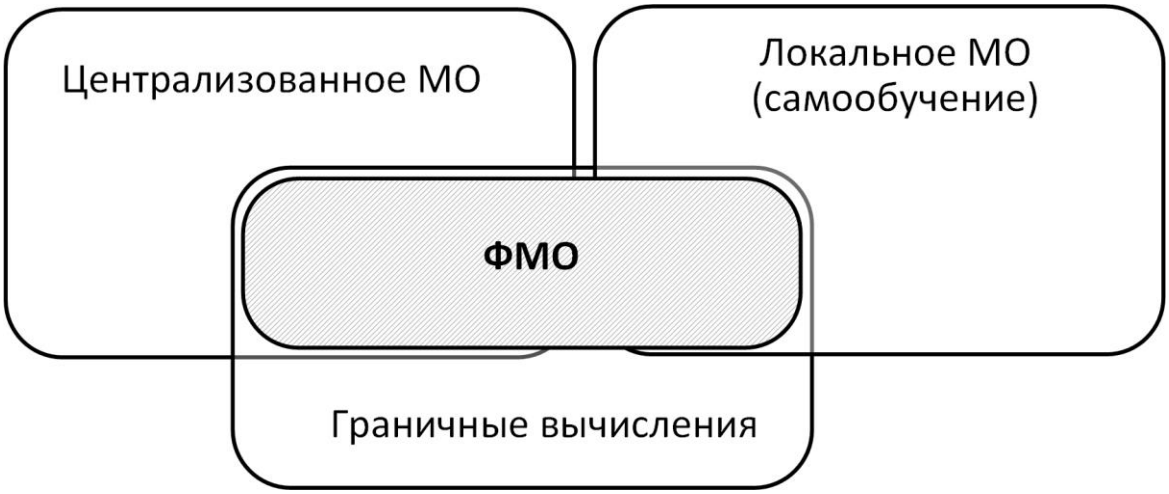

Рис. 1.1 Обобщенное представление места предметной области федеративного машинного обучения

Обобщение системной и функциональной моделей ФМО на основе решения указанных выше задач, было представлено в 2015 году в работе [16].

Исходя из этого обобщенного представления, ФМО можно представить как специализированный процесс распределенного МО, при котором ММО, именуемая консенсусная или глобальная ММО (ГМО), находится у центрального поставщика услуг (обобщенно – сервер или центральный узел) и распространяется на подмножество клиентов (обобщенно исполнителей или Worker-узлов). Распространяемая ГМО используется в качестве начальной версии ММО и обучается на данных каждого Worker-узла. Обученная на Worker-узле ММО именуется локальная ММО (ЛМО). В силу структурной и функциональной организации системы ФМО обученная ЛМО и наборы данных, на которых она обучается, остаются Worker-узле. После обучения ЛМО их обновленные параметры (веса, градиенты, др.) передаются на центральный узел для агрегации и обновления параметров ГМО. Указанный процесс именуется раундом ФМО. После его завершения обновленная ГМО передается на то же или другое подмножество Worker-узлов для запуска следующего раунда ФМО. Указанный подход представлен

в виде обобщенной структурно-функциональной схемы системы ФМО (рисунок 1.2).

Рис. 1.2 Обобщенная структурно-функциональная схема системы ФМО

Таким образом, систему ФМО можно определить, как распределенную итеративную среду МО. В таблице 1.1 представлены основные отличия процесса ФМО и процесса централизованного МО [17].

Таблица 1.1

Отличительные признаки процессов централизованного и федеративного машинного обучения

Критерий	**Централизованное МО**	**Федеративное МО**
Цель МО	Сбор и обобщение данных	Распределенный процесс МО
Обучение ММО	На центральном узле	На Worker-узле
Агрегация ММО	Отсутствует	На центральном узле
Особенность ММО	ММО для множественного совместного использования	Локальная ММО для персонального и/или множественного совместного использования
Процесс распределения	Отсутствует	Параметры ММО

Итеративность процесса МО	Отсутствует	Раунды МО

В настоящее время рынок решения для систем ФМО испытывает существенный подъем, что подтверждается обобщенными данными аналитических агентств (рисунки 1.3 и 1.4).

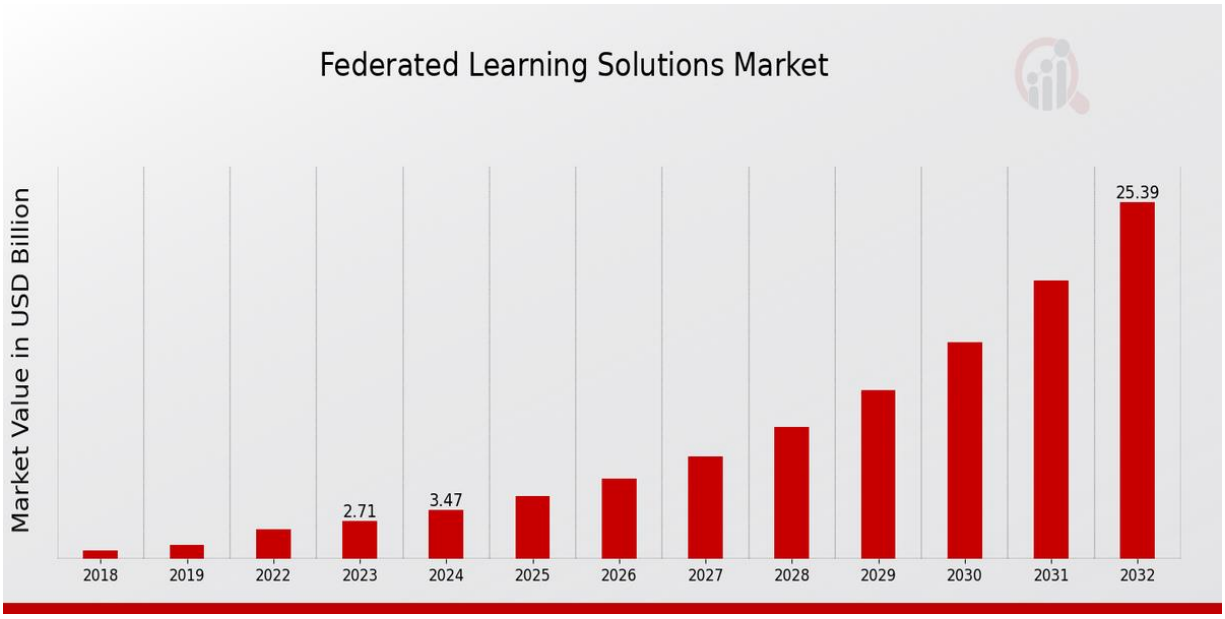

Рис. 1.3 Прогнозная динамика роста стоимости рынка решений систем ФМО на период 2018-2032 гг. [18]

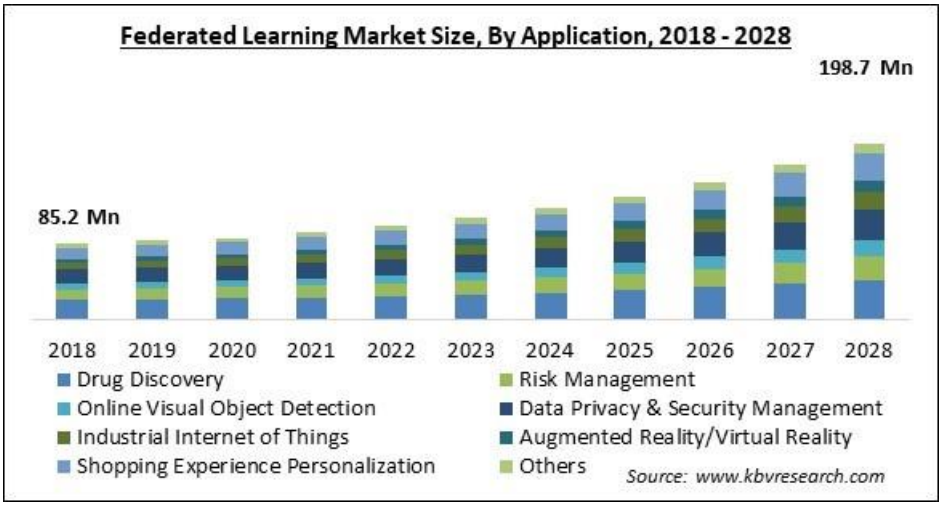

Рис. 1.4 Прогнозная динамика роста стоимости рынка решений систем ФМО на период 2018-2028 гг. с распределением по областям использования [19]

Также в [20] приводится анализ роста числа публикаций, посвященных системам ФМО (рисунок 1.5).

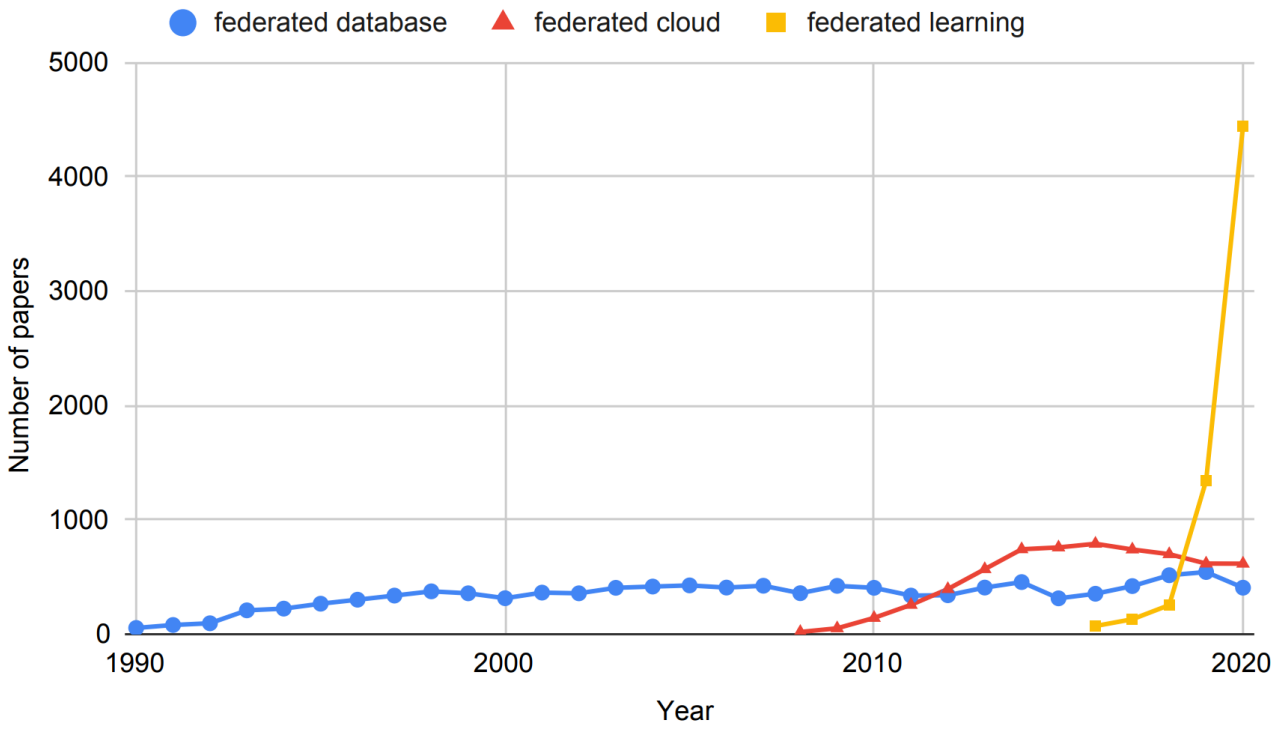

Рис. 1.5 Рост числа научных публикаций, посвященных системам ФМО на период 1990-2020 гг.

1.2. Классификация систем федеративного машинного обучения

Безусловно, развитие и совершенствование предметной области систем ФМО привело к разнообразию методологических и прикладных решений. В [20] приводится развернутая таксономия систем ФМО (рисунок 1.6), позволяющая определить место рассматриваемой в настоящей работе исследовательской задачи.

Из рисунка 1.6 видно, что предложенная классификация является многомерной и включает в себя наиболее важные аспекты архитектуры систем ФМО, а именно:

- разделение данных;
- модель МО;
- обеспечение конфиденциальности;
- модель взаимодействия;

- модель масштабирования;

- модель мотивирования.

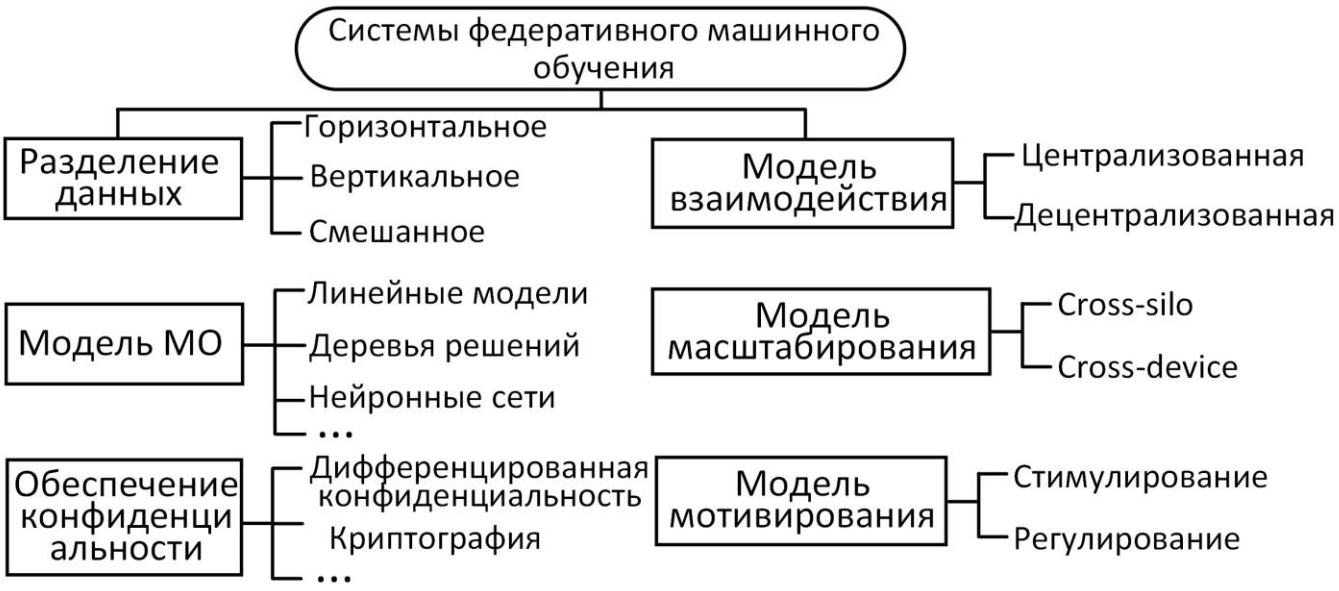

Рис. 1.6 Классификация систем ФМО.

К наиболее важным архитектурным аспектам систем ФМО, применительно к исследовательской задаче данной работы, следует отнести:

1. Модель масштабирования системы с ФМО. Активное развитие систем ФМО в последнее время позволяет выделить с точки зрения масштаба из организации и принципов взаимодействия узлов следующие два класса:

- cross-silo – подобные системы с ФМО характеризуются небольшим количеством узлов, обладающих относительно высокими вычислительными ресурсами. В первую очередь к ним следует отнести распределенные ЦОД, а также системы, поддерживаемые специализированными государственными или коммерческими структурами, например, сети клиник, научно-исследовательских центров единой специфики, распределенные гибкие промышленные производства;

- cross-device – подобные системы с ФМО характеризуются достаточно большим количеством узлов, обладающих относительно низкими вычислительными ресурсами. Примерами подобных систем являются сервисы интеллектуальных голосовых ассистентов на базе мобильных операционных систем (централизованная

модель взаимодействия), а также получающие активное развитие в последнее время сети устройств Интернета вещей (Internet of Things – IoT) и беспилотных транспортных средств (autonomous and self-driving vehicle).

2. Модель МО. К наиболее распространенным видам ММО, используемым в современных системах с ФМО относятся:

- линейные модели. К ним в первую очередь следует отнести модели линейной (задача регрессии) и логистической регрессии (задача классификации), которые предсказывают целевую переменную, используя линейную функцию входных признаков. Являются относительно простыми и вычислительно эффективными с точки зрения интерпретации выходных данных. При этом такие ММО предполагают линейность и независимость переменных, что для реальных задач бывает редко достижимо;

- деревья решений (деревья классификации, регрессии) – иерархические древовидные структуры, состоящие из решающих правил вида «If-Then-Else», которые автоматически генерируются в процессе обучения на обучающем наборе данных. Являются просто интерпретируемыми. Основными проблемами использования деревьев решений является высокая ресурсоемкость их алгоритмов и возможность переобучения;

- нейронные сети. Являются наиболее обширным и разнообразным классом ММО для систем ФМО, благодаря высокой скорости обучения. К особенностям этого класса ММО следует отнести высокую зависимость от качества обучающих наборов данных, требующих сложного и трудоемкого этапа их подготовки, а также высокая обобщающая способность и возможность переобучения. Подход к моделированию для эффективного захвата нелинейности в данных, в котором процесс вывода происходит очень быстро. Кроме того, процесс обучения нейронных сетей требует больших вычислительных ресурсов, в силу чего разрабатываются специализированные вычислители (GPU, TPU, NPU и др.).

Классификационные признаки, связанные с разделением данных, их конфиденциальностью, и моделью мотивирования систем с ФМО рассматриваются в п. 1.3.1.

1.3. Исследование проблем неоднородности наборов данных в системах федеративного машинного обучения

В рамках представленной в п. 1.2 классификации систем с ФМО наиболее существенное значение имеют признаки, связанные с неоднородностью наборов данных, используемых для обучения ММО на Worker-узлах системы.

В случае систем с централизованным МО важным условием является соблюдение принципа независимости и одинакового распределения данных в обучающей и тестовой выборках данных (так называемая проблема IID – Independent and Identically Distributed Data [21]). Он достигается благодаря централизованному хранению данных и решению задачи их предварительной нормализации.

В рамках систем с ФМО использование принципа IID является практически недостижимым по следующих причинам:

- различные Worker-узлы могут иметь доступ к различным по объему и однородности наборам данных;

- политики безопасности, реализуемые в Worker-узлах (в первую очередь это относится к cross-silo системам), обеспечивают конфиденциальность или цензурирование части данных для их использования в обучающих наборах данных.

Кроме того, системы с ФМО обучают ГМО на данных, используемых Worker-узлами, распределенными как территориально, так и логически. Второе означает, что данные, собираемые и анализируемые отдельными Worker-узлами, являются в общем случае уникальными относительно других Worker-узлов. Это приводит к эффекту, аналогичному эффекту дрейфа данных, который известен в централизованной схеме МО [25], когда наблюдается сдвиг распределения между обучающими и тестовыми наборами данных. В системах ФМО этот сдвиг распределения следует рассматривать, как разницу между наборами данных ЛМО i-го и j-го Worker-узлов, обусловленные перекосом в:

- распределении признаков;

- распределении идентификаторов (меток классов);

- несбалансированном количестве данных в обучающих выборках Worker-узлов.

Кроме того, к подобному сдвигу может приводить изменение количества Worker-узлов в каждом раунде обучения ЛМО, что наиболее характерно для cross-device систем с ФМО.

Таким образом, с точки зрения однородности наборов данных для процесса МО, системы с ФМО относят к Non-IID системам.

1.3.1. Схемы неоднородного разделения наборов данных в системах с федеративным машинным обучением

Исходя из этого, в настоящее время в исследованиях, посвященных ФМО, рассматриваются две базовые схемы разделения данных: горизонтальное ФМО и вертикальное ФМО (рисунок 1.7).

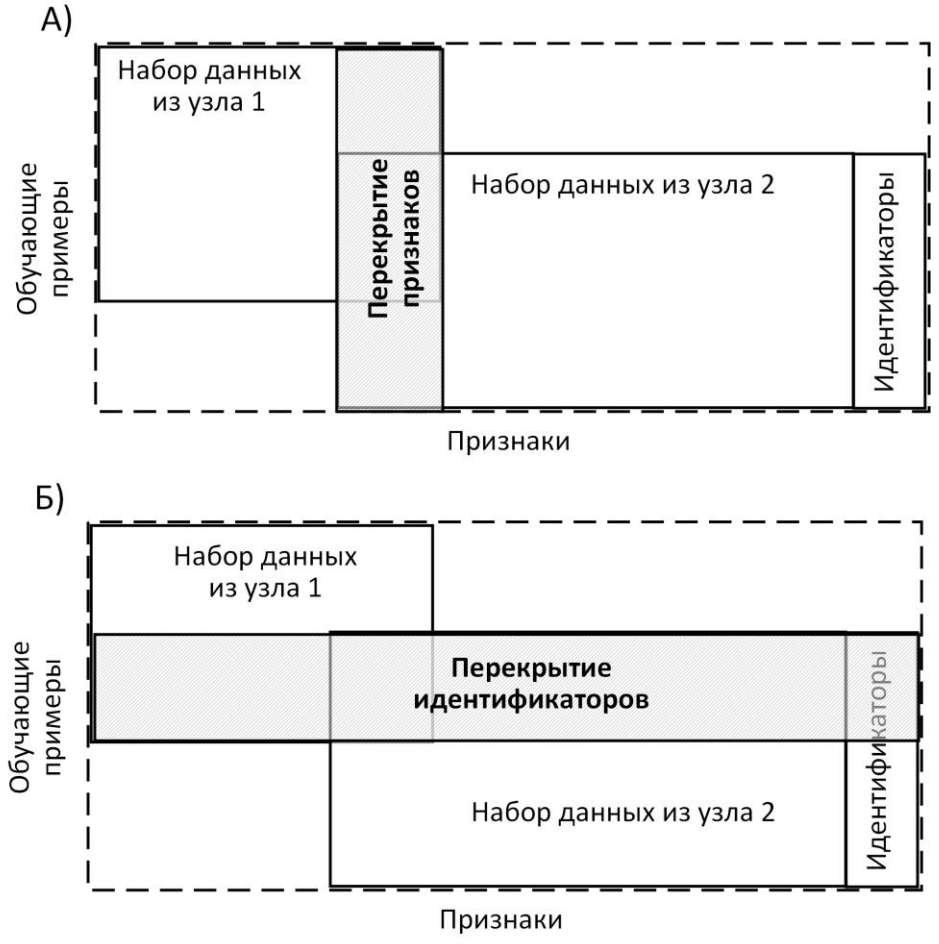

Рис. 1.7 Вертикальная (А) и горизонтальная (Б) схемы разделения данных в федеративном машинном обучении

Вертикальное федеративное обучение (Vertical FL - VFL) [26] предполагает, что наборы данных от разных Worker-узлов не разделяют одно и то же пространство признаков, но могут разделять одно и то же пространство идентификаторов выборки (рисунок 1.7 Б). Кроме того, предполагается, что информация об идентификаторах (метках) хранится одним Worker-узлом. Например, при обучении рекомендательной ММО в предметной области электронной коммерции, Worker-узлами являются маркетплейсы, где зарегистрирован пользователь и банк, хранящий его денежные средства. При этом только банк хранит в своих наборах данных информацию об идентификаторах. Ключевой проблемой вертикального ФМО является вопрос о включении в процессе обучения ГМО локальной информации о метках от одного Worker-узла с сохранением конфиденциальности данных.

Горизонтальное федеративном обучении (Horizontal FL - HFL) [26] предполагает, что наборы данных от разных Worker-узлов разделяют одно и то же пространство признаков, но могут не разделять пространство идентификаторов выборки (рисунок 1.7 А).

В рамках настоящего исследования рассмотрение ограничивается горизонтальной схемой ФМО.

1.3.2. Исследование подходов к обеспечению конфиденциальности наборов данных и параметров моделей в системах с федеративным машинным обучением

Очевидно, что проблема неоднородности данных в системах с ФМО порождает не только исследовательские проблемы их условной нормализации относительно множества узлов.

Не являясь в общем случае носителем информации о признаках и идентификаторах объектов обучающих выборок, они, тем не менее, опосредованно могут способствовать нарушению конфиденциальности наборов данных.

Так в [27] рассматривается следующие внешние факторы обобщенной модели угроз конфиденциальности данных в процессе МО:

- манипулирование входными данными ММО;

- реконструкция объектов, описываемых наборами данных, по пространству признаков (частичная или полная);

- деанонимизация объектов, описываемых наборами данных, по обезличенным персональным данным и иным признакам, используемым для обучения или тестирования ММО;

- определение принадлежности объектов, описываемых наборами данных, с заданным набором признаков к обучающей или тестовой выборкам;

- восстановление значений признаков, с использованием которых производилось обучение ММО (инверсное обучение), а также ряд других факторов.

Решению проблемы обеспечения конфиденциальности наборов данных и параметров ММО посвящен ряд исследований [28, 29, 30, 31]. Их рассмотрение позволило выделить два класса подходов к решению проблемы конфиденциальности в системах ФМО:

1. Методы дифференцированной конфиденциальности (Differentially Privacy – DP).

2. Методы защищенных многосторонних вычислений (Secure Multi-Party Computation – SMPC).

В [31] дифференцированная конфиденциальность определяется как набор методов, предназначенных для защиты конфиденциальности отдельных записей набора данных с обеспечением сохранности информации в них. Фактически, дифференциальная конфиденциальность – это механизм, который гарантирует статистическую неразличимость отдельных входных данных путем внесения в их значения возмущений. В общем случае этот механизм реализует принцип «приватность через неопределенность» за счет включения в состав данных случайной (псевдослучайной) компоненты. Общим подходом к обеспечению дифференцированной приватности являются подход, связанный с применением шумоподобных функций. К наиболее известным механизмам подобного рода

относятся механизм Лапласа и Гаусса [31]. Очевидным недостатком дифференцированной конфиденциальности является сложность реализации ее механизмов в обеспечении достаточного уровня производительности ММО, в первую очередь по показателю ее точности.

Методы защищенных многосторонних вычислений основаны на использовании специализированных протоколов MPC, которые функционируют следующим образом: после процесса обучения ЛМО на базе локальных наборов данных Worker-узлы отправляют параметры ЛМО на сервер, используя схему шифрования, которая позволяет серверу выполнять вычисления над зашифрованными данными. В частности, сервер имеет возможность вычислить средневзвешенное значение всех зашифрованных параметров, полученных от Worker-узлов, но не может получить исходные параметры ЛМО отдельного Worker-узла. В [30] приводятся результаты сравнительных экспериментальных исследований, подтверждающих несущественное снижение точности ММО при использовании методов SMPC.

При этом, следует отметить, что внедрение методов SMPC требует от системы ФМО включения в ее состав инфраструктуры обеспечения информационной безопасности, основанной на шифровании данных, что в большинстве случаев является недостижимым для систем ФМО гетероморфного типа (с независимыми Worker-узлами).

В силу этого в настоящем исследовании рассматривается подход обеспечения конфиденциальности наборов данных и параметров моделей на основе подходов дифференцированной конфиденциальности (DP).

1.3.3. Исследование основ дифференцированной конфиденциальности и подходов к ее реализации

Впервые механизмы DP были рассмотрены и реализованы в предметной области распределенных баз данных [31]. В общем виде определение DP дается в [48] – гарантия системы обработки данных для каждого ее пользователя, который предоставляет данные для анализа: результат дифференцированного анализа

конфиденциальности будет примерно одинаковым, независимо от того, предоставляет ли пользователь свои (конфиденциальные) данные или нет. Обобщенно это представлено на рисунке 1.8.

Рис. 1.8 Обобщенное представление условия дифференцированной конфиденциальности

В формальном виде условие DP формулируется следующим образом:

$$\frac{P\left[\mathrm{DP}(\mathrm{D}_1)\in \mathrm{O}\right]}{P\left[\mathrm{DP}(\mathrm{D}_2)\in \mathrm{O}\right]} \le \mathrm{e}^{\varepsilon}, \qquad (1)$$

где $P\left[\mathrm{DP}(\mathrm{D}_1)\in \mathrm{O}\right]$ и $P\left[\mathrm{DP}(\mathrm{D}_2)\in \mathrm{O}\right]$ – вероятности того, что полученный результат O принадлежит данным, поддерживаемым механизмом DP, из БД D_1 и D_2 соответственно, а ε – параметр конфиденциальности, именуемый потерей конфиденциальности (confidentiality loss).

Как было указано в п. 1.3.1, наиболее распространенным механизмом реализации DP является «зашумление» – смешивание конфиденциальных данных со статистическим шумом (например гауссовским или Лапласса). Пример такого смешивания для изображения из обучающего набора данных CIFAR100 представлен на рисунке 1.9.

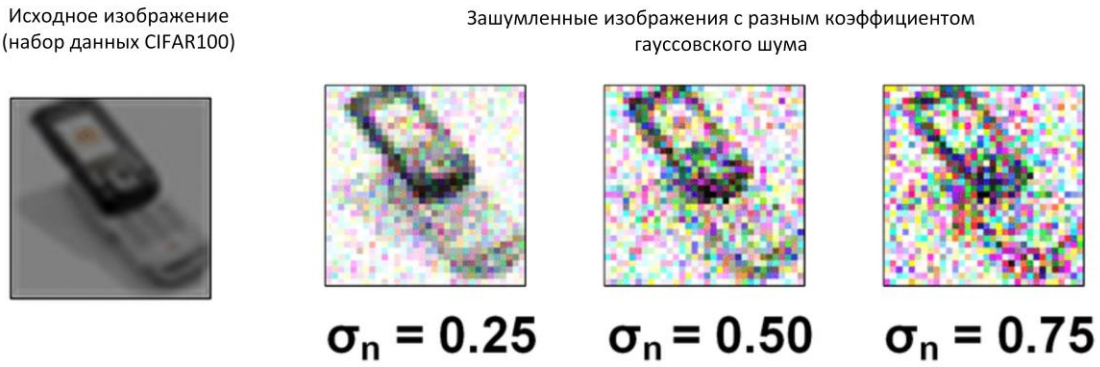

Рис. 1.9 Пример гауссовского «зашумления» изображения

Анализ источников, посвященных проблеме DP в системах ФМО [49, 50, 51] показал, что наиболее известными механизмами реализации DP в процессе ФМО являются:

1. Дифференцированная конфиденциальность на основе стохастического градиентного спуска (Differentially Private Stochastic Gradient Descent (DP-SGD)) [49, 50].

Дифференцированная конфиденциальность на основе стохастического градиентного спуска основана на модификации обновления МО, вычисляемой наиболее распространенным оптимизатором, используемым в глубоком обучении: стохастическим градиентным спуском (SGD).

Как правило, метод SDG обучается итеративно. На каждой итерации из обучающего набора данных отбирается подмножество обучающих примеров (batch) (в методе DP-SGD именуется minibatch). Оптимизатор вычисляет среднюю ошибку модели в выбранном подмножестве minibatch, а затем дифференцирует ее по каждому из параметров модели для получения вектора градиента. Наконец, параметры модели (θ_t) обновляются путем вычитания этого градиента (∇_t), умноженного на константу η (скорость обучения, которая определяет, насколько быстро оптимизатор обновляет параметры модели).

Для получения условия DP (выражение 1) метод DP-SGD делает две модификации относительно традиционного метода SGD:

- градиенты, которые вычисляются на основе каждого minibatch (а не усредняются по нескольким примерам) обрезаются для управления их чувствительностью;

- к их сумме добавляется сферический гауссов шум b_t для получения неразличимости, необходимой для DP.

В общем виде принцип реализации метода DP-SGD представлен на рисунке 1.10, а формальное его определение дается в выражении 2.

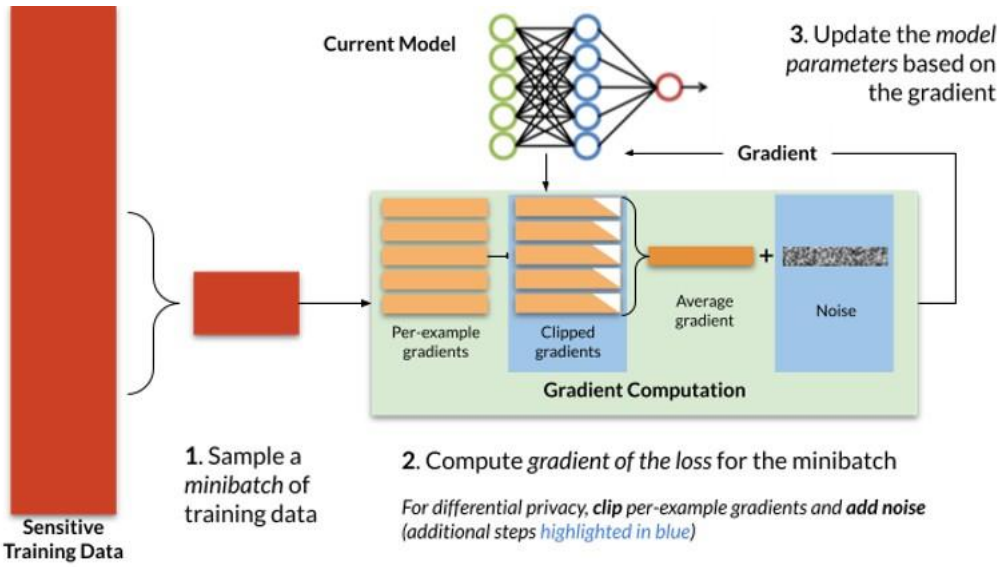

Рис. 1.10 Принцип реализации метода DP-SGD (взято из [50])

$$\theta_{t+1} \leftarrow \theta_t - \eta(\nabla_t + b_t) \qquad (2)$$

К основным недостаткам метода DP-SGD, рассмотренным в [50] относятся:

- снижение точности модели МО за счет добавления сферического гауссова шума b_t в процессе обучения МО;

- сложность использования в процессе многоклассовой классификации, который рассматривается в п. 1.4;

- сложность использования в системах с ФМО.

2. Модельно-независимое конфиденциальное обучение (Model Agnostic Private Learning M-APL или APL) [50, 51]. Оно основана на предварительном

использовании специализированного (закрытого для множества ЛМО) фреймворка для разметки векторов открытых признаков. Далее эти заново размеченные открытые данные используются для обучения открытых ЛМО для получения, в конечном итоге, ГМО.

1.4. Исследование подходов к решению задачи многоэлементной классификации в системах с федеративным машинным обучением

В общем случае процесс МО заключается в извлечении необходимой информации из некоторого множества обучающих данных (набора данных – dataset, sample set) $N=\{x_1,...,x_n\}$, характеризующихся множеством признаков $a_1,...,a_i \in A$, где A – признаковое пространство. Признаки могут иметь числовые или номинальные значения, где каждый экземпляр связан с требуемым выходным значением y_j.

Цель состоит в том, чтобы сформировать систему МО способную предсказывать с заданной или пригодной способностью к обобщению выход y_j для нового (не рассматриваемого ранее) элемента $x_k \notin N$.

В зависимости от реализации процесса МО и вида выходного значения y_j реализуются следующие задачи МО [32]:

- задача регрессии – формирование прогноза на основе выборки данных с различными признаками. Выходное значение $y_j \in \mathbb{R}$ при этом принадлежит всему множеству вещественных чисел;

- задача классификации – формирование категориального отклика системы МО на основе набора признаков. Выходное значение $y_j \in \mathbb{C}$ принадлежит конечному множеству, именуемому множество классов $\mathbb{C}=\{c_1,...,c_m\}$, где элемент c_i именуется меткой класса;

- задача кластеризации – формирование групп данных, со схожим набором признаков;

- задача снижения размерности – решение проблемы редукции, то есть сведения множества признаков к другому множеству, имеющему меньшую мощность;

- задача выявления (распознавания) аномалий – решение проблемы разделения стандартных и нестандартных данных. От задач классификации и кластеризации эта задача отличается малой мощностью множества $N=\{x_1,...,x_n\}$ в силу редкости аномальных явлений.

В рамках настоящего исследования рассматривается задача классификации.

В задаче классификации система МО, сгенерированная обучающим алгоритмом, представляет собой функцию отображения, определенную по шаблонам $A^i \to \mathbb{C}$. Такая функция именуется **классификатором**.

Задачи классификации широко используются в реальных приложениях. Многие из них являются задачами классификации, которые включают две метки классов, например значения $c_1=1$ (да) и $c_2=0$ (нет). Такие задачи именуются задачами бинарной (двухэлементной) классификации При наличии более двух меток классов задачи классификации именуются многоэлементными (многоклассовыми).

Таким образом, под **многоэлементной классификацией** (multi-element, multi-class classification) подразумевается возможность с помощью модели классификатора решать задачу классификации относительно для более чем для двух меток классов. То есть, являясь вариантом бинарной классификации, такая классификация расширяет модель классификатора для более сложных случаев.

Область применения многоэлементной классификации разнообразна. Например, в области биоинформатики, классификации микрочипов, которые работают с несколькими метками классов [33], приложениях компьютерного зрения (Computer Vision – CV), таких как распознавание заданных объектов [34], отпечатков пальцев [35] и языка жестов [36], в предметной области медицины в задачах как классификация рака [37] или сигналов электроэнцефалограммы [38].

Обобщенная схема системы многоэлементной классификации с ФМО представлена на рисунке 1.11.

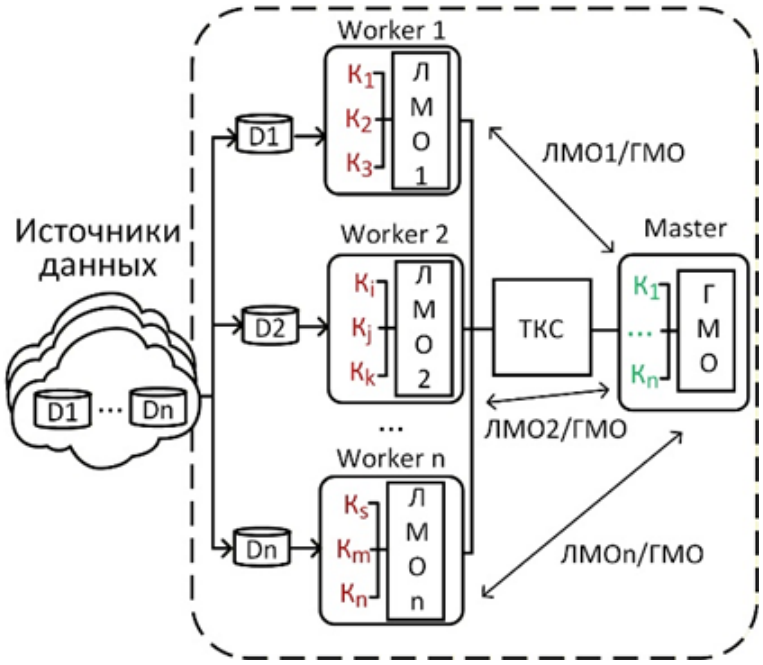

Рис. 1.11 Схема системы обработки данных с ФМО для многоэлементной классификации

Из рисунка видно, что Worker-узлы системы представлены локальными многоэлементными классификаторами (далее, локальные классификаторы) $C^{(i)}$, каждый из которых реализует локальную МО (ЛМО), которая поддерживает подмножество меток классов $K_i \subset K$, принадлежащих множеству меток классов $K=\{k_1, k_2,\ldots, k_n\}$. Итоговый многоэлементный классификатор $C^{(N)}$ реализует глобальную МО (ГМО), агрегируя множество ЛМО. Параметры ГМО возвращаются локальным классификаторам, позволяя им распространить собственные ЛМО на все множество меток классов K.

Как было указано выше, многоэлементная классификация является обобщенным случаем бинарной классификации, основанным на расширении модели бинарного классификатора на некоторое множество меток классов мощностью K.

В графическом виде пример формирования разделяющих гиперплоскостей $a_i(x,\omega_i)$ для трех (i=3) классов приведен на рисунке 1.13.

Обобщенное представление такой системы моделей классификаторов для мощности j множества K элементов (классов) определяется как:

$$\begin{cases} a_1(x_i,\theta_1)=+1 \rightarrow x_i \in k_1 \\ a_2(x_i,\theta_2)=+1 \rightarrow x_i \in k_2 \\ \dots \\ a_j(x_i,\theta_j)=+1 \rightarrow x_i \in k_j \end{cases}, \quad (3)$$

где θ_i – параметры модели, например, весовые коэффициенты, значения которых выбираются таким образом, чтобы положительное (+1) решение классификатора соответствовало k_i классу.

Обобщая выражение 3, получаем выражение 4, определяющее модель многоэлементного классификатора на основе знаковой (+1,-1) или индикаторной (1, 0) функций.

$$a_i(x,\theta_i) = \mathrm{sign}\left(g_i(x,\theta_i,\theta_j)\right) = \begin{cases} +1 \rightarrow y=k_i \\ -1 \rightarrow y \neq k_i \end{cases} \quad (4)$$

После обучения j моделей локальных классификаторов, решением задачи многоэлементной классификации является модель многоэлементного классификатора $C^{(K)}$, реализующая выбор класса с наибольшим положительным отступом от разделяющей гиперплоскости:

$$C^{(K)} = \arg \max_{k \in K} g_i(x,\theta_i,\theta_j) \quad (5)$$

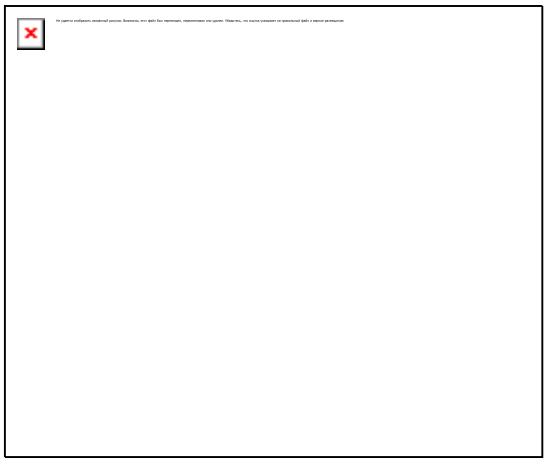

Рис. 1.12 Обобщенное представление многоэлементной (многоклассовой) классификации с тремя классами

В общем случае представленная выше постановка задачи многоэлементной классификации является труднореализуемой. В силу этого в большинстве исследований, рассматривается подход к ее решению на основе принципа бинаризации – сведения задачи многоэлементной классификации к частным задачам бинарной классификации и синтез классификатора $C^{(K)}$ путем агрегирования результатов функционирования бинарных классификаторов.

1.4.1. Исследование способов бинаризации в задачах многоэлементной классификации

Как было указано выше, разделение задачи многоэлементной классификации на ряд подзадач бинарной классификации упрощает моделирование многоэлементного классификатора. Обычно такие бинарные классификаторы именуют базовыми обучаемыми устройствами (БОУ) или базовыми классификаторами [39]. При этом выходные значения $y_j \in \mathbb{C}$ каждого бинарного классификатора должны быть объединены (агрегированы) с целью получения окончательного решения о предсказываемом классе. Обычно простейшим способом агрегации является применение стратегии голосования, где каждый классификатор голосует за предсказанный класс, и класс с наибольшим количеством голосов предсказывается.

Анализ исследований, посвященных проблемам бинаризации многоэлементных задач позволил выявить три наиболее распространенные стратегии их декомпозиции:

1. Стратегия «один против одного» (OVO, от One-Vs-One) [40], которая заключается в использовании отдельного бинарного классификатора для различения каждой пары классов.

2. Стратегия «один против всех» (OVA, от One-Vs-All) [41], которая заключается в использовании бинарного классификатора для различения одного класса от всех остальных.

3. Стратегия «все против всех» (AVA, от All-Vs-All) [41], которая заключается в комбинаторном использовании бинарных классификаторов.

1.4.1.1. Исследование стратегии бинаризации OVO

Стратегия бинаризации OVO делит m-классовую задачу на $\frac{m(m-1)}{2}$ бинарных задач. Каждая задача решается бинарным классификатором, который отвечает за различение разной пары классов. Раунд обучения классификаторов выполняется с использованием в качестве обучающего набора данных только такого подмножества экземпляров из исходного набора данных, который содержит любую из двух соответствующих меток классов множества $\mathbb{C}=\{c_1,...,c_m\}$, тогда как экземпляры с разными метками классов игнорируются.

На этапе проверки шаблон представляется каждому из бинарных классификаторов.

Выход классификатора, заданный $r_{ij} \in [0,1]$, является оценкой бинарного классификатора, различающего классы i и j в пользу первого класса. Для противоположного класса оценка вычисляется, как $r_{ij}=1-r_{ji}$. Если классификатор не предоставляет такой оценки, класс с наибольшей уверенностью является выходным классом классификатора).

Эти выходные данные представлены матрицей оценок R:

$$R=\begin{pmatrix} - & r_{12} & \cdots & r_{1m} \\ r_{21} & - & \cdots & r_{2m} \\ \vdots & & & \vdots \\ r_{m1} & r_{m2} & \cdots & - \end{pmatrix} \quad (6)$$

Конечный выход системы выводится из матрицы оценок с помощью того или иного метода агрегации.

Несмотря на то, что в способе OVO число бинарных классификаторов имеет порядок m^2, каждый классификатор обучается только на образцах из соответствующих классов, и, следовательно, требуемое время на обучения является относительно небольшим. В [42] рассматриваются недостатки способа OVO.

Когда новый шаблон предоставляется всему множеству бинарных классификаторов, некоторое подмножество из них в процессе обучения не рассматривали его, поэтому их выход $y_j \in \mathbb{C}$ не будет значимым (в [42] эти экземпляры называются некомпетентными примерами). Такое предположение не позволяет рассматривать простейший способ агрегации на основе большинства голосов и требует рассмотрения или разработки новых способов агрегации.

1.4.1.2. Исследование стратегии бинаризации OVA

Стратегия бинаризации OVA делит m-классовую задачу на m бинарных задач. Каждая задача решается бинарным классификатором, который отвечает за различение одного из классов относительно всех остальных классов.

Раунд обучения бинарных классификаторов выполняется с использованием всего обучающего набора данных, рассматривая шаблоны из одного класса как положительные, а все остальные примеры как отрицательные.

На этапе тестирования шаблон представляется каждому из бинарных классификаторов, а затем классификатор, который дает положительный выход, указывает выходной класс. Во многих случаях положительный выход не является уникальным, и требуются некоторые методы разрешения конфликтов, рассматриваемые в [42]. Наиболее распространенный из них использует оценки классификаторов для определения окончательного выходного значения $y_j \in \mathbb{C}$, предсказывая класс из классификатора с наибольшей уверенностью.

В способе OVA вместо матрицы оценок (выражение 6) при работе с выходными данными используется вектор оценок: $R=\left(r_1,r_2,...,r_i,...,r_m\right)$.

Несмотря на использование всего набора данных для обучения каждого классификатора, что предотвращает появление некомпетентных примеров, схема OVA усложняет глобальный классификатор и требует более высокого времени обучения. Еще одной проблемой является возможность формирования несбалансированного обучающего набора данных, что приводит к проблемам, рассмотренным в п. 1.3.1.

1.4.1.3. Исследование стратегии бинаризации AVA

Одним из недостатков стратегии обучения OVA является потенциальная возможность недостаточной эффективности получаемой многоэлементной модели, в силу того, что обучение каждой из моделей локальных классификаторов производится независимо от других моделей. В стратегии «все-против-всех» (AVA) бинарные классификаторы решают задачу классификации относительно двух классов. Обобщенное представление такой системы моделей классификаторов для мощности j множества K классов определяется как:

$$\begin{array}{cccc} & 1 & ... & K \\ 1 & 0 & ... & a_{1K}(x,\theta_{12}) \\ ... & ... & ... & ... \\ K & a_{K1}(x,\theta_{K1}) & & 0 \end{array} \qquad (7)$$

Таким образом, модель бинарного классификатора формирует следующие выходные значения:

$$a_{ij}(x,\theta_{ij}) = sign\left(g_{ij}(x,\theta_{ij})\right) = \begin{cases} +1 \rightarrow y=k_i \\ -1 \rightarrow y=k_j \end{cases} \qquad (8)$$

Соответственно в многоэлементном (итоговом, глобальном) классификаторе итоговый результат классификации может быть сформирован по мажоритарному принципу:

$$C^{(K)}=\arg\max_{k\in K}\sum_{i=1}^{K}\sum_{\substack{j=1\\ j\neq i}}^{K}\left[a(x_{ij})=k\right] \qquad (9)$$

1.5. Проблемы не наблюдаемости и неполноты классов в системах многоэлементной классификации с ФМО и подходы к их решению

Очевидно, что на выбор стратегии обучения существенно влияет полнота данных обучающей выборки, получаемой каждым бинарным классификатором.

В силу особенностей эксплуатации систем с ФМО, связанных с неоднородностью (п. 1.3.1) и/или конфиденциальностью подмножеств данных (п. 1.3.2), обрабатываемых каждым из бинарных классификаторов на Worker-узлах, использование стратегий OVO и AVA не представляется возможным. Поэтому в качестве базовой стратегии бинаризации в исследовании предлагается использовать стратегию OVA. В случаях неоднородности и/или конфиденциальность подмножества обрабатываемых данных в системе многоэлементной классификации с ФМО (рисунок 1.11) возникает проблема не наблюдаемости части классов локальными классификаторами $C^{(i)}$. Схема такой системы представлена на рисунке 1.13.

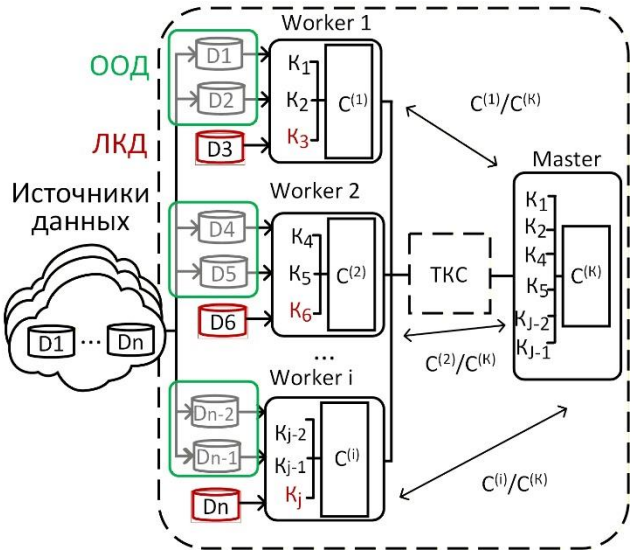

Рис. 1.13 Проблема не наблюдаемости классов в системах многоэлементной классификации с ФМО

Из рисунка видно, что данные, поступающие на вход каждого локального классификатора, делятся на два подмножества:

- общедоступные всем локальным классификаторам данные (далее, ООД);

- данные, конфиденциальные в пределах каждого конкретного локального классификатора $C^{(i)}$ (далее, локальные конфиденциальные данные – ЛКД).

В целом такие данные можно именовать гибридными данными (ГД).

При обработке ЛКД локальный классификатор может выделять объекты с некоторым подмножеством меток классов, которое будет недоступным для других локальных классификаторов (в силу конфиденциальности относящихся к ним объектов). Обобщенное по всем локальным классификаторам, такое подмножество меток классов будем именовать не наблюдаемыми классами. На рисунке 1.13 подмножество наблюдаемых классов представлено классами $K_{набл} = \left\{K_1, K_2, K_4, K_5, K_{j-2}, K_{j-1}\right\}$, а подмножество не наблюдаемых классов представлено классами $K_{\overline{набл}} = \left\{K_3, K_6, K_j\right\}$. Вид элементной матрицы классов для данных ГД представлен выражением 10.

$$X_{j\times d}^{ГД} = \left\{x_j^{ООД}, x_j^{ЛКД}\right\} \tag{10}$$

Проблема не наблюдаемости классов порождает проблему неполноты классов, формируемых итоговым многоэлементным классификатором $C^{(N)}$, использующим только подмножество наблюдаемых классов.

Очевидно, что для представленных проблем требуется разработка нового варианта модели многоэлементного классификатора.

Для решения этой задачи могут быть использованы рассмотренные в п. 1.3.1 подходы к дифференцированной конфиденциальности (DP) и, в частности, принцип независимого конфиденциального обучения (APL).

Вариантом метода APL, получившим практическое применение, является конфиденциальное объединение ансамблей учителей (PATE - Private Aggregation of Teacher Ensembles) [64]. В основе метода PATE лежит принцип агрегации и передачи знаний (knowledge aggregation and transfer). Он основан на предварительном обучении множества моделей ЛМО (локальных классификаторов в задаче многоэлементной классификаци), выполняющих функцию «учитель», на непересекающихся подмножествах конфиденциальных данных. В дальнейшем итоговая модель ГМО (многоэлементный классификатор), выполняющий функцию «ученик», обучается на агрегированном выходе ансамбля ЛМО-учителей на подмножестве вспомогательных немаркированных данных ООД, имитируя результат работы этого ансамбля узлов ЛМО.

Таким образом, метод PATE предполагает, что ансамбль моделей ЛМО решает задачу многоэлементной классификации (в общем случае – любую задачу МО) в совокупности, не раскрывая результат функционирования локальных классификаторов. Ансамбль моделей получается путем разбиения частного набора данных на неперекрывающиеся подмножества данных. Затем их прогнозы агрегируются путем голосования (метод VOTE. Рассматривается в главе 2, п. 2.1), где прогнозируемая метка класса – это метка, чей подсчет голосов является наибольшим. Случайный шум, добавляемый при подсчете голосов, не позволяет результатам агрегирования отражать голоса отдельных узлов ЛМО в целях защиты

конфиденциальности. То есть, внесение шума не изменяет выходные данные агрегирования при достижении консенсуса голосования.

В обобщенном виде принцип реализации дифференцированной конфиденциальности с использованием фреймворка PATE представлен на рисунке 1.14.

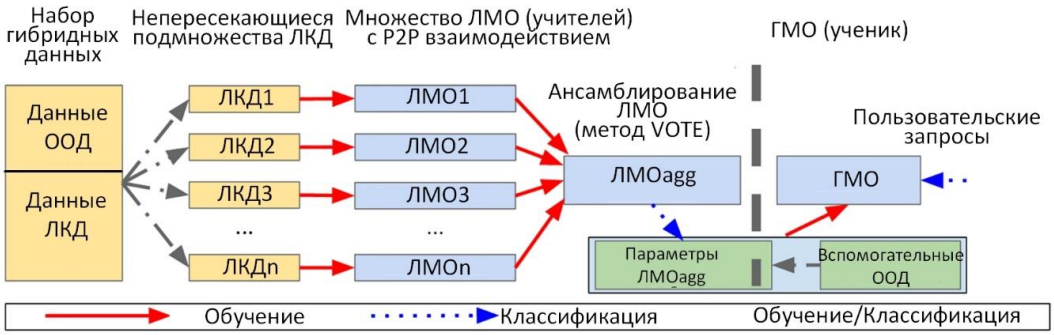

Рис. 1.14 Принцип независимого конфиденциального обучения (APL), реализованный в методе PATE

1.6. Подходы к оцениванию производительности многоэлементной классификации в системах с федеративным машинным обучением

Выбор метрики оценивания производительности задачи многоэлементной классификации в системах с ФМО является важной исследовательской задачей. Анализ источников, посвященных указанной проблеме, выявил, что большинство метрик оценивания производительности предлагаются только для задач бинарной классификации [43,44]. К ним, в первую очередь, относятся:

- accuracy (или коэффициент классификации – Classification Rate);
- precision (или точность классификации);
- recall (или полнота классификации);
- F-мера (среднее гармоническое precision и recall);
- specificity (специфичность – производная метрики recall);
- FDR (fault detection rate) – доля обнаруженных аномалий;
- G-mean (среднее геометрическое – комбинация метрик specificity и FDR);

- ROC-анализ (Receiver Operating Characteristic curve – анализ кривой ошибок), имеющий разновидность AUC-ROC (Area Under Curve – анализ площади под кривой ошибок.

Указанные метрики основаны на так называемой матрице несоответствий (conclusion matrix), представленной в таблице 1.2.

В таблице 1.2 под y понимается истинная метка класса, а под $\hat{y}$ – предсказанная алгоритмом классификации метка класса.

Таблица 1.2

Матрица несоответствий

	y=1	y=0
$\hat{y}$=1	TP (True Positive)	FP (False Positive)
$\hat{y}$=0	FN (False Negative)	TP (True Negative)

Так, исходя из матрицы несоответствий, метрика accuracy рассчитывается следующим образом:

$$accuracy=\frac{TP+TN}{TP+TN+FP+FN} \quad (11)$$

В свою очередь метрики precision и recall рассчитываются по выражениям 12 и 13 соответственно.

$$precision=\frac{TP}{TP+FP} \quad (12)$$

$$recall=\frac{TP}{TP+FN} \quad (13)$$

Очевидно, что различные метрики позволяют оценивать различные аспекты производительности классификации, что, в общем случае, усиливает эмпирические исследования систем МО путем обобщения частных выводов по отдельным метрикам.

Рассмотренные метрики в той или иной степени были адаптированы для оценивания задач многоэлементной классификации, что представлено в [45].

При этом ряд метрик разрабатывался специально для задач многоэлементной классификации. Наиболее известной из них является Каппа Коэна (Cohen's kappa или kappa) [46]. Она является альтернативной метрики accuracy, поскольку компенсирует случайные попадания. В отличие от метрики accuracy, метрика kappa оценивает долю попаданий, которые можно отнести только к работе классификатора, по отношению ко всем классификациям, которые нельзя отнести только к случайности. В ее основе также лежит матрица несоответствий, отличающаяся от представленной в Таблице 1.2. Матрица несоответствий для метрики kappa представлена в Таблице 1.3.

Таблица 1.3

Матрица несоответствия метрики kappa

Правильный класс	Предсказанный класс				
	C_1	C_2	…	C_m	Общий строки
C_1	h_{11}	h_{12}	…	h_{1m}	T_{r1}
C_2	h_{21}	h_{22}	…	h_{2m}	T_{r2}
…	…	…	…	…	…
C_m	h_{m1}	h_{m2}	…	h_{mm}	T_{rm}
Общий столбцы	T_{c1}	T_{c2}	…	T_{cm}	T

Из этой матрицы метрика kappa вычисляется следующим образом:

$$kappa = \frac{n\sum_{i=1}^{m} h_{ii} - \sum_{i=1}^{m} T_{ri} T_{ci}}{n^2 - \sum_{i=1}^{m} T_{ri} T_{ci}}, \qquad (14)$$

где h_{ii} – количество ячеек на диагонали матрицы (количество истинно положительных результатов для каждого класса), n – количество примеров, m – количество меток классов, $T_{ri} = \sum_{j=1}^{m} h_{ij}$ и $T_{ci} = \sum_{j=1}^{m} h_{ji}$ – общее количество

строк и столбцов соответственно. Являясь видом коэффициента, метрика kappa варьируется от -1 (полное несогласие) до 0 (случайная классификация) и до 1 (полное согласие).

Основное различие между метриками accuracy и kappa заключается в оценке правильных классификаций. Метрика accuracy оценивает все успехи по всем классам, тогда как метрика kappa оценивает успехи независимо для каждого класса и объединяет их.

В рамках исследования предлагается совместное использование указанных метрик.

1.7. Постановка задачи исследования

Дано:

Система с ФМО, реализующая N-узловой многоэлементный классификатор (выражение 15), состоящий из N Worker-узлов, которые используют для обучения локальных классификаторов $C^{(i)}$ данные ГД (выражение 10).

$$C^{(N)} = \sum_{i=1}^{N} C^{(i)} \tag{15}$$

Определена функция локальных классификаторов $C^{(i)}$:, как минимизация функции потерь f_L по всему множеству меток классов:

$$C^{(i)} = \underset{\substack{k \in K \\ i}}{\arg\min} f_L \left(y_{ik}^{набл}, \overline{y_{ik}^{набл}}\right) = \tag{16}$$

Задана матрица классов многоэлементного классификатора (множество меток классов мощностью K) (выражение 17).

$$Y_{k\times 1} = \left\{ y_1, y_2, \ldots, y_K \right\} \tag{17}$$

Обучающая выборка многоэлементного классификатора задана выражением 17.

$$\begin{Bmatrix} x_{ij}^{ООД}, y_{ik}^{набл} \\ x_{ij}^{ЛКД}, y_{ik}^{\overline{набл}} \end{Bmatrix}_{i=1}^{k_j} \quad (18)$$

Определены условия полноты классов многоэлементного классификатора:

$$K_i \subset K \quad (19)$$

$$\sum_{i \in N} \sum_{k \in K_i} C^{(i)}(k) = 1 \quad (20)$$

Требуется:

1. Разработать модель многоэлементного классификатора (выражение 15) при не выполнении для каждого узла (выражение 16) условий (выражения 19, 20).

2. Разработать алгоритмы, реализующие в элементной матрице i-го узла (выражение 16) подмножество меток не наблюдаемых классов, поступающих от других узлов многоэлементного классификатора, с целью повышения качества процесса многоэлементной классификации по показателям точности (Accuracy) и Каппа Коэна (kappa).

1.8 Выбор и обоснование функции потерь модели локальных классификаторов

Выбор функций, определяющих модель классификатора является важной исследовательской задачей, поскольку, с одной стороны, они существенно зависят от условий функционирования классификатора, а с другой – оказывают влияние на обоснованный выбор метрик, которые используются для оценивания производительности модели классификатора после процедуры его обучения.

В [22] при решении задачи выбора целевой функции, следует определиться с типом задачи, решаемой моделью классификатора.

В [32] рассматриваются типы задач на основе двух базовых подходов: классификации и регрессии.

В общем виде и в задачах регрессии и в задачах классификации выполняется поиск функциональной зависимости $f:X \to Y$ между множеством входных данных X и множеством целевых значений Y.

При этом в процессе обучения модели подбирается множество параметров $\Theta=\{\theta_1,\theta_2,...,\theta_n\}$, обеспечивающих оптимальное решение представленной выше функциональной зависимости.

Таким образом, общий вид модели можно записать, как:

$$\hat{Y}=f(X,\Theta), \qquad (21)$$

где $\hat{Y}$ – множество полученных в ходе использования модели (предсказанных) целевых значений.

Очевидно, что значение $f(X,\Theta)$ в выражении 21 является точечной оценкой, получение которой, в большинстве случаев, на практике сложно реализуемо. Поэтому при разработке модели классификатора следует рассматривать вероятностный подход [22]. Его идея заключается в том, что модель классификатора вместо одного целевого значения y при заданном входном значении $x \in X$ предсказывает распределение вероятностей на всем множестве целевых значений Y. Такое распределение $p(y|x)$ именуется условным распределением, и с учетом параметров модели классификатора (выражение 21), вероятностную модель классификатора можно записать как $p(y|x,\theta)$. Использование вероятностного подхода связано с тем, что в общем случае модель классификатора предполагает наличие для каждого значения входных данных x_i выходного значения y_i, являющегося эталонным. Однако на практике y_i не всегда принимает конкретное значение, а чаще является распределением вероятностей на множестве Y, так же как и значения, предсказанные моделью классификатора.

Важной задачей, связанной с разработкой модели классификатора, является оценивание вектора параметров $\Theta=\{\theta_1,\theta_2,...,\theta_n\}$. В [32] производится детальный

анализ функций такого оценивания. В частности, наибольшее распространение получили:

- оценка максимального правдоподобия (MLE - maximum likelihood estimation);

- оценка апостериорного максимума (MAP - maximum a posteriori probability), использующая при оптимизации дополнительно априорное распределение величины параметра, оценка которого производится.

Выражения 22 и 23 отражаются получение оценок MLE и MAP соответственно.

$$\hat{\theta}=\operatorname{argmax}\sum_{i=1}^{N}\log p(x^{(i)};\theta), \tag{22}$$

где N – мощность множества входных данных X.

$$\theta=\arg\max_{\theta}\log p(x|\theta)+\log p(\theta) \tag{23}$$

В общем случае, вне зависимости от типа задачи (регрессия или классификация), модель максимизирует или минимизирует функцию потерь f_L (выражение 16).

Очевидно, что, поскольку значения $p(y|x,\theta)$ определены на всем множестве целевых значений $y \in Y$, то, чем меньшее значение вероятности модель классификатора присваивает конкретному истинному значению $y \in Y$, тем величина ошибки классификации выше. Таким образом величина функции потерь может быть задана количественно.

Рассмотренные оценки параметров модели классификатора (выражения 19 и 20) являются механизмом получения функции потерь. Обоснование этого дано в [22, 32].

1.8.1 Обоснование выбора категориальной кросс-энтропийной функции потерь модели локальных классификаторов

К наиболее используемым в предметной области МО функциям потерь в [32] относят:

- среднеквадратичная ошибка (MSE - mean squared error) – представляет собой усредненную по всему множеству значений X возведенную в квадрат разницу между предсказанными и истинными выходными значениями множества Y (выражение 24);

- логарифмическая функция потерь (log loss) или кросс-энтропия (CE – cross entropy) – отображает расхождение между двумя вероятностными распределениями. Высокое значение кросс-энтропии определяет высокую разницу между распределениями и наоборот (выражение 25).

$$MSE=\frac{1}{N}\sum_{i=1}^{N}\left\|\hat{y}^{(i)}-y^{(i)}\right\|^2 \quad (24)$$

$$CE=H(P,Q) = -\sum_{x} P(x)\log Q(x), \quad (25)$$

где P – распределение истинных выходных значений, а Q – распределение предсказанных выходных значений.

Так, для бинарной классификации (K=2) для множества данных мощностью N выражение 25 принимает вид:

$$CE=-\frac{1}{N}\sum_{i=1}^{N}(y_i\log(p_i)+(1-y_i)\log(1-p_i)), \quad (26)$$

где y_i – двоичный индикатор (0 или 1), определяющий истинность метки класса относительно некоторого i-го наблюдения, а p_i – прогнозируемая вероятность модели.

Необходимость рассмотрения указанных видов функции потерь связано с особенностями бинарных классификаторов, когда значение p_i формируется в диапазоне от 0 до 1 для каждого из двух классов. Таким образом, они эффективно сохраняют параметры модели классификатора. Это связано с тем, что в бинарной

классификации знание одной вероятности подразумевает знание другой. То есть, полученный прогноз (0,8; 0,2) можно сохранить, как значение 0,8, а второе значение вычислять (1 – 0,8 = 0,2).

Вариантом кросс-энтропийной функции потерь является категориальная кросс-энтропийная функция потерь (Categorical CE - CCE) [63]. Ее также определяют как softmax loss. Как и CE она измеряет разницу между прогнозируемым распределением вероятностей и фактическим (истинным) распределением классов.

Функция CCE используется, когда в задаче многоэлементной классификации. Он измеряет разницу между двумя распределениями вероятностей: предсказанным распределением вероятностей и истинным распределением, которое представлено вектором значений классов, в котором истинный класс представлен как «1», а все остальные классы – как «0». Функция CCE определяет прогноз, основанный на том насколько модель классификатора «уверена» в истинном классе.

Если модель присваивает истинному классу высокую вероятность, то функция CCE будет низкой. И наоборот, если модель присваивает низкую вероятность истинному классу, функция CCE будет высокой.

Таким образом, при многоэлементной классификации на выходном уровне формируется вектор предсказанных вероятностей p_i.

Для этого случая выражение 26 принимает вид:

$$CCE = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{i,k}\log(p_{i,k}), \qquad (27)$$

где $y_{i,k}$ – двоичный индикатор (0 или 1), определяющий истинность метки класса k относительно некоторого i-го наблюдения, а $p_{i,k}$ – прогнозируемая вероятность для k-го класса относительно некоторого i-го наблюдения.

С целью более наглядного представления сравниваемых значений при расчете логарифмической функции потерь применяется отрицательное значение логарифма.

Это связано с тем, что логарифм чисел меньше 1 имеет отрицательные значения. Обоснование этому дано в [13] (рисунок 1.15).

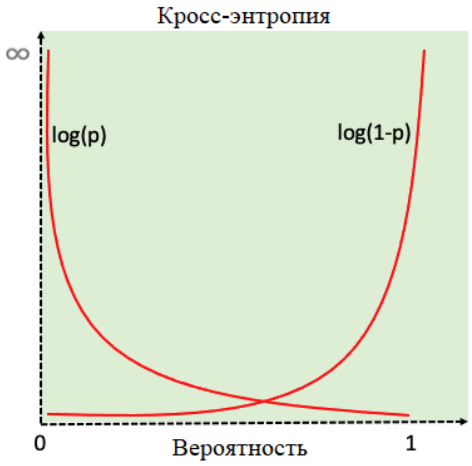

Рис. 1.15 Обобщенное представление категориальной кросс-энтропийной функции потерь для бинарной классификации с тремя классами

Как следует из выражения 27, кросс-энтропийные потери – это скалярная величина, которая количественно определяет, насколько далеки прогнозы модели многоэлементного классификатора от истинных меток классов. Для каждой выборки во множестве входных данных X, кросс-энтропийные потери отражают насколько близко прогноз модели классификатора соответствует истинной метке. Так, меньшие потери для выборки данных указывают на более точный прогноз модели и наоборот. Это связано с тем, что только истинная метка класса вносит свой вклад в значение функции потерь, поскольку для других меток нулевое значение ничего к нему не добавляет.

Таким образом, в рамках исследования предлагается рассматривать вероятностный подход к формированию предсказанных значений классов в модели многоэлементного классификатора, а в качестве функции оценивая его эффективности рассматривать функцию кросс-энтропийных потерь CCE.

1.9 Моделирование многоэлементного классификатора системы с ФМО в условиях полноты классов

Рассмотрим систему многоэлементной классификации с ФМО, функционирующую в условиях полноты классов (рисунок 1.11) Рассмотренная выше схема организации такой системы детально представлена в [47].

Формализуем задача, решаемая i-м классификатором C(i), как задача минимизации кросс-энтропийной функции потерь в условиях полноты классов (выражение 28)

$$C^{(i)} = \arg\min_{\substack{k \in K \\ i}} f_L^{CCE}(y_{ik}^{набл}), \tag{28}$$

где функция потерь f_L^{CCE} определяется как:

$$-\sum_{i=1}^{n_j} \sum_{k \in K_j} l(y_{ik}^{набл}) \log C^{(i)} x_{ij}, . \tag{29}$$

а $l(y_{ik}^{набл})$ - бинарная индикаторная функция $\begin{cases} 1 \rightarrow y_{ik} \in y_{ik}^{набл} \\ 0 \rightarrow y_{ik} \notin y_{ik}^{набл} \end{cases}$, отражающая наличие или отсутствие принадлежности y_{ik} исхода классификации всему множеству наблюдаемых классов.

На основании этого выполнено моделирование многоэлементного классификатора, функционирующего в условиях полноты классов локальных классификаторов:

$$C^{(N)} = \sum_{j=1}^{N} C_j = \underset{C^{(N)}}{\operatorname{argmin}} \left\{ -\sum_{j=1}^{N} \sum_{i=1}^{n_j} \sum_{k=0}^{K} l(y_{ik}^{набл}) \log C^{(i)} x_{ij}) \right\} \tag{30}$$

1.10 Моделирование многоэлементного классификатора для условий неполноты классов

Выполним моделирование многоэлементного классификатора, функционирующего в условиях неполноты классов локальных классификаторов.

При этом модель локального классификатора (выражение 28) в соответствии с условиями постановки задачи исследования трансформируется в выражение 16, а с учетом выбранного вида функции потерь – в выражение 31:

$$C^{(N)} = \sum_{j=1}^{N} C_j = \underset{C^{(N)}}{\operatorname{argmin}} \left\{ - \sum_{j=1}^{N} \sum_{i=1}^{n_j} \sum_{k=0}^{K} l(y_{ik}^{набл}, \overline{y_{ik}^{набл}}) \log C^{(i)} x_{ij}) \right\} \quad (31)$$

Определим гибридные данные, который обрабатывает j-й локальный классификатор, как $ГД_j = ОДД_j + ЛКД_j$.

При этом, в силу конфиденциальности данных $ЛКД_j$, значение конкретного элемента k множества меток классов K можно получить только путем интегрирования оценки плотности вероятности значения x_{ij} каждого j-го локального классификатора. Итоговый многоэлементный классификатор $C^{(N)}$ при этом оперирует суммой условных вероятностей:

$$P(y=k, ГД_j \in C^{(j)} | X), \quad (32)$$

Определив функцию распределения вероятностей меток классов для всей матрицы входных данных $X_{j \times d}$ j-го локального классификатора как $f_X^{(j)}$, а условную вероятность $P(y=k, ГД_j \in C^{(j)} | X)$ как $p_N^{(j)}$, было определено представление глобального классификатора для условия неполноты классов локальных классификаторов:

$$C^{(N)} = \sum_{j=1}^{N} \nu_{j,k} C^{(j)}, \quad (33)$$

где $\nu_{j,k}$ – весовой коэффициент, пропорциональный двум функциям:

$$\nu_{j,k} \propto l(y_{ik}^{набл}, \overline{y_{ik}^{набл}})\delta(f_X^{(j)}, p_N^{(j)}), \quad (34)$$

где $\delta(f_X^{(j)}, p_N^{(j)})$ – некоторая функция с неопределенными параметрами. При этом, в силу неопределенности значений параметров $f_X^{(j)}, p_N^{(j)}$, одним из видов функции $\delta(f_X^{(j)}, p_N^{(j)})$ может быть их произведение $f_X^{(j)} \cdot p_N^{(j)}$. Тогда весовой коэффициент $\nu_{j,k}$ определяется как:

$$\nu_{j,k} = \frac{f_X^{(j)} p_N^{(j)}}{\sum_{j=1}^{L} f_X^{(j)} p_N^{(j)}} \quad (35)$$

Тогда выражение 33 представляется как:

$$C^{(N)} = \sum_{j=1}^{N} C^{(j)} \times \frac{f_X^{(j)} p_N^{(j)}}{f_X} = \sum_{j=1}^{N} C^{(j)} \times \frac{f_X^{(j)} p_N^{(j)}}{\sum_{j=1}^{N} f_X^{(j)} p_N^{(j)}} \quad (36)$$

В силу того, что первый множитель можно задать индикаторной функцией $l(y_{ik}^{набл}, \overline{y_{ik}^{набл}})$ а , второй – весовым коэффициентом $\nu_{j,k}$, было получено итоговое представление модели многоэлементного классификатора, функционирующего в условиях неполноты классов локальных классификаторов:

$$C^{(N)} = \sum_{j=1}^{N} l(y_{ik}^{набл}, \overline{y_{ik}^{набл}}) \times \nu_{j,k} C^{(j)} \quad (37)$$

Для этой модели было определено условие, определяющее наличие неполноты классов хотя бы в одном из локальных классификаторов:

$$\nu_{j,k} \geq \frac{\left(C^{(j)}\right)^{N-1}}{N} \quad (38)$$

Используя представление функции потерь j-го локального классификатора, как кросс-энтропийную функцию $\sum_{k=0}^{K} l(y_{ik}^{набл}, \overline{y_{ik}^{набл}}) \log C^{(j)}(x_{ij})$ и подставив его в

модель многоэлементного классификатора (выражение 36) было получено ее представление для множества локальных классификаторов:

$$\sum_{K}\log\sum_{j=1}^{N}\sum_{k=0}^{K} l(y_{ik}^{набл}, \overline{y_{ik}^{набл}})\log C^{(j)}(x_{ij}) = \sum_{K}\log\sum_{j=1}^{N} \nu_{j,k} C^{(j)}(x_{ij}) \qquad (39)$$

Как показано в [58], с использованием неравенства Йенсена для выпуклой функции среднего случайной величины, можно доказать, что множество локальных классификаторов, достигая минимума категориальных кросс-энтропийных потерь, обеспечивает минимум потерь многоэлементного классификатора:

$$\sum_{K}\log C^{(j)}(x_{ij}) = \sum_{j}\nu_{j,k} l(y_{ik}^{набл}, \overline{y_{ik}^{набл}})\log C^{(j)}(x_{ij})=1 \qquad (40)$$

При этом в силу выдвинутого предположения о том, что в условиях использования в обучающих выборках j-го локального классификатора данных ГД, метки ненаблюдаемых классов данных x_{ij} можно задать значением оценки плотности вероятности.

1.11. Выбор метода классификации для многоэлементного классификатора в условиях неполноты классов локальных классификаторов

Важной задачей, требующей решения в рамках разработки модели классификатора ЛМО, является выбор и обоснование метода классификации, реализуемого в его рамках.

Как было предложено в п. 1.10 (выражения 34-36), в обучающей выборке данных $ЛКД_j$ значения классов, ненаблюдаемых другими узлами ЛМО в матрице предикторов, можно заменить их плотностью вероятности.

То есть, вероятность принадлежности исхода классификации y классу k – $P(y=k, ГД \in w_j, X)$, можно представить, как $p_N^{(j)}$, а распределение вероятностей предикторов $X_{j\times d}$ j-го элемента множества W, как $f_X^{(j)}$.

Очевидно, что на практике значение $p_N^{(j)}$ достаточно просто получить, зная размер обучающей выборки n_j :

$$p_N^{(j)} = \frac{n_j}{\sum_j n_j} \quad (41)$$

При этом, для получения значения $f_X^{(j)}$, в силу его вероятностного характера, невозможно получить строгое аналитическое выражение. Это связано со сложностью ковариационных связей элементов матрицы предикторов $X_{j \times d}$, не позволяющих получить разделение достоверных оценок значений элементов данных $ОДД_j$ и $ЛКД_j$ в рамках их смеси $ГД_j$.

Анализ исследований в предметных областях смешанных вероятностных моделей [59, 60] показал, что для подобных условий традиционные модели статистической классификации, используемые для многоклассовой классификации и основанные, например, на методах SVM и k-means, являются недостаточно подходящими, в силу того, что для получения значения $f_X^{(j)}$ используют простые модели композиционных данных, ограничивающие сумму их компонентов постоянным значением.

Для получения значения $f_X^{(j)}$ предлагается так же использование моделей статистической классификации из предметной области моделей на основе смеси распределений [60].

В общем случае модель смеси представляет собой распределение вероятностей наблюдений в некоторой общей совокупности. При этом они используются для решения двух классов задач:

1. Получение свойств общей совокупности из свойств субпопуляций, входящих в эту совокупность.

2. Формирование статистических выводов о свойствах субпопуляций в отсутствии идентификационной информации об их свойствах, в составе совокупности, только на основе наблюдений за ней.

Очевидно, что вторая задача, соответствует использованию моделей со смешанными распределениями и часто применяется для решения задач кластеризации. Однако, исследования [65, 66, 67] и другие показывают, что эти модели находят применение и в задачах классификации, в частности, многоэлементной классификации.

Анализ этих исследований, а также исследований, связанных с решением задач классификации в условиях неполноты идентификационной информации об используемых данных [61], показал, что широкое применение нашла модель классификатора на основе гауссовой смеси распределений (GMM — Gaussian mixture model) – GMM-classifier.

GMM – это статистическая модель для представления подсовокупностей (субпопуляций) внутри общей совокупности, имеющих нормальное распределение. Такая модель может быть описана двумя типами параметров:

- смесью весов компонентов;

- средними значения компонентов (для одномерного случая) или их ковариациями (для многомерного случая) [59].

Исходя из такого выбора в качестве классификатора модели GMM, значение элементов матрицы $X_{j\times d}$ можно представить гауссовой смесью распределений с R компонентами:

$$f_X^{(J)}(X)=\sum_{r=1}^{R}\pi_r^j f_{x|z_j=r}(X), \qquad (42)$$

где $x|z_j=r \sim N(\mu_r^{(j)},\sum_r^{(j)})$ определяет значение r-го гауссового компонента. При этом z_j является скрытой переменной, которая указывает на принадлежность $ЛКД_j$ к r-му гауссовому компоненту ($z_j=r$), а значение $\pi_r^j=P(z_j=r)$.

Известной проблемой использования модели GMM является волюнтаристский характер выбора мощности множества компонентов R. В [60] для преодоления этой проблемы предлагается использование «графика осыпи» (scree plot) [62] – линейного графика, применяемого в многомерной статистике. График осыпи отображает значения переменной в виде нисходящей кривой, упорядочивая их от наибольшего к наименьшему. Алгоритм нахождения с помощью графика осыпи статистически значимых компонентов называется тест осыпи. В его основе лежит поиск «колена» графика, где значения практически выравниваются, и компоненты слева от этой точки могут быть сохранены как наиболее значимые.

Также из [61] следует, что для оценивания значений модели гауссовой смесью распределения с R компонентами наиболее подходящим является использование итерационных алгоритмов, выбор конкретного варианта из которых является отдельной исследовательской задачей (рассмотрена в п. 3.2.1).

Таким образом, получение значений матрицы предикторов $X_{j\times d}$ является итерационной процедурой, что определяет особенности разрабатываемого в следующей главе алгоритма формирования модели итогового классификатора.

1.12. Выводы по главе

В главе рассмотрен подход к разработке модели многоэлементного классификатора, функционирующего в условиях неполноты классов локальных классификаторов.

В рамках решения этой задачи был выполнен выбор: метода агрегирования, применяемого в задачах бинаризации задач многоэлементной классификации и модельно-независимого машинного обучения, функции потерь модели локальных классификаторов. Выполнено моделирование локального и итогового многоэлементного классификаторов системы с ФМО, функционирующей в условиях, как полноты, так и неполноты классов локальных классификаторов.

В ходе моделирования было выявлено, что решение задачи не наблюдаемости подмножества классов множеством локальных классификаторов может быть связано возможностью обмена вероятностными моделями подмножеств не наблюдаемых классов, формируемыми их классификаторами.

Рассмотрены подходы к решению этой задачи, и выбран тип классификатора на основе модели гауссовой смеси распределений (GMM-classifier).

Таким образом, предложена модель классификатора системы многоэлементной классификации с ФМО, базирующаяся на статистической вероятностной модели, описывающей плотность вероятности ненаблюдаемых меток классов подмножества локальных классификаторов. Разработанная модель отличается от известных возможностью реализации процесса многоэлементной классификации в условиях неполноты классов локальных классификаторов, связанной с наличием в их обучающих выборках гибридных (общедоступных и конфиденциальных) данных.

Глава 2. Разработка алгоритма получения значений оценок вероятностной функции ненаблюдаемых классов системы многоэлементной классификации, функционирующей в условиях неполноты классов локальных классификаторов

2.1 Исследование особенностей многоэлементного классификатора на основе модели гауссовой смеси распределений

Как было рассмотрено в главе 1 (п. 1.10) не наблюдаемые классы локальных классификаторов можно представить гауссовой смесью распределений (GMM) с R компонентами (выражение 42).

В [66-68] достаточно подробно исследованы свойства модели GMM и классификатора на ее основе.

В частности определяется, что в общем случае она представляет собой взвешенную сумму R компонентных гауссовых плотностей:

$$p(x|\lambda)=\sum_{i=1}^{R} w_i g(x|\mu_i,\Sigma_i), \qquad (43)$$

где x – это D-мерный непрерывный вектор данных (в рамках исследования – классов модели ЛМО), w_i,i=1,...,R – смеси весов, а $g(x|\mu_i,\Sigma_i)$,i=1,...,R – D-вариантная гауссовская функция определяющая компоненты гауссовой плотности вероятностей:

$$g(x|\mu_i,\Sigma_i)=\frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma_i|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}(x-\mu_i)'\Sigma_i^{-1}(x-\mu_i)\right\}, \qquad (44)$$

где μ_i – среднее значение вектора, а Σ_i – ковариационная матрица.

При этом веса смеси GMM удовлетворяют следующему ограничению $\sum_{i=1}^{R} w_i$=1i=1 wi = 1.

Очевидно, что полная GMM-модель параметризуется средними векторами, ковариационными матрицами и весами смеси из всех компонентных плотностей:

$$\lambda=\left\{w_i,\mu_i,\Sigma_i\right\},i=1,...,R \qquad (45)$$

Как показано в [67], зависимости от вида векторов GMM-модель будет иметь различную конфигурацию. Например, ковариационные матрицы Σ_i могут быть полной или ограниченной диагональю. Кроме того, параметры могут быть общими или связанными между компонентами Гаусса, например, иметь общую ковариационную матрицу для всех компонентов. Выбор конфигурации GMM-модели (количество компонентов, полные или диагональные ковариационные матрицы и взаимосвязь параметров) часто определяется объемом данных, доступных для оценки ее параметров.

На практике, в силу статистической независимости признаков, вместо полных ковариационных матриц чаще всего используются неполные ковариационные матрицы в виде линейной комбинации диагональных базисных гауссовских ковариаций, а эффект использования множества R полных ковариационных матриц достигается использованием большей мощности множества диагональных ковариационных матриц.

Как было отмечено в п. 1.10, выбор модели GMM-классификатора обусловлен ее способностью представлять большой класс выборочных распределений. То есть, в рамках исследования можно допустить, что отдельные плотности компонентов GMM могут моделировать подмножество не наблюдаемых классов локальных классификаторов.

2.2. Конкретизация модели GMM-классификатора для решения задачи многоэлементной классификации изображений

Выбранный в п. 1.10 метод классификации на основе модели гауссовой смеси распределений (GMM-classifier) имеет широкую область применения и, в рамках

исследования, требует конкретизации для решаемого класса задач многоэлементной классификации.

Полученные в процессе исследования результаты, в первую очередь модель многоэлементного классификатора, функционирующая в условиях неполноты классов локальных классификаторов, предлагается рассматривать в аспекте решения задачи многоэлементной классификации цифровых изображений. Выбор этой задачи обусловлен двумя факторами, актуальными с точки зрения проводимого исследования:

- ростом необходимости решения задач многоэлементной классификации цифровых изображений в различных предметных областях (результаты томографии органов человека, результаты аэросъемки земной поверхности, результаты визуально-технического контроля качества продукции и т.д.);

- наличием во многих предметных областях ограничений, связанных с конфиденциальным характером классифицируемых изображений.

Предлагаемая модель GMM-классификатора обобщенно представлена на рисунке 2.1. В качестве обучающего и тестового набора данных выбрано подмножество цифровых изображений из набора данных CIFAR100 [94], сгруппированное в 10 классов.

Предлагаемая модель является двухуровневой:

- уровень предобработки изображений;

- уровень многоэлементной классификации изображений.

Очевидно, что являясь вероятностной моделью, модель GMM требует предварительной обработки поступающих на ее вход цифровых изображений. Целью такой предобработки является сопоставление классифицируемого изображения с некоторым набором его признаков, относительно которых и формируется модель GMM класса изображения.

В рамках решения задачи предобработки изображений существует множество методов ее реализации. К наиболее известным следует отнести:

- сверточные автоэнкодеры [69];

- методы выделения ключевых (особых) точек [70].

В исследовании в качестве базового метода предобработки выбран метод Scale-Invariant Feature Transform для выделения SIFT-дескрипторов, реализованный в библиотеке SIFT фреймворка компьютерного зрения OpenCV [71].

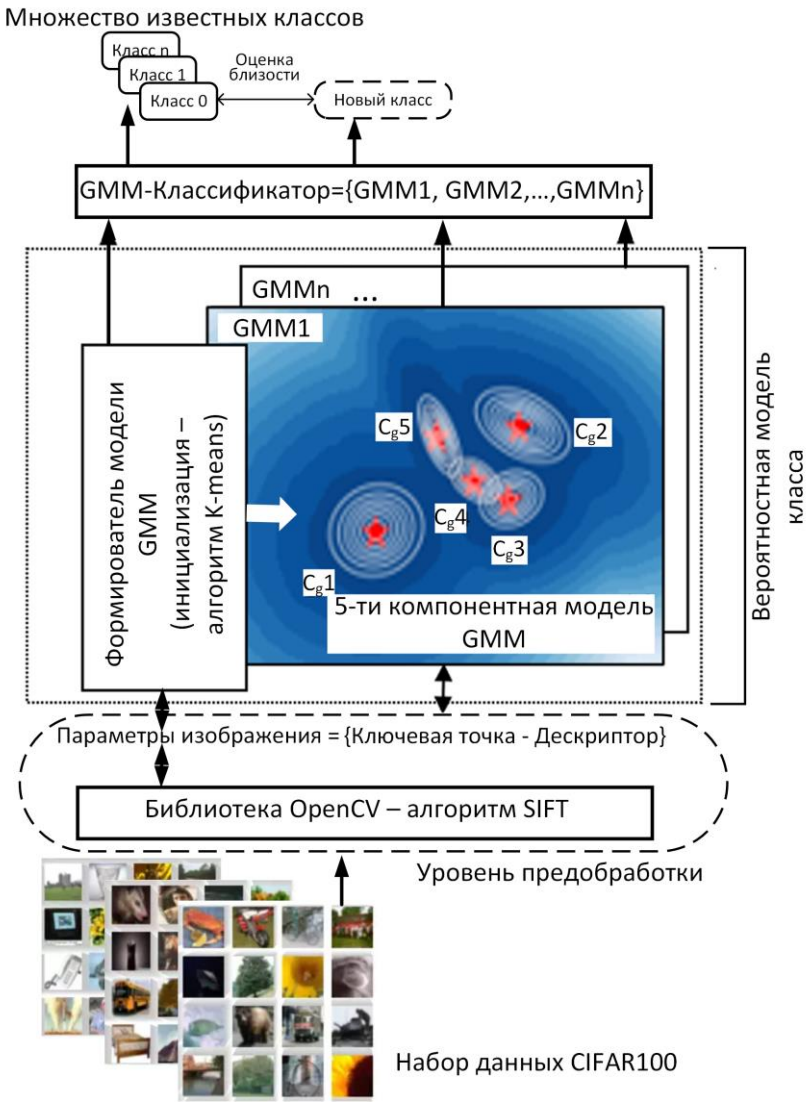

Рис. 2.1 Обобщенное представление предложенной модели GMM-классификатора изображений для узлов ЛМО и ГМО системы многоэлементной классификации

Из рисунка видно, что на уровне предобработки метод SIFT для каждого изображения из набора данных формирует его параметрическую модель – множество параметров вида «key_point – descriptor» (ключевая (особая) точка изображения – дескриптор ключевой точки).

Получаемое параметрическое пространство является основой для вероятностной модели класса изображения.

Получение этой модели является двухэтапным:

1. Инициализация модели – выделение центров кластеров параметров с использованием метода K-средних (K-means).

2. Формирование GMM-модели класса.

В ее основе лежит предположение о возможности генерации всех ключевых точек параметрического пространства смесью конечного числа гауссовых распределений.

GMM-модель обучается со следующими гиперпараметрами:

1. $C_g n$ – число используемых компонентов GMM.

2. Тип матрицы ковариации каждого $C_g n$ из следующего множества типов:

- связанная (tied) – все $C_g n$ имеют одну и ту же матрицу ковариации;

- диагональная (diagonal) – каждый $C_g n$ имеет собственную диагональную матрицу ковариации;

- сферическая (spherical) – каждый $C_g n$ имеет единственную дисперсию;

- полная (full) – каждый $C_g n$ имеет собственную общую матрицу ковариации.

3. Параметр неотрицательная регуляризация (non-negative regularisation), который добавляется к диагонали матрицы ковариации и гарантирует положительность всех матриц ковариации.

Таким образом, параметрическое пространство изображения преобразуется в множество параметрических пространств GMM-моделей, определяющих вероятностную модель класса изображения – $\{GMM_1, GMM_2, ..., GMM_n\}$.

Для сокращения размерности этого пространства выбирается GMM-модель GMM_i – представитель класса. Для решения задачи такого выбора для каждого элемента множества $\{GMM_1, GMM_2, ..., GMM_n\}$ рассчитывается байесовский информационный критерий (BIC) [66]:

$$BIC = -2 \cdot \ln(L) + k \cdot \ln(n), \qquad (46)$$

где L – максимизированное значение функции правдоподобия GMM-модели (рассматривается далее в п. 2.2.1), k – количество параметров GMM-модели, а n – размер выборки (количество наблюдений).

Модель GMM_i с наименьшим значением BIC и выбирается в качестве символического класса.

Множество GMM-моделей представителей классов формируют пространство классов предлагаемого многоэлементного GMM-классификатора.

При получении модели GMM_j с параметрами, не совпадающими с параметрами известных классов (используется принцип близости параметров), такая модель считается моделью нового или ненаблюдаемого класса.

Разработка алгоритмического решения предложенной модели многоэлементного GMM-классификатора представлена в п. 2.3.

Решение задачи программной реализации предложенной модели многоэлементного GMM-классификатора изображений в составе структуры программного комплекса системы многоэлементной классификации, а также особенности ее использования в сравнительном имитационном эксперименте представлены в пп. 4.4 и 4.9.

2.3 Исследование особенностей методов нахождения оценок максимального правдоподобия параметров модели гауссовой смеси распределений

2.3.1 Оценка параметров максимального правдоподобия

В общем случае задачей, связанной с получением параметров скрытых классов ЛМО с использованием GMM-модели является оценивание ее параметров λ, которые наилучшим образом соответствуют распределению обучающих векторов признаков скрытых классов.

В [68] выполнено детальное исследование методов оценки параметров GMM, среди которых наиболее известным и, в некотором роде, устоявшимся является метод оценивания максимального правдоподобия (ML - Maximum Likelihood). Цель метода максимального правдоподобия заключается в поиске параметров модели, которые максимизируют правдоподобие модели GMM с учетом обучающих данных.

Так, для последовательности R обучающих векторов $X=\{x_1,x_2,...,x_R\}$ правдоподобие модели GMM, с учетом предположения о независимости векторов, определяется выражением:

$$p(X|\lambda)=\prod_{t=1}^{T} p(x_t|\lambda) \quad (47)$$

Очевидно, что выражение 47 является нелинейной функцией параметров λ, что означает невозможность получения аппроксимации аналитическим путем. Однако, существуют подходы к оцениванию параметров максимального правдоподобия, основанные на итеративных процедурах.

В общем случае такие итеративные подходы основаны на том, что, начиная с начальной модели λ, выполняют оценивание новой модели $\bar{\lambda}$, такой что выполняется условие $p(X|\bar{\lambda}) \geq p(X|\lambda)$. Далее на следующей итерации новая модель $\bar{\lambda}$ становится начальной моделью. Этот процесс повторяется до тех пор, пока не будет достигнут некоторый порог сходимости. При этом начальная модель λ обычно получается использованием некоторого варианта бинарной классификации.

В подобных итеративных методах параметры модели GMM (выражение 49) задаются следующими выражениями: веса смеси (выражение 48), значения параметров векторов (выражение 49) и ковариационные матрицы (выражение 50)

$$\bar{w}_i=\frac{1}{R}\sum_{r=1}^{R} P(i|x_r,\lambda) \quad (48)$$

$$\bar{\mu}_i=\frac{\sum_{r=1}^{R} P(i|x_r,\lambda)x_r}{\sum_{r=1}^{R} P(i|x_r,\lambda)} \quad (49)$$

$$\bar{\Sigma}_i=\bar{\sigma}_i^2=\frac{\sum_{r=1}^{R} P(i|x_r,\lambda)x^2{}_r}{\sum_{r=1}^{R} P(i|x_r,\lambda)}-\bar{\mu}_i^2 \quad (50)$$

Апостериорная вероятность для i-го компонента модели определяется выражением:

$$P(i|x_r,\lambda) = \frac{w_i g(x_r|\mu_i,\Sigma_i)}{\sum_{i=1}^{R} w_i g(x_r|\mu_i,\Sigma_i)} \quad (51)$$

В исследовании был проведен анализ и выбор конкретного варианта итеративной процедуры для решения задачи разделения параметров модели GMM с целью получения параметров скрытых классов модели ЛМО.

2.3.2 Алгоритм «Ожидание-максимизация» - EM-алгоритм

Этот алгоритм сочетает статистическую методологию с алгоритмическим подходом и является широко используемым методом при наличии в множестве неполных данных. Он увеличивает связь между пропущенными данными и неизвестными параметрами модели данных. Основан на принципе дуальности, связанном с тем, что, когда известны параметры модели, можно сделать оценки для пропущенных значений и наоборот [68]. Относится к классу итеративных алгоритмов. При правильном выборе пространства данных, EM-алгоритм эффективно оценивает пропущенные значения данных.

Исходя из названия EM-алгоритм состоит из двух шагов:

1. Шаг условного ожидания (E (expectation)-шаг).

2. Шаг максимизации (M (maximization)-шаг).

На E-шаге вычисляются условные ожидания отсутствующих данных с учетом наблюдаемых данных и оценок параметров модели GMM – Θ-параметров. На M-шаге находятся оценки Θ-параметров для максимизации полно-логарифмической функции правдоподобия данных из E-шага. Эти шаги повторяются до тех пор, пока итерации не сойдутся. То есть, EM-алгоритм чередует шаги E и M для обновления некоторой оценки Θ_n неизвестных Θ-параметров на каждой итерации. Условные ожидания отсутствующих данных с учетом наблюдаемых данных и оценок параметров модели вычисляются на E-шаге с помощью выражения:

$$Q\left(\Theta_0|\Theta_n\right)=E_{Z|x,\Theta_n}\left[\log L(\Theta,x,z)\right], \qquad (52)$$

где $L(\Theta,x,z)$ – функция правдоподобия, Θ – вектор параметров, Θ_n – оценка параметров модели GMM, x – наблюдаемые данные (ГД – гибридные данные), z – пропущенные данные (ЛКД – локальные конфиденциальные данные).

На М-шаге, с целью максимизации полно-логарифмической функции правдоподобия данных из Е-шага, Θ-параметры вычисляются с помощью выражения:

$$\Theta^*=\arg_\Theta \max Q\left(\Theta|\Theta_n\right) \qquad (53)$$

На рисунке 2.2 приведена обобщенная схема EM-алгоритма.

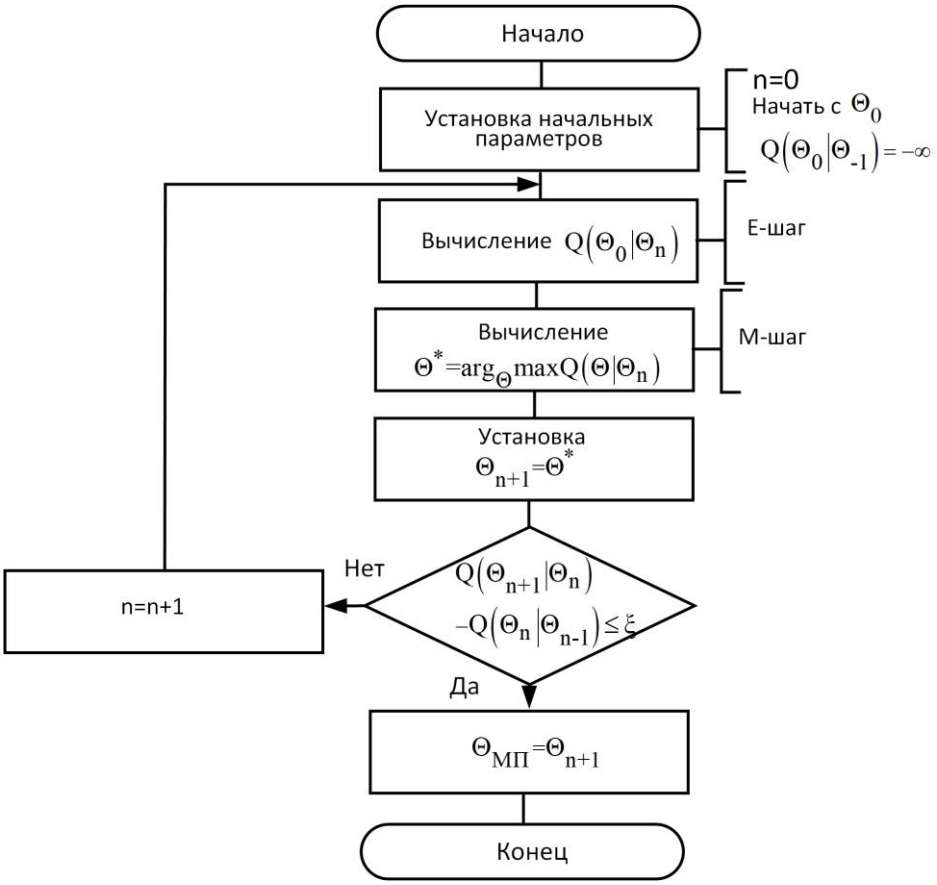

Рис. 2.2 Обобщенная схема EM-алгоритма

2.3.3 Метод оценивания апостериорного максимума (MAP - Maximum A posteriori Probability)

Альтернативным методом оценивания параметров GMM является метод апостериорного максимума (MAP) [66]. Распространенной областью его применения являются задачи распознавания образов, где для адаптации предшествующей модели ГМО используются ограниченные маркированные обучающие данные. Как и EM-алгоритм метод MAP является двухэтапным итерационным процессом.

Первый шаг идентичен шагу «Ожидание» EM-алгоритма, на котором оценки обучающих данных вычисляются для каждой смеси в предыдущей модели GMM. Однако, в отличие от второго шага EM-алгоритма (максимизация), эти обновленные статистические оценки объединяются с оценками из предыдущих параметров смеси с использованием специально введенного коэффициента смешивания, зависящего от данных.

Коэффициент смешивания разработан таким образом, что смеси с большим количеством новых данных больше полагаются на новую достаточную статистику для окончательной оценки параметров, а смеси с малым количеством новых данных больше полагаются на старую достаточную статистику для окончательной оценки параметров.

Таким образом, учитывая параметры предыдущей модели GMM и обучающие векторы $X=\{x_1,x_2,...,x_R\}$, определяется их вероятностное выравнивание в компоненты предварительной смеси. Это выполняется вычислением вероятности $P\left(i\middle|x_t,\lambda_{априори}\right)$ для i-й смеси. После этого, как и в EM-алгоритме, вычисляются параметры модели GMM: веса смеси (выражение 54), значения параметров векторов (выражение 55) и ковариационные матрицы (выражение 56)

$$n_i = \sum_{r=1}^{R} P\left(i\middle|x_r,\lambda_{априори}\right) \qquad (54)$$

$$E_i(x)=\frac{1}{n_i}\sum_{r=1}^{R} P\left(i\middle|x_r,\lambda_{априори}\right)x_r \quad (55)$$

$$E_i(x^2)=\frac{1}{n_i}\sum_{r=1}^{R} P\left(i\middle|x_r,\lambda_{априори}\right)x^2{}_r \quad (56)$$

Как было указано выше данный шаг метода MAP соответствует шагу «Ожидание» EM-алгоритма.

Полученные статистики из обучающих данных используются для обновления предыдущих статистик i-й смеси, чтобы создать ее адаптированные параметры, представленные выражениями 57-59.

$$\hat{w}_i=\left[\frac{\alpha_i^w n_i}{R}+\left(1-\alpha_i^w\right)w_i\right]\gamma \quad (57)$$

$$\hat{\mu}_i=\alpha_i^m E_i(x)+\left(1-\alpha_i^m\right)\mu_i \quad (58)$$

$$\hat{\sigma}_i^2=\alpha_i^v E_i(x^2)+(1-\alpha_i^v)(\sigma_i^2+\mu_i^2)-\hat{\mu}_i^2 \quad (59)$$

Множество коэффициентов адаптации $\left\{\alpha_i^w,\alpha_i^m,\alpha_i^v\right\}$ для весов, средних и дисперсий соответственно определяют баланс между старыми и новыми оценками. Коэффициент масштабирования γ вычисляется по всем адаптированным весам смеси, чтобы гарантировать, что они в сумме равны единице. Для каждой смеси и каждого параметра в приведенных выше уравнениях используется зависящий от данных коэффициент адаптации $\alpha_i^\rho,\rho\in\{w,m,v\}$, определяемый выражением:

$$\alpha_i^\rho=\frac{n_i}{n_i+r^\rho} \quad (60)$$

Исходя из анализа рассмотренных методов нахождения оценок максимального правдоподобия параметров модели гауссовой смеси распределений в качестве базового метода в разрабатываемом алгоритме было приято использовать EM-алгоритм. Его выбор обусловлен:

- простотой алгоритмической реализации E и M шагов;

- особенностями данных ЛКД, не в полной мере позволяющими получить достаточную апостериорную статистику, необходимую для второго шага метода MAP.

2.4 Разработка алгоритма получения значений оценок вероятностной функции ненаблюдаемых классов для многоэлементного классификатора, функционирующего в условиях неполноты классов локальных классификаторов

Исходя из обоснованно выбранного EM-алгоритма, как базового метода нахождения оценок максимального правдоподобия параметров модели GMM, которой представляются классы, к которым относятся данные ЛКМ множества моделей ЛМО системы федеративного обучения, в исследовании предлагается следующий подход к реализации алгоритма многоэлементного классификатора, функционирующего в условиях неполноты классов локальных классификаторов (ЛМО):

1. На уровне локальных классификаторов (ЛМО) на основе данных ОДД и собственных данных ЛКД реализуется EM-алгоритм обеспечивающий получение вероятности отнесения ненаблюдаемых данных к тому или иному классу (схема на рисунке 2.2).

2. На уровне итогового многоэлементного классификатора (ГМО) реализуется цикл агрегирования параметров множества GMM-моделей локальных классификаторов с параметрами подмножества не наблюдаемых классов.

Таким образом, предлагаемый алгоритм представляет совокупность двух частных алгоритмов, реализуемых на узлах Worker (локальные классификаторы) и Master (итоговы многоэлементный классификатор) соответственно.

Схема разработанного обобщенного алгоритма представлена на рисунке 2.3.

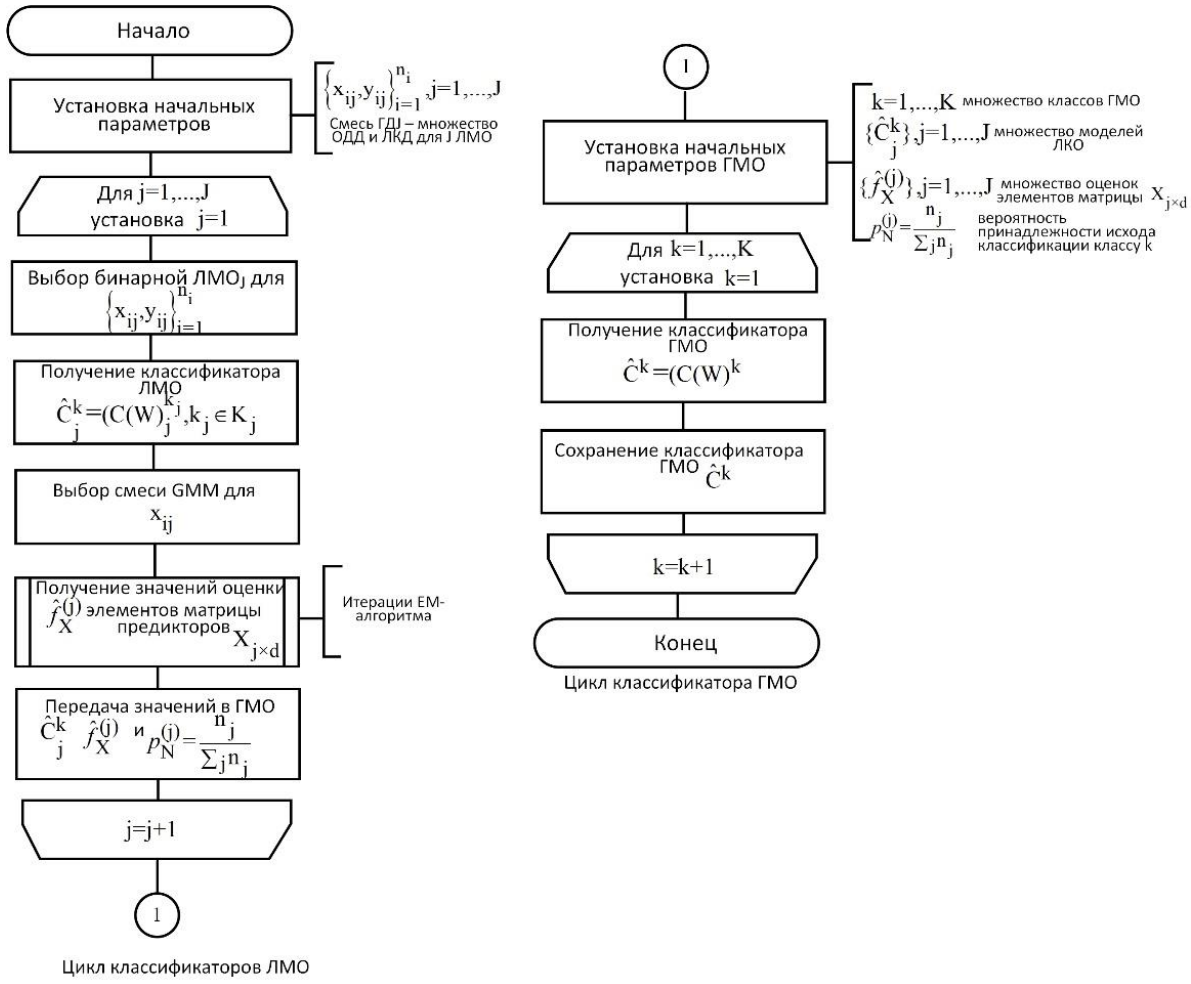

Рис. 2.3 Схема разработанного алгоритма получения значений оценок вероятностной функции не наблюдаемых классов системы многоэлементной классификации, функционирующей в условиях неполноты классов локальных классификаторов

Из рисунка 2.3 видно, что множество локальных классификаторов циклически реализуют процедуру формирования локальной модели GMM для всего множества наблюдаемых и не наблюдаемых классов. Этот цикл реализован итерациями EM-алгоритма, обеспечивающего получение вероятностных оценок GMM-моделей новых (ненаблюдаемых другими локальными классификаторами) классов.

На уровне итогового многоэлементного классификатора реализуется цикл агрегирования параметров моделей GMM множества локальных классификаторов с полученными значениями новых (ненаблюдаемых) классов данных ЛКД.

В общем случае подобная схема алгоритма соответствует традиционной схеме алгоритма функционирования системы с ФМО (рисунок 1.2). При этом, включение в этап формирования классификаторов моделей ЛМО итераций ЕМ-алгоритма, обеспечивающего получение оценок вероятностных характеристик не наблюдаемых классов данных ЛКД, обеспечивает необходимый уровень полноты классов многоэлемментного классификатора.

Очевидно, что полученные в ходе реализации ЕМ-алгоритма оценки являются прогнозными значениями, что может оказывать влияние на точность многоэлементной классификации. Однако, в сравнении с многоэлементным классификатором, функционирующим в условиях неполноты классов, полученное значение точности модифицированного алгоритма будет превышать возможности традиционного алгоритма за счет формирования полной матрицы элементов.

2.4 Выводы по главе

В главе представлен разработанный алгоритм получения значений оценок вероятностной функции ненаблюдаемых классов для многоэлементного классификатора, функционирующего в условиях неполноты классов локальных классификаторов.

Выполнена конкретизация модели GMM-классификатора для многоэлементной классификации цифровых изображений.

Разработана схема этапов предварительной обработки цифровых изображений и формирования параметров GMM-модели для существующих и новых (потенциально ненаблюдаемых) классов изображений.

Обоснован критерий валидности параметров формируемых GMM-моделей классов на основе метода максимального правдоподобия.

Обосновано использование итерационной процедуры EM-алгоритма для получения этих параметров.

Таким образом, разработан алгоритм получения значений оценки вероятностной функции ненаблюдаемых классов моделей локальных классификаторов и формирования модели классификатора системы многоэлементной классификации с федеративным машинным обучением, обеспечивающие динамическое формирование подмножества моделей локальных классификаторов, использующих матрицы элементов (классов) с рядом ненаблюдаемых значений и синтез на их основе модели итогового классификатора. В их основе лежит представление оценок функции вероятности ненаблюдаемых классов гауссовской смесью распределений и использование итерационной процедуры получения их параметров на основе метода максимального правдоподобия и этапов EM-алгоритма, а также выбор тех из них, которые наиболее полно представляют распределение векторов признаков ненаблюдаемых классов по критерию близости параметров известных и новых (потенциально ненаблюдаемых) классов.

Глава 3. Разработка алгоритма децентрализованного управления обменом данными системы многоэлементной классификации, функционирующей в условиях неполноты классов локальных классификаторов

3.1 Исследование парадигм обмена данными в системах с федеративным машинным обучением

Как было рассмотрено в классификации систем с ФМО (п. 1.1.2) одним из их существенных классификационных признаков является парадигма обмена данными между их функциональными узлами. Являясь вариантом распределенных систем, выбор той или иной парадигмы обмена данными (или их комбинации) является важным фактором, определяющим функциональные характеристики системы.

Исследования источников, посвященных архитектурным решениям систем с ФМО и, в частности, систем многоэлементной классификации [14, 16, 17], позволил выявить две парадигмы взаимодействия узлов в процессе их функционирования:

1. Централизованная: агрегация ММО всегда выполняется на центральном узле (глобальном сервере), а результаты отправляются локальным узлам (клиентам). Взаимодействие в подобной схеме в основном является асимметричным, и при этом может быть синхронным или асинхронным. В обобщенном виде централизованная модель взаимодействия узлов системы ФМО представлена на рисунке 1.2. Детальное представление процесса обучения ММО по этой схеме представлено на рисунке 3.1.

Из рисунка 3.1 видно, что структурно такая система с ФМО содержит два вида узлов (Worker и Master), которые имеют строго определенные функции. Раунд МО состоит из четырех этапов. Раунды МО циклически повторяются до достижения моделью ГМО-new необходимых значений показателей эффективности решения задачи, для которой разрабатывалась система ФМО (регрессии, классификации,

прогнозирования, распознавания). Очевидно, что централизованная схема наиболее приемлема для cross-silo систем с ФМО.

2. Децентрализованная: параметры ММО совместно используются распределенным образом, как правило, посредством peer-to-peer взаимодействия с подмножеством одноранговых Worker-узлов. Рассматриваемыми проблемами подобной схемы являются: накладные расходы на взаимодействие, и сохранение консенсуса ММО. Детальное представление процесса обучения ММО по этой схеме представлено на рисунке 3.2.

Из рисунка 3.2 видно, что раунд МО в системе ФМО с децентрализованной схемой состоит из трех этапов (2-4 на рисунке). Первый этап, являющийся по сути инициализацией ЛМО на Worker-узлах, реализуется по централизованной схеме, путем выбора Worker-узла инициатора.

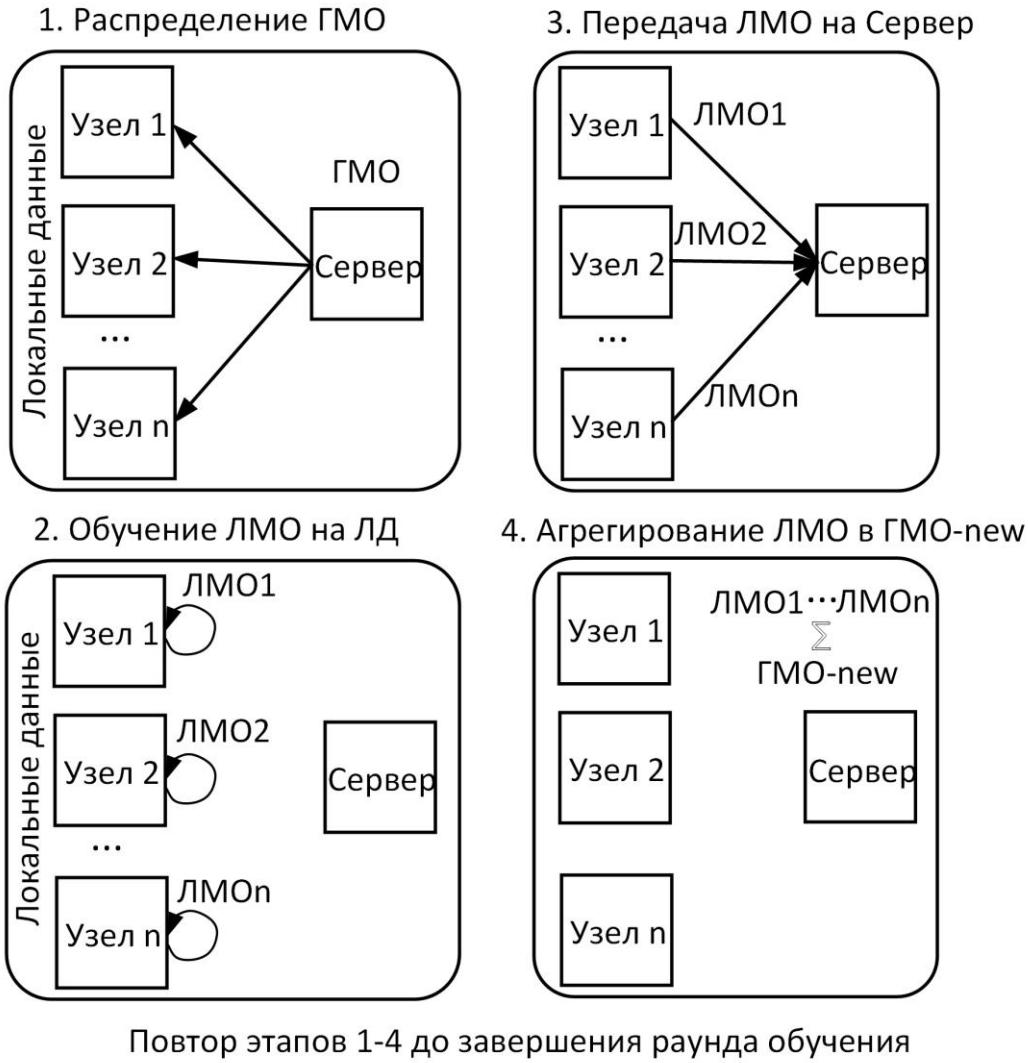

Рис. 3.1 Этапы раунда машинного обучения в централизованной схеме системы с ФМО

Также из рисунка 3.2 видно, что этап 3 предполагает взаимный информационный обмен Worker-узлов параметрами ЛМО по схеме peer-to-peer. Конкретные реализации P2P-протоколов при этом определяются в зависимости от ряда факторов, к которым относятся: пропускная способность каналов связи между Worker-узлами, схемы их доступности, наличие сбоев и отказов в их функционировании. Очевидно, что децентрализованная схема наиболее приемлема для cross-device систем с ФМО.

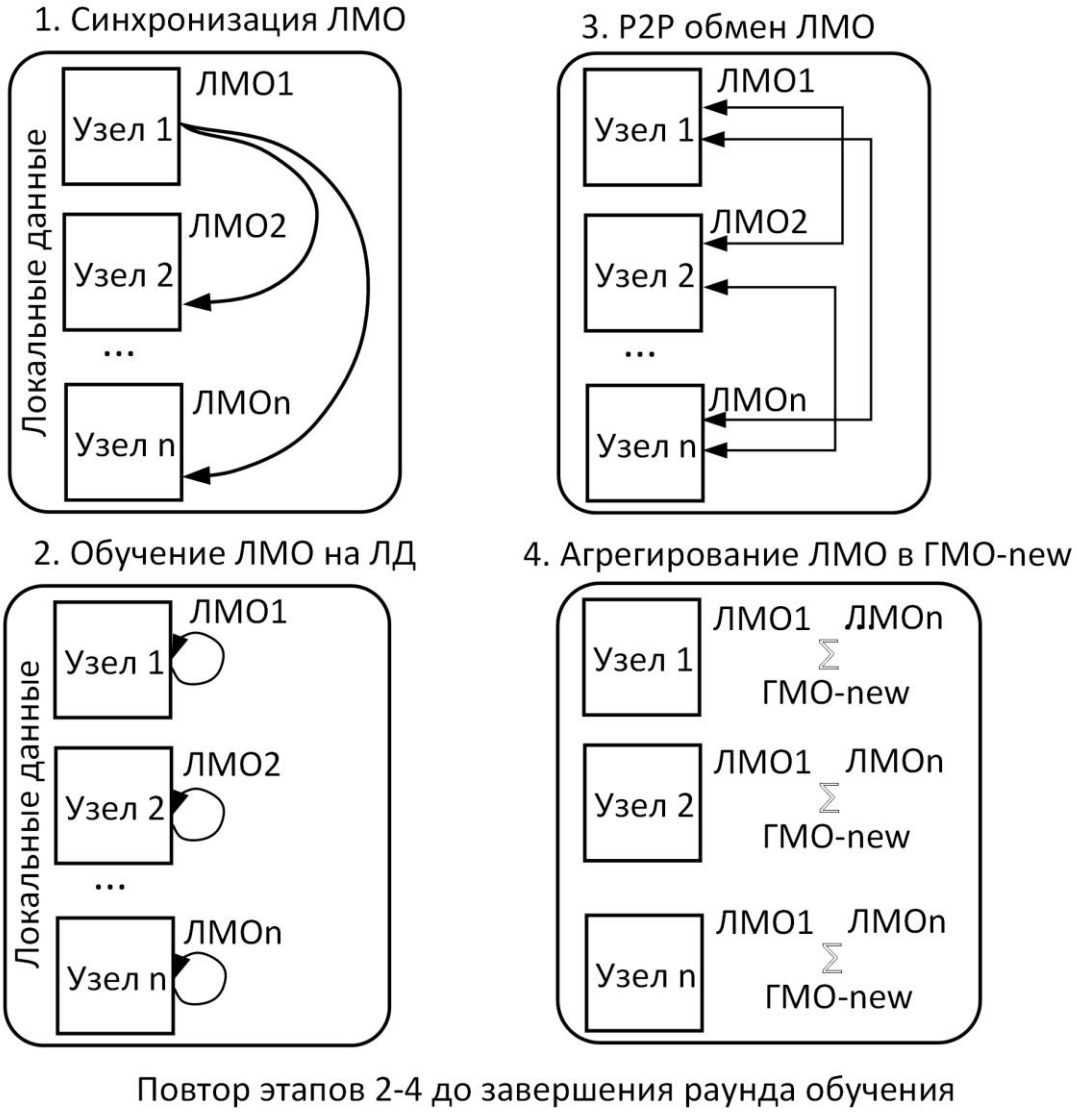

Рис. 3.2 Этапы раунда машинного обучения в децентрализованной схеме системы с ФМО

3.1.1. Реализации централизованной и децентрализованной схем систем с федеративным машинным обучением

Рассмотренные в п. 3.1 базовые схемы систем с ФМО, итерационно выполняющие раунд МО, являются основой для реализации конкретных алгоритмических схем, в частности, для систем многоэлементной классификации.

В [22] выполнено исчерпывающее исследование алгоритмов реализации схем ФМО, как централизованного, так и децентрализованного типов. Детальные

характеристики наиболее актуальных систем, реализующих ФМО на базе этих алгоритмов, приведены в таблице Приложения 1.

К наиболее распространенным и хорошо исследованным относятся алгоритмы Federated Averaging (FedAvg) [23] и Similarity- based Federated Learning (SimFL) [24].

Алгоритм (FedAvg) реализован по централизованной схеме раунда обучения (рисунок 3.1). В каждом раунде обучения сервер (Master-узел) сначала отправляет параметры текущей ГМО выбранному подмножеству Worker-узлов. Эти узлы обновляют параметры своих ЛМО на основе своих локальных наборов данных. Обновленные параметры подмножества ЛМО отправляются обратно на сервер. Сервер усредняет все полученные ЛМО, чтобы получить модель ГМО-new. Алгоритм FedAvg повторяет вышеуказанный процесс до достижения указанного количества раундов МО.

В отличие от FedAvg, алгоритм SimFL реализует децентрализованную схему (рисунок 3.2). В каждой итерации множество Worker-узлов сначала обновляют параметры своих ЛМО. Эти параметры отправляются подмножеству Worker-узлов для взаимного обновления параметров своих ЛМО. При достижении сходимости ЛМО итерационный процесс останавливается. Чтобы обеспечить справедливость и использовать данные от разных Worker-узлов, каждый из них выбирается для обновления модели примерно для одинакового количества раундов. SimFL повторяет указанное количество итераций и выводит окончательную модель ГМО-new.

3.2. Формальное представление централизованной схемы федеративного машинного обучения

Как было рассмотрено в п. 1.1.1 в системах с ФМО, вне зависимости от схемы их организации (рисунки 3.1 и 3.2) каждый Worker-узел реализует процесс МО одной и той же параметрической архитектуры локального классификатора на собственных наборах данных. Процесс МО при этом является асинхронным. Это означает, что как только определенное количество Worker-узлов завершает раунд T, обновленные веса в МО локального классификатора (или их градиенты) отправляются для агрегации либо на Master-узел (централизованная схема), либо на

подмножество Worker-узлов (децентрализованная схема). В силу особенностей разработанного в главе 2 алгоритма получения значений оценок вероятностной функции не наблюдаемых классов, далее будет рассмотрена централизованная схема на основе алгоритма FedAvg.

После этапа агрегации в ГМО-new новые веса на сервере перераспределяются Worker-узлам, и начинается новый раунд T+1 обучения МО локального классификатора.

Процесс итеративно продолжается до факта сходимости МО на всем множестве или определенном подмножество локальных классификаторов. После этого каждый Worker-узел выбирает вариант собственной МО из множества МО на всех предшествующих раундах путем оценивания метрик ее производительности на локальном тестовом наборе данных. При этом Worker-узел может выбрать не только вариант своей МО, но и многоэлементного классификатора МО, возвращаемую с Master-узла после процесса агрегации.

В наиболее общей форме алгоритм FedAvg пытается минимизировать некоторую глобальную функцию потерь f_{L}, которая может быть взвешенной комбинацией K локальных потерь $\left\{L_{\mathrm{k}}\right\}_{\mathrm{k}=1}^{\mathrm{K}}$, каждая из которых вычисляется на локальных данных k-го Worker-узла.

Следовательно, в общем случае задачу ФМО можно сформулировать как задачу поиска параметров некоторой модели ϕ (ГМО-new), которые минимизируют функцию f_{L}, учитывая некоторые локальные данные $\mathrm{X}_{\mathrm{k}} \in \mathrm{X}$, где X – комбинация всех локальных наборов данных.

$$\min f_{\mathrm{L}}(\mathrm{X},\phi), \tag{61}$$

где $f_{\mathrm{L}}(\mathrm{X},\phi)=\sum_{\mathrm{k}=1}^{\mathrm{K}} \mathrm{w}_{\mathrm{k}}\mathrm{L}_{\mathrm{k}}(\mathrm{X}_{\mathrm{k}},\phi)$, а $\mathrm{w}_{\mathrm{k}}>0$ – весовые коэффициенты для каждого k-го Worker-узла. При этом в схеме FedAvg Worker-узлы напрямую

не обмениваются локальными данными X_k, а разности w_k накапливаются и агрегируются на Master-узле.

В начале раунда t=1 k-й Worker-узел получает весовые коэффициенты модели ГМО $\phi^{(0)}$, применяет их к собственной ЛМО модели и получает разность $w_k = (\Delta\phi_k^{(t)}, n_k)$, где n_k – номер итерации обучения ЛМО. На Master-узле выполняется процедура обновления параметров ЛМО k-го Worker-узла $\phi_k^{(t)} = \phi_k^{(t-1)} + \Delta\phi_k^{(t)}$. Когда итерация n_k будет выполнена всем подмножеством Worker-узлов, на сервере выполняется процедура агрегации всех k ЛМО:

$$\text{ГМО-new}=\phi^{(t)} = \frac{1}{\sum_k n_k} \sum_k \left(n_k \cdot \phi_k^{(t)} \right) \tag{62}$$

В общем виде алгоритм схемы FedAvg представлен на рисунке 3.3.

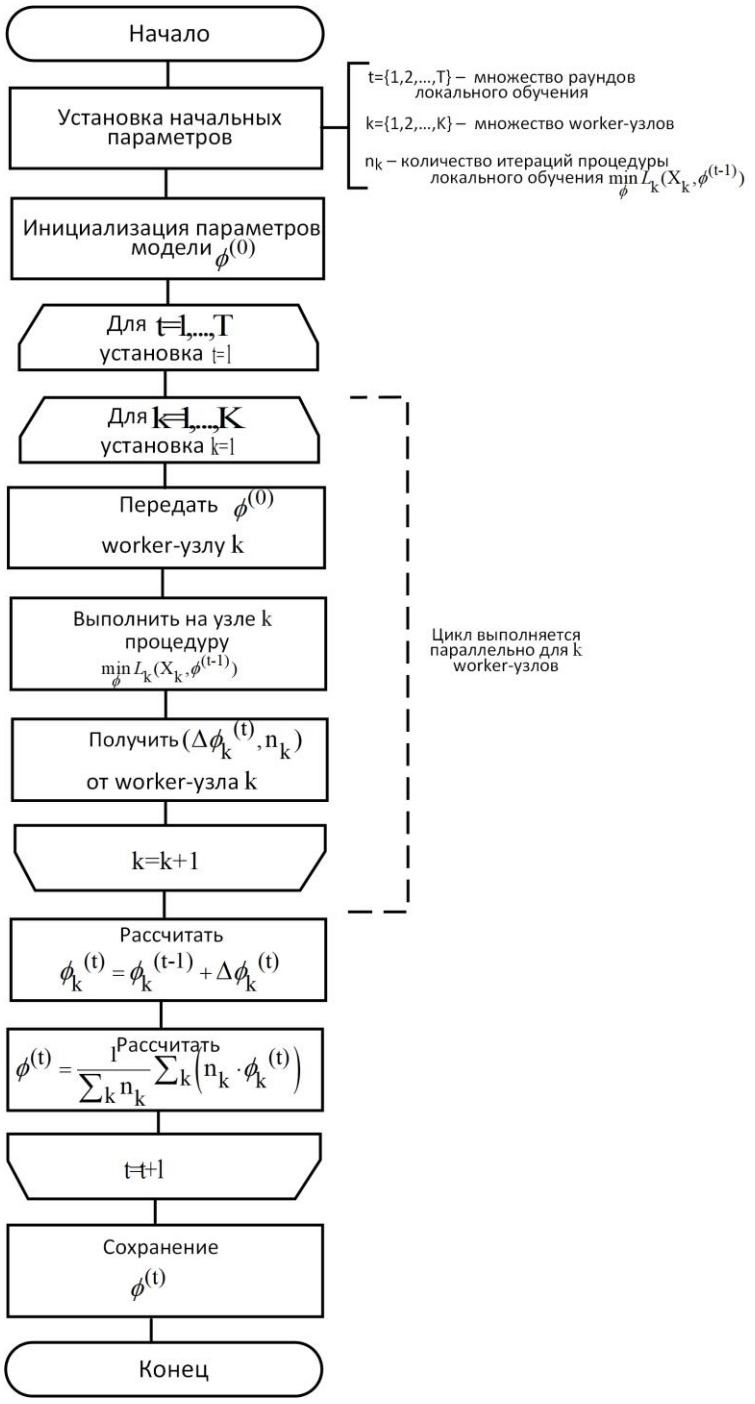

Рис. 3.3 Схема алгоритма FedAvg

3.3 Исследование и выбор метода агрегирования, применяемого в задачах бинаризации задач многоэлементной классификации и модельно-независимого машинного обучения

В качестве исследовательской основы рассмотрения методов агрегирования, применяемых в рассмотренных в п. 1.4 стратегиях решения задачи многоклассовой классификации, а также методах обеспечения дифференцированной конфиденциальности, основанных на модельно-независимом машинном обучении, использовались данные из источников [52-55].

Метод голосования (VOTE) (бинарное голосование, правило Max-Wins): Каждый классификатор ЛМО дает голос за предсказанный класс. Подсчитываются голоса, полученные каждым классом, и предсказывается класс с наибольшим количеством голосов:

$$K=\arg\max_{i=1,...,m}\sum_{1\le j\ne i\le m} s_{ij}, \qquad (63)$$

где $s_{ij}=1$, если $r_{ij}>r_{ji}r$, и 0 в противном случае.

Метод взвешенного голосования (WV): Каждый классификатор ЛМО голосует за оба класса. Вес голоса определяется уверенностью классификатора, предсказывающего класс. Класс с наибольшим значением суммы является окончательным выходным классом:

$$K=\arg\max_{i=1,...,m}\sum_{1\le j\ne i\le m} r_{ij}, \qquad (64)$$

Метод попарного связывания: Этот метод оценивает совместную вероятность для всех классов из попарных вероятностей классов классификаторов ЛМО.

Следовательно, когда $r_{ij}=P(k_i|k_i \text{ or } k_j)$, метод находит наилучшее приближение апостериорных вероятностей класса $\hat{\mathbf{p}}=\left(\hat{p}_1,...,\hat{p}_m\right)$ в соответствии

с выходными данными классификаторов. Класс K прогнозируется, как наибольшая апостериорная вероятность:

$$K=\arg\max_{i=1,\ldots,m}\hat{p}_i \qquad (65)$$

Для вычисления апостериорных вероятностей метод Кульбака – Лейблера - расстояние между r_{ij} и m_{ij} минимизируется:

$$l(p)=\sum_{1\le j\ne i\le m} n_{ij}r_{ij}\log\frac{r_{ij}}{\mu_{ij}}=\sum_{i<j} n_{ij}\left(r_{ij}\log\frac{r_{ij}}{\mu_{ij}}+(1-r_{ij})\log\frac{1-r_{ij}}{1-\mu_{ij}}\right), \qquad (66)$$

где $\mu_{ij}=\frac{p_i}{(p_i-p_j)}$, а n_{ij} – количество данных в выборках i-го и j-го классов.

Метод направленного на принятие решений ациклического графа (DDAG): конструирует корневой бинарный ациклический граф, где каждый узел связан со списком классов и бинарным классификатором. На каждом уровне классификатор различает два класса, а класс, который не предсказан, удаляется. Последний класс, оставшийся в списке, является конечным выходным классом.

Метод ценностного предпочтения (LVPC) [52, 53]: этот метод рассматривает матрицу оценок как нечеткое отношение предпочтений; на основе моделирования нечетких предпочтений исходное отношение разлагается на три новых отношения с разными значениями: строгое предпочтение, конфликт и незнание. Основан на следующем правиле принятия решения:

$$K=\arg\max_{i=1,\ldots,m}\sum_{1\le j\ne i\le m} P_{ij}+\frac{1}{2}C_{ij}+\frac{N_i}{N_i+N_j}I_{ij}, \qquad (67)$$

где

N_i – количество примеров из i-го класса в обучающих данных (и, следовательно, несмещенная оценка вероятности класса),

C_{ij} – степень конфликта (степень, в которой оба класса поддерживаются),

I_{ij} – степень незнания (степень, в которой ни один из классов не поддерживается),

P_{ij} и P_{ji} – соответственно, строгое предпочтение для i и j соответственно.

Степени предпочтения, уверенности и незнания вычисляются следующим образом:

$$P_{ij} = r_{ij} - \min(r_{ij}, r_{ji}) \quad (68)$$
$$P_{ji} = r_{ji} - \min(r_{ij}, r_{ji})$$
$$C_{ij} = \min(r_{ij}, r_{ji})$$
$$I_{ij} = 1 - \max(r_{ij}, r_{ji})$$

Метод предпочтения, основанный на критерии недоминирования (ND): в нем, как и в методе LVPC, матрица оценок рассматривается как нечеткое отношение предпочтения, которое в дальнейшем нормализуется. Затем вычисляется степень недоминирования (степень, в которой класс i не доминируется ни одним из оставшихся классов) и предсказывается класс с наибольшей степенью.

Метод двоичного дерева классификаторов (BTC): метод на основе двоичных деревьев (B-Trees). Заключается в том, чтобы сократить количество классификаторов и увеличить глобальную точность, используя некоторые бинарные классификаторы, которые различают два класса, чтобы различать другие классы одновременно. Дерево строится рекурсивно и аналогично подходу DDAG, каждый узел связывает бинарный классификатор и список классов. Но в этом случае решение классификатора может различать другие классы, а также пару классов,

используемых для обучения. Таким образом, в каждом узле, когда решение принято, из списка можно удалить более одного класса. Чтобы избежать ложных предположений, используется вероятность, когда примеры из класса находятся близко к границе дискриминанта, поэтому класс не может быть удален из списков на следующем уровне.

Метод вложенности OVO: этот метод разработан для решения неклассифицируемой области, полученной в методе голосования VOTE.

Использует стратегию голосования VOTE, но когда существуют примеры в неклассифицируемой области, конструируется новая система OVO, используя только примеры из этой области, чтобы сделать их классифицируемыми. Этот процесс выполняется до тех пор, пока в неклассифицируемой области вложенной OVO не останется ни одного примера.

Метод оценивания вероятности с помощью парного связывания: является видом метода парного связывания. Также оценивает апостериорные вероятности (**p**) каждого класса, начиная с парных вероятностей. Правило принятия решения эквивалентно методу парного связывания, а оптимизационная задача определяется, как:

$$\min_{\mathbf{p}} \sum_{i=1}^{m} \sum_{1 \le j \ne i \le m} \left(r_{ji} p_i - r_{ij} p_j \right)^2 \qquad (69)$$

при выполнении условия

$$\sum_{i=1}^{k} p_i = 1,\ p_i \ge 0,\ \forall i$$

Метод максимальной уверенности (MAX): схож с методом взвешенного голосования. Выходной класс берется из классификатора с наибольшим положительным ответом:

$$K = \arg \max_{i=1,\ldots,m} r_i \qquad (70)$$

Метод динамического упорядочивания: Используется наивный байесовский классификатор, который обучается (используя образцы из всех классов) вместе со всеми другими классификаторами. Этот новый классификатор устанавливает порядок, в котором классификаторы ЛМО выполняются для заданного шаблона. Затем экземпляр отправляется каждому классификатору ЛМО в этом порядке, пока не будет получен положительный ответ, который указывает на предсказанный класс. Это делается динамически для каждого примера. Таким образом, наивный байесовский классификатор избегает связей априори, а не полагаясь на степень уверенности, которую дают результаты классификаторов.

Исследование рассмотренных методов, а также требования формальной постановки задачи позволяют определить, что базовый метод голосования (VOTE) в общем случае является наиболее подходящим для синтеза модели классификатора ГМО, функционирующего в условиях неполноты классов.

3.4 Разработка функциональной схемы децентрализованного управления обменом данными классификатора системы многоэлементной классификации

Разработанный в п. 2.3 алгоритм итогового многоэлементного классификатора, функционирующего в условиях неполноты классов локальных классификаторов, определяет порядок получения обобщенного вида этого классификатора на основе агрегирования множества локальных классификаторов, с метками классов, представленных параметрами их GMM-моделей. Его особенностью является учет наличия в пространстве классов отдельных ЛМО новых классов, которые, возможно, являются не наблюдаемыми для остального или части подмножества ЛМО.

Учет этой особенности в рамках разрабатываемой системы классификации требует разработки этапов обобщенного алгоритма функционирования ее узлов: множества узлов Worker_node и центрального узла Master_node.

Очевидно, что этапу формирования итоговой модели должны предшествовать этапы:

- формирования ЛМО отдельными узлами Worker_node с поддержкой нахождения новых (потенциально ненаблюдаемых) классов;

- взаимного информационного согласования ЛМО множества Worker_node с учетом наличия ненаблюдаемых классов, обеспечивающим решение проблемы неполноты классов ЛМО.

Для поддержки указанных этапов в исследовании предлагается применение гибридной схемы взаимодействия узлов системы многоэлементной классификации с ФМО, включающей децентрализованную и централизованную компоненты, раунды которых рассмотрены в п. 3.1.

В общем виде принцип реализации такой гибридной схемы представлен на рисунке 3.4.

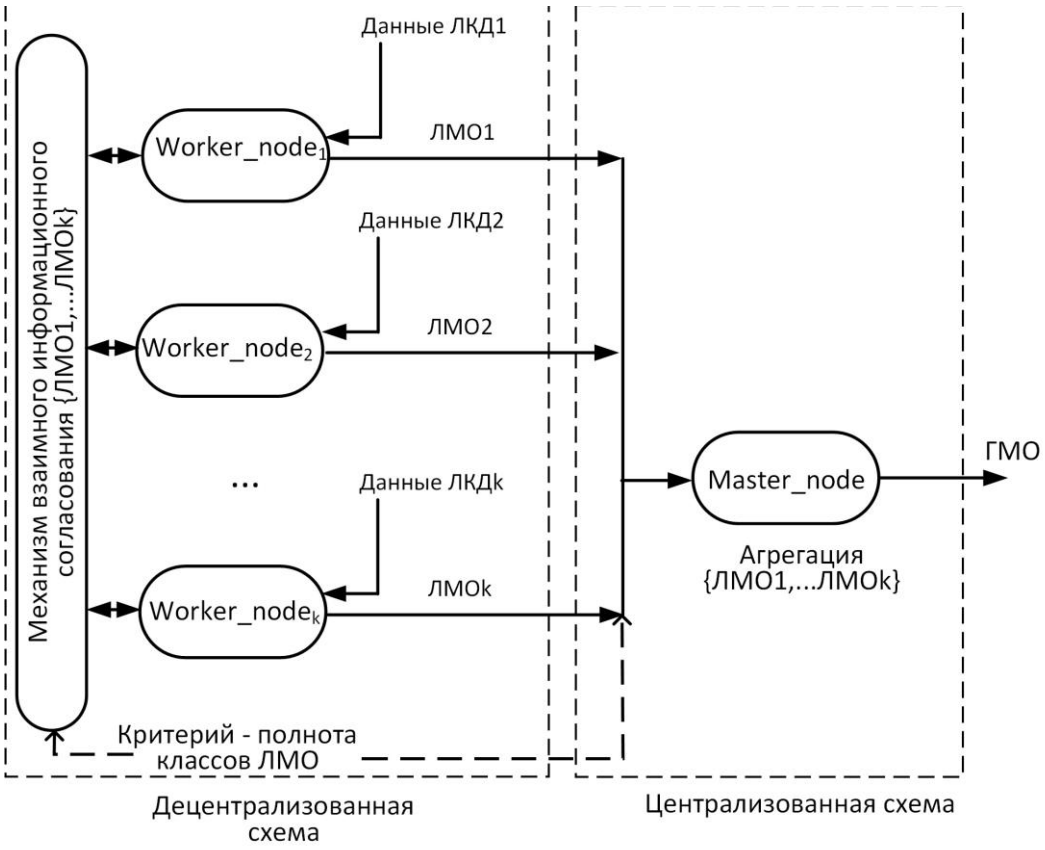

Рис. 3.4 Предлагаемая схема обмена данными узлов системы многоэлементной классификации с ФМО

Из рисунка видно, что основой децентрализованной составляющей предлагаемой схемы обмена данными является механизм взаимного информационного согласования (ВИС) множества узлов Worker_nodes.

В общем случае такой механизм базируется на одном или комбинации протоколов децентрализованного взаимодействия. Развернуты анализ таких протоколов сделан в [72].

Поскольку в процессе ВИС узлы Worker_node выполняют широковещательную рассылку параметров GMM-модели классов-кандидатов для отнесения к новому классу, в качестве основы механизма ВИС рассматривались протоколы децентрализованного Peer-to-Peer (P2P) взаимодействия, поддерживающие:

- режим широковещательной рассылки;
- формат данных – сообщения (message based protocols).

Указанным условиям соответствуют варианты протокола DHT [72].

3.5 Разработка этапов алгоритма децентрализованного обмена данными классификатора системы многоэлементной классификации

Выбор механизма ВИС позволил разработать этапы алгоритма взаимодействия узлов системы федеративного машинного обучения, функционирующей в условиях неполноты классов локальных классификаторов. Схема разработанного алгоритма представлена на рисунке 3.5.

Из рисунка 3.5 видно, что алгоритм является трехэтапным. Этапы выполняются последовательно. Переход к очередному этапу выполняется при полной сходимости алгоритма предыдущего этапа.

Этап 1 определяет автономное функционирование множества узлов Worker_node. В силу автономности его выполнения этап носит асинхронный характер и его временная сложность зависит от таких особенностей функционирования отдельных узлов Worker_node при формировании модели GMM-классификатора ЛМО как: объем входных данных, результат из предобработки (процедура SIFT), результат кластеризации полученного множества ключевых точек (метод K-Means).

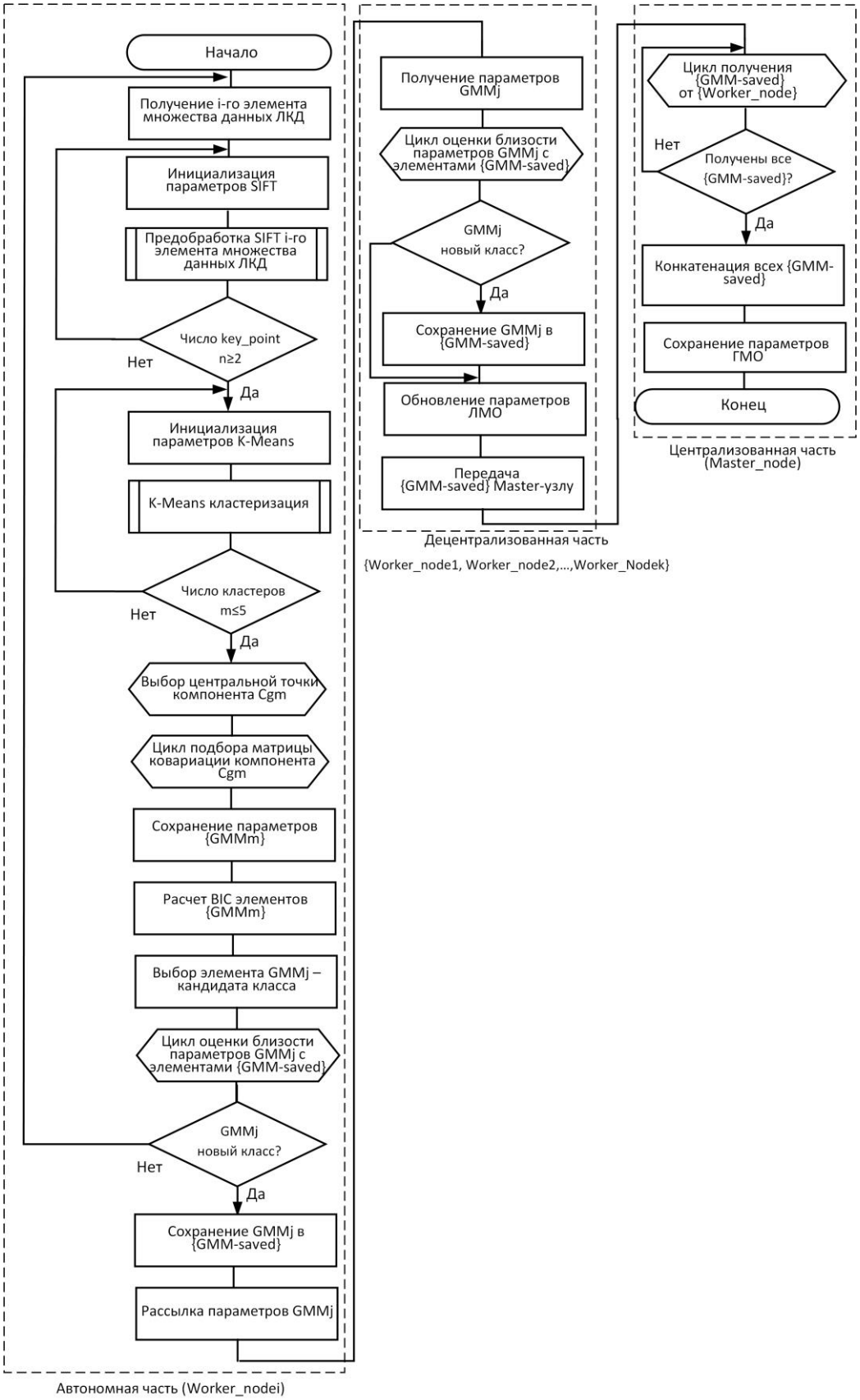

Рис. 3.5 Схема алгоритма децентрализованного обмена данными классификатора системы многоэлементной классификации

В целях снижения размерности признаковых пространств ключевых точек и компонентов Cg множества вероятностных GMM-моделей в алгоритме введены проверки ограничений размерности.

Так, алгоритм выполняет повторную предобработку с новыми параметрами нормализации при получении на предыдущем шаге предобработки числа ключевых точек <2 или >100, что дает возможность на этапе кластеризации получить меньшее количество компонентов Cg. При этом при получении значения Cg>5 этап кластеризации повторяется с новыми параметрами нормализации. Обоснование такого ограничения при расчете показателей GMM-модели подтверждается эмпирическими исследованиями, представленными в [68]. Результатом автономного этапа алгоритма является процесс рассылки подмножеством (в пределе – множеством) узлов Worker_node параметров некоторой j-й GMM-модели, являющейся кандидатом на новый класс (потенциально ненаблюдаемый другими узлами Worker_node).

Этап 2 реализует децентрализованное взаимодействие подмножества (или всего множества) узлов Worker_node в процессе их ВИС по отнесению подмножества разосланных GMM-моделей к классу ненаблюдаемых. С этой целью узлы worker_node циклично выполняют оценку близости параметров каждой полученной GMM-модели с параметрами подмножества моделей GMM-saved, включающего модели известных узлу Worker_node классов. В случае достижения близости параметров хотя бы с одной моделью подмножества GMM-saved, класс, определяемый полученной моделью, считается известным. В противном случае он относится к категории new и дополняет подмножество GMM-saved.

Этап ВИС-взаимодействия завершается, когда у каждого из взаимодействующих узлов Worker_node отсутствуют GMM-модель класса new. Это считается критерием полноты классов ЛМО (рисунок 3.1).

Этап 3 является классическим централизованным этапом раунда ФМО (рисунок 3.1). На этом этапе параметры множеств GMM-моделей всех Worker-узлов передаются Master-узлу, где выполняется их конкатенация. В дальнейшем обучение модели ГМО выполняется на указанной совокупности GMM-моделей.

Одной из необходимых задач при разработке алгоритмических решений является этап оценивания вычислительной сложности разработанных алгоритмов.

Как следует из [73] совокупная вычислительная сложность многоэтапных алгоритмов определяется предельной оценкой вычислительной сложности итерационных этапов. В рамках разработанных алгоритмов такими этапами являются:

- EM-алгоритм выбора параметров GMM-модели класса, реализуемый подмножеством Worker-узлов на подмножестве вероятностных моделей классов;

- алгоритм кластеризации K-Means, реализуемый на этапе инициализации расчета параметров вероятностных моделей классов и определяющий предельное значение множества элементов Cg.

Как указывается в [73], в основе метода K-Means лежит вариант алгоритма Ллойда, на каждой итерации требующий вычисления расстояния Махаланобиса, что определяет сложность $O(n^k)$, где n – число итераций, а k – количество k-средних в алгоритме Ллойда. На практике, однако, вычислительная сложность оценивается как O(n*k), поскольку алгоритм имеет тенденцию к останову сходимости через несколько десятков итераций.

В EM-алгоритме EM объекты назначаются не одному кластеру, а относительно всех кластеров. Теоретически это означает, что без какого-либо порога остановки EM-алгоритм будет бесконечно оптимизировать назначения кластера вплоть до бесконечной точности. Таким образом, сложность EM-алгоритма можно оценить O(n*k*i), где i – число итераций, эмпирически определенное исследователем.

3.6 Выводы по главе

В главе представлен разработанный алгоритм децентрализованного обмена данными классификатора системы многоэлементной классификации, функционирующей условиях неполноты классов локальных классификаторов.

В рамках разработки схема алгоритма:

- рассмотрены централизованная и децентрализованная схема реализации раундов ФМО;

- исследованы методы агрегирования моделей локальных классификаторов и, применительно к предложенному в главе 2 GMM-классификатору, обоснованно выбран метод голосования VOTE;

- обоснована трехэтапная схема его организации, включающая множество автономных этапов узлов Worker_node, децентрализованный этап их взаимного информационного согласования с достижением критерия полноты классов моделей ЛМО и централизованная часть, представляющая раунд агрегирования итоговой модели.

Таким образом, разработан алгоритм обмена данными узлов системы многоэлементной классификации, функционирующей в условиях неполных данных для принятия решения, основанный на трехэтапной гибридной (автономная-децентрализованная-централизованная) схеме функционирования и реализующий обобщенный цикл получения матриц элементов (классов) подмножества локальных классификаторов с учетом потенциально ненаблюдаемых классов.

ГЛАВА 4. РАЗРАБОТКА АРХИТЕКТУРЫ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ МНОГОЭЛЕМЕНТНОЙ КЛАССИФИКАЦИИ, ФУНКЦИОНИРУЮЩЕЙ В УСЛОВИЯХ НЕПОЛНОТЫ КЛАССОВ ЛОКАЛЬНЫХ КЛАССИФИКАТОРОВ

4.1 Разработка структуры программного комплекса системы многоэлементной классификации, функционирующей в условиях неполноты классов локальных классификаторов

4.1.1 Выбор и обоснование фреймворка федеративного машинного обучения

В настоящее время разработка программного обеспечения (ПО) систем ФМО для решения различных задач и специализированных для различных предметных областей ведется как в сфере коммерческих проектов, так и проектов с открытым исходным кодом [74-78].

В то время как коммерческие проекты систем ФМО обычно разрабатываются и оптимизируются под конкретное функциональное применение, проекты с открытым исходным кодом, в силу особенностей разработки, являются, с точки зрения функциональности, более универсальными и настраиваемыми для конкретного применения. В связи с этим в рамках исследования выбор фреймворка ФМО выполнялся среди проанализированного множества проектов с открытым исходным кодом.

В качестве альтернатив фреймворков ФМО были рассмотрены:

- Federated AI Technology Enabler (FATE) – разработчик компания Baidu [74];
- TensorFlow Federated (TFF) – разработчик компания Google [75];
- PySyft – разработчик проект с открытым исходным кодом OpenMined [76];
- PaddleFL – разработчик компания Baidu [77];
- FedML – разработчик Университет Южной Калифорнии [79].

В таблице 4.1 представлены обобщенные характеристики рассматриваемых фреймворков.

Таблица 4.1

Характеристики фреймворков ФМО на основе проектов с открытым исходным кодом

Характеристики		FATE	TFF	PySyft	PaddleFL	FedML
Поддерживаемые ОС	Windows			X	x	x
	Linux	x	x	X	x	x
	Mac OS	x	x	X	x	x
	Android					x
	iOS					x
Разделение данных	горизонтальное	x	x	X	x	x
	вертикальное				x	x
Модели МО	Нейронные сети	x	x	X	x	x
	Деревья решений	x				
	Логистическая регрессия	x	x	X	x	x
Механизм конфиденциальности	Дифференцированная конфиденциальность		x	X	x	x
	защищенные многосторонние вычисления	x		X	x	
Структура	централизованная	x	x	X	x	x
	децентрализованная	x	x	X	x	x
Вычислительная платформа	CPU	x	x	X	x	x
	GPU, TPU		x	X		x

Далее рассматриваются структурно-функциональные характеристики выбранных фреймворков.

4.1.1.1 Фреймворк FATE

FATE — это фреймворк ФМО промышленного уровня, разработанный компанией WeBank, который обеспечивает предоставление услуг ФМО для cross-silo систем. В основе FATE лежит IDE Python.

На рисунке 4.1 представлена функциональная схема модулей FATE.

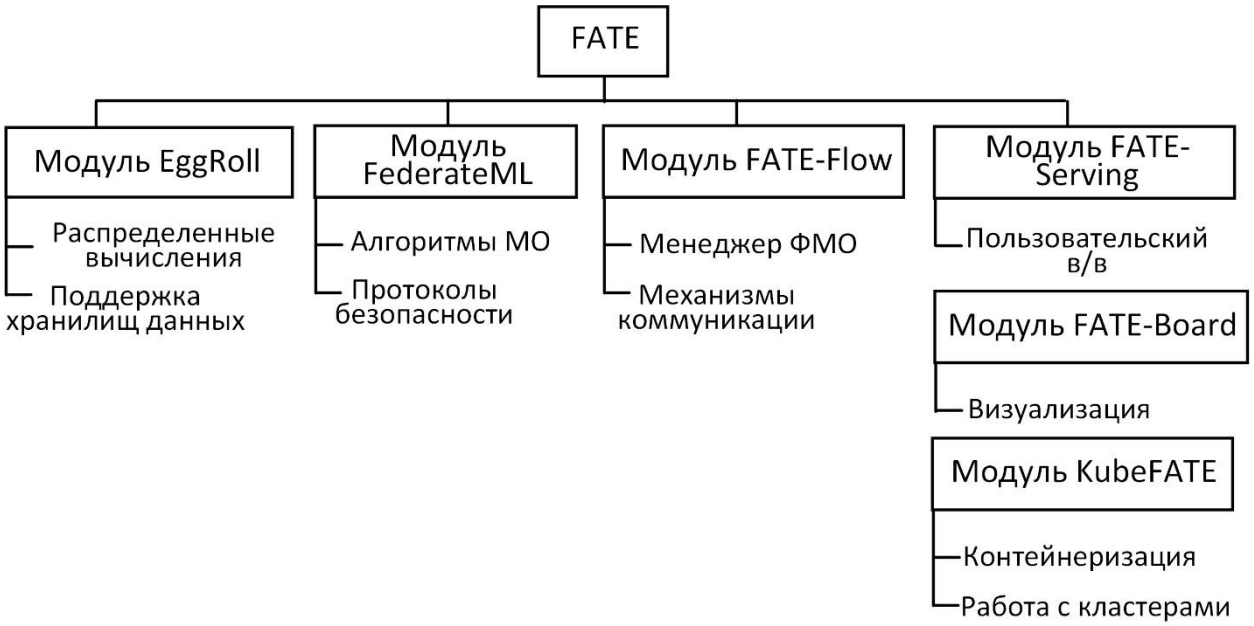

Рис. 4.1 Структура функциональных модулей FATE

Из рисунка 4.1 видно, что фреймворк FATE состоит из шести основных модулей: EggRoll, FederatedML, FATE-Flow, FATE-Serving, FATE-Board и KubeFATE.

Модуль EggRoll управляет распределенными вычислениями и хранением. Он предоставляет вычислительные API и API СХД для других модулей.

Модуль FederatedML включает в себя базовые алгоритмы ФМО и протоколы обеспечения конфиденциальности.

В настоящее время он поддерживает обучение многих видов моделей машинного обучения как в горизонтальной, так и в вертикальной федеративной настройке, включая нейронные сети, деревья решений и логистическую регрессию.

С точки зрения конфиденциальности FATE поддерживает только безопасные многосторонние вычисления и гомоморфное шифрование.

Модуль FATE-Flow – это платформа, на которой пользователи могут организовать собственную среду ФМО, которая будет включать: предварительную обработку данных, алгоритмы ФМО, механизм оценивания, управление ММО.

Модуль FATE-Serving предоставляет услуги вывода для пользователей. Он поддерживает загрузку ММО и проведение онлайн-вывода по ним.

Модуль FATE-Board – это инструмент визуализации. Он предоставляет визуальный способ отслеживания выполнения процесса ФМО и производительности ММО.

Модуль KubeFATE обеспечивает развертывание FATE на кластерах с помощью систем контейнеризации Docker или Kubernetes.

4.1.1.2 TensorFlow Federated (TFF)

Фреймворк TFF, разработанный компанией Google, является надстройкой ФМО над фреймворком TensorFlow. Он предоставлен пакетом для IDE Python. На рисунке 4.2 представлена функциональная схема модулей TFF.

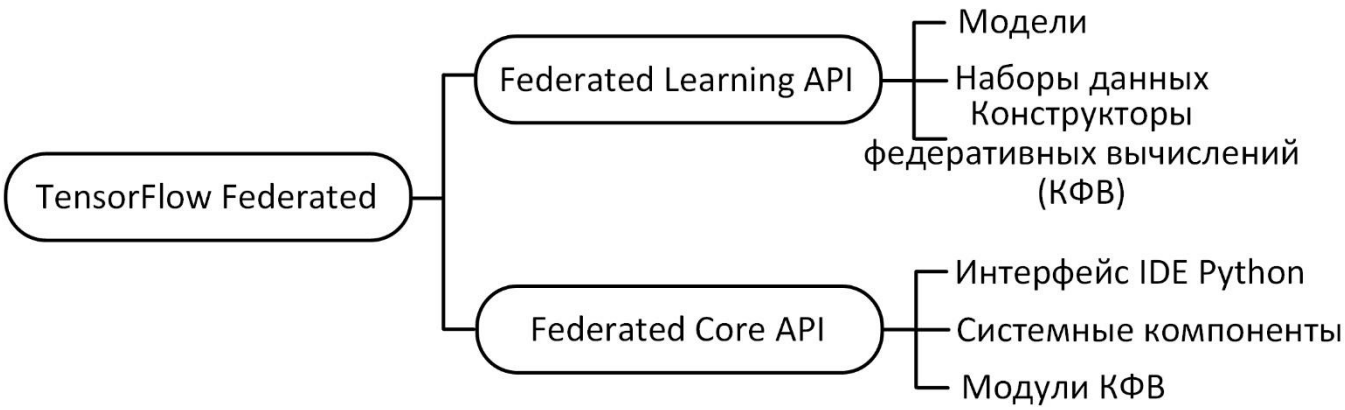

Рис. 4.2 Структура функциональных модулей TFF

Из рисунка 4.2 видно, что фреймворк TFF включает в себя модули:

FL API – который предоставляет высокоуровневые интерфейсы и, в свою очередь, состоит из:

- поддерживаемых ММО;
- наборов данных для процесса ФМО;
- конструкторов ФМО, основанные на алгоритме FedAvg.

FC API – который позволяет пользователям самостоятельно формировать ММО или импортировать ММО из фреймворка Keras.

Помимо интерфейсов высокого уровня, FC API также включает интерфейсы более низкого уровня в качестве основы процесса ФМО, с помощью которых разработчики могут реализовывать собственные функции и интерфейсы ФМО.

Он поддерживает такие алгоритмы как федеративная сумма, федеративная редукция и федеративная трансляция. Разработчики могут определять свои собственные операторы для реализации алгоритма ФМО.

4.1.1.3 Фреймворк PySyft

PySyft представляет собой библиотеку IDE Python, которая предоставляет интерфейсы для разработчиков для реализации алгоритмов ФМО.

Его особенностью является возможность развертывания структуры ФМО на одном вычислительном узле, где связь между узлами ФМО осуществляется через API web-сокетов. Из таблицы 4.1 видно, что фреймворк PySyft предоставляет развитые механизмы конфиденциальности, включая безопасные многосторонние вычисления и дифференциальную конфиденциальность.

4.1.1.4 Фреймворк PaddleFL

PaddleFL – это фреймворк на основе проекта свободного ПО PaddlePaddle. Это платформа глубокого МО, разработанная компанией Baidu. Фреймворк реализован на языке C++ и IDE Python.

На рисунке 4.3 представлена функциональная схема модулей PaddleFL.

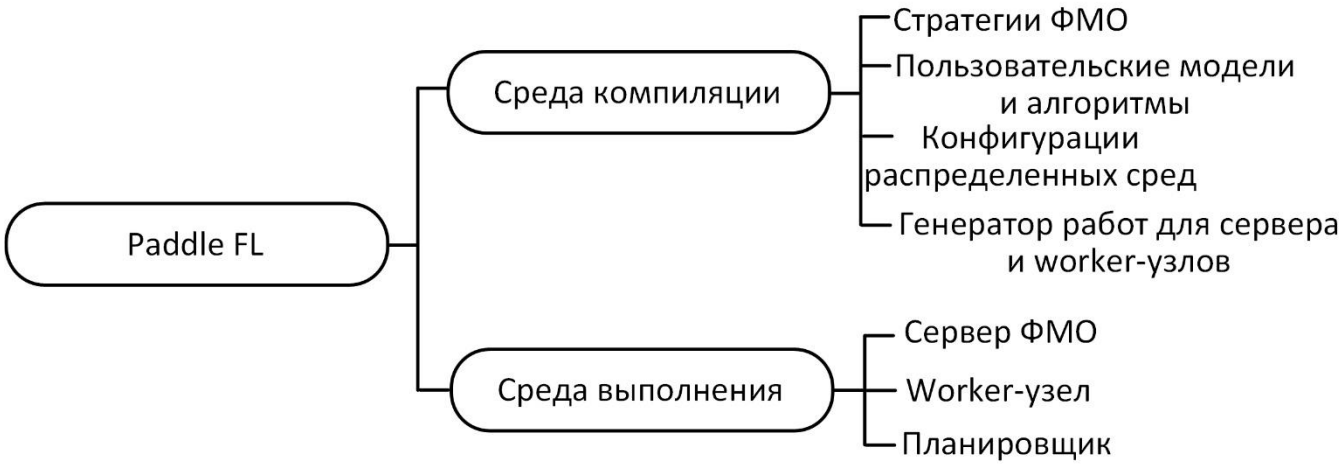

Рис. 4.3 Структура функциональных модулей PaddleFL

Из рисунка 4.3 видно, что модуль Среда компиляции поддерживает четыре компонента, определяемые пользователем фреймворка:

- ММО и алгоритмы МО;
- стратегии процесса ФМО;
- конфигурацию распределенной среды ФМО;
- генератор заданий (работ) для сервера системы ФМО централизованного типа.

Модуль Среда выполнения поддерживает базовые компоненты системы ФМО централизованного типа:

- сервер;
- модуль worker-узла;
- планировщик процесса ФМО.

4.1.1.5 FedML

Фреймворк FedML, являясь исследовательским проектом, поддерживает не только среду ФМО, но репозиторий тестов (benchmarks) для систем ФМО.

На рисунке 4.4 представлена функциональная схема модулей FedML.

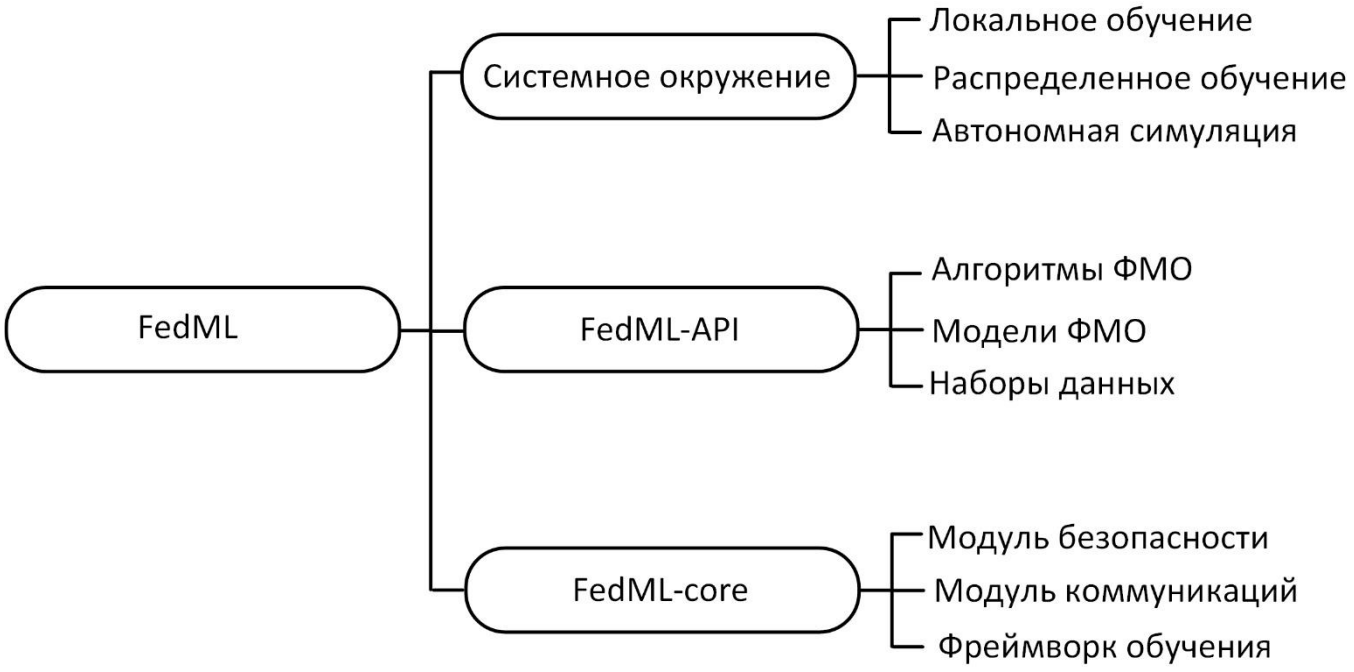

Рис. 4.4 Структура функциональных модулей FedML

Из рисунка 4.4 видно, что в низкоуровневом ядре FedML-core реализованы сам фреймворк МО, а также модули обеспечения конфиденциальности и коммуникаций узлов систем ФМО.

Уровень FedML-API базируется на низкоуровневом ядре и поддерживает работу МО, алгоритмов ФМО, а также наборы данных для них. Дополнительно в этом модуле реализованы алгоритмы тестирования производительности систем ФМО.

FedML поддерживает три вычислительные парадигмы:

- автономное моделирование;

- распределенные вычисления;

- поддержка процесса МО на устройстве, что обеспечивает среду моделирования для широкого спектра приложений.

Анализ функциональных возможностей представленных проектов систем ФМО позволил обоснованно выбрать в качестве фреймворка, применяемого в настоящем исследовании, среду FedML, как наиболее развитую функционально с точки зрения возможностей тестирования производительности предлагаемого решения.

4.2 Выбор и обоснование фреймворков GMM-классификатора и обеспечения дифференцированной конфиденциальности APL

Программная реализация классификаторов на основе модели GMM представлена рядом исследовательских и коммерческих фреймворков разной степени функциональности, включая возможности интеграции с библиотекой FedML [98-100]. В результате анализа их функциональных возможностей был выбран фреймворк Scikit-Learn [100] в силу его развитой модельной функциональности и возможности интеграции с другими Python-проектами с использованием набора функций ScikitAPI.

Как было рассмотрено в главе 1, наиболее приемлемыми методом дифференцированной конфиденциальности, применимым для многоэлементной

классификации, с точки зрения вопроса баланса производительности распределенной системы классификации и точности результатов ее функционирования, является метод независимого конфиденциального обучения (APL), общий теоретический подход которого представлен в [79].

Анализ исследований применения метода APL позволил выявить две его практические реализации:

1. Модельно-независимое конфиденциальное обучение (MAPL – Model-Agnostic Private Learning) [80]. Он основан на предварительном использовании специализированного (закрытого для множества ЛМО) фреймворка для разметки векторов открытых признаков. Далее эти заново размеченные открытые данные используются для обучения открытых ЛМО для получения, в конечном итоге, итоговой МО. Авторы метода доказывают, что, поскольку ограничения на приватность связанные с открытыми данными отсутствуют, в целом схема СФМО с точки зрения дифференцированной приватности остается закрытой. Однако, представленное ПО метода MAPL носит больше исследовательский характер, необходимый для доказательства теоретической базы метода, что не позволяет его интегрировать в существующими реализациями систем с ФМО.

2. Конфиденциальное объединение ансамблей учителей (PATE - Private Aggregation of Teacher Ensembles), рассмотренный в п. 1.5.

Таким образом, в качестве базового модуля дифференцированной конфиденциальности для разрабатываемого программного комплекса системы многоэлементной классификации предлагается использование фреймворка PATE.

4.3 Разработка структуры программного комплекса распределенной системы многоэлементной классификации, функционирующей в условиях неполноты классов локальных классификаторов

Представленный в пп. 4.1 и 4.2 выбор фреймворков для реализации как функций базовых узлов разрабатываемой системы многоэлементной классификации (множество Worker_Node узлов и узел Master_Node), так и для реализации функций дифференцированной конфиденциальности, обеспечивающей поддержку процесса обучения ЛМО в условиях неполноты классов, позволил разработать структуру программного комплекса, поддерживающего функции указанных узлов

разрабатываемой системы. Его обобщенная структурная схема представлена на рисунке 4.5. Толщиной линии выделены функциональные блоки решения, предлагаемого в рамках проводимого исследования.

Из рисунка 4.5 видно, что предлагаемая структура системы базируется на комбинированной схеме централизованной и децентрализованной схемах реализации раунда ФМО (глава 1, рисунки 1.7, 1.8), где функции менеджера реализует узел Master_Node, а функции множества агентов реализует узел Worker_Node. Их программной основой выступает блок FedML-Core (рисунок 4.4), поддерживающий фреймворк обучения ЛМО и ГМО для узлов Worker_Node и Master_Node соответственно, а также модуль коммуникаций, реализующий схемы «точка-многоточка» для централизованной модели «менеджер-агенты» и «точка-точка» для децентрализованной модели «агент-агент».

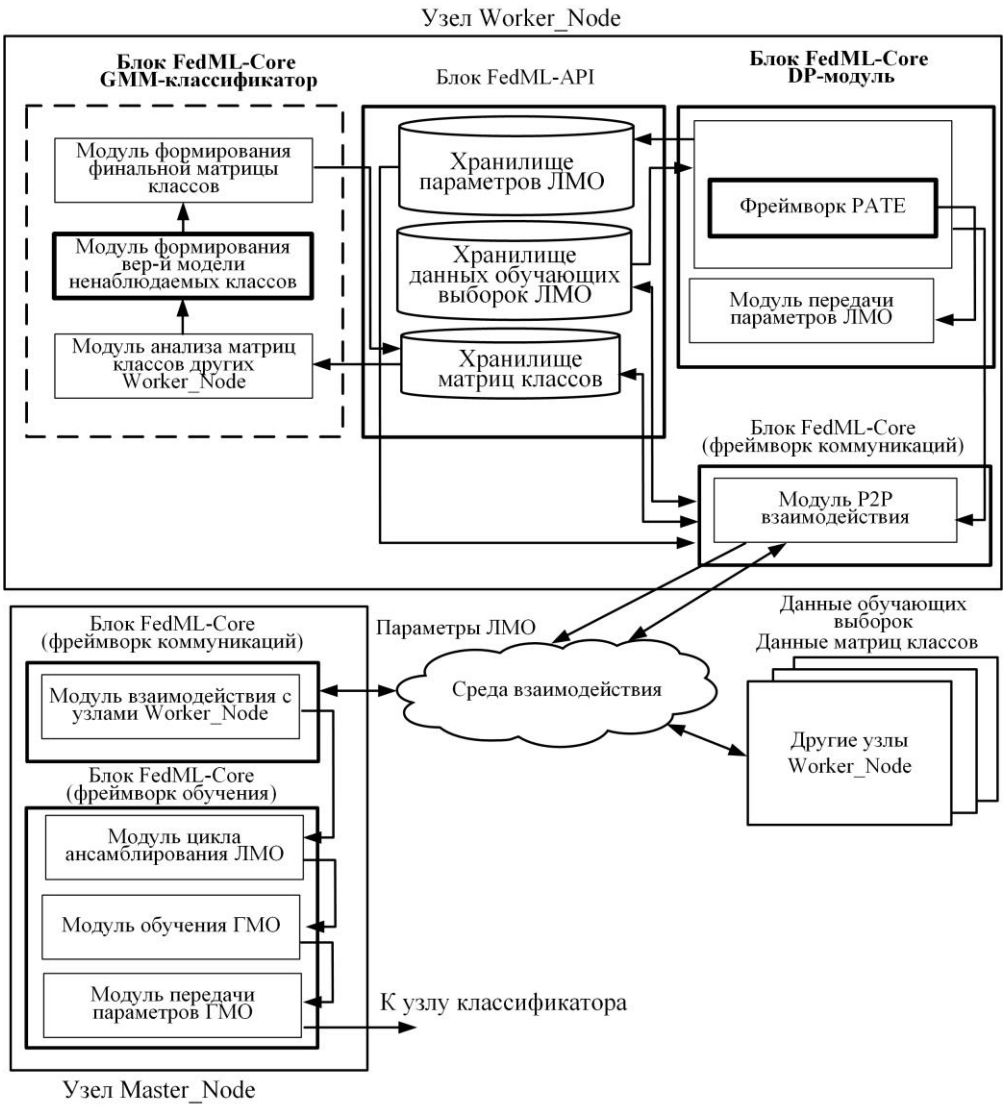

Рисунок 4.5 – Структура программного комплекса узлов системы многоэлементной классификации, функционирующей в условиях неполноты классов локальных классификаторов

Из рисунка 4.5 видно, что в составе структуры узла Worker_Node реализован блок GMM-классификатора (выделен пунктиром), подключаемый через интерфейс библиотеки FedML-API. Он представлен тремя модулями, функционирующими последовательно: модулем анализа матриц классов остальных узлов Worker_Node, работающих в составе системы, модулем формирования вероятностных моделей ненаблюдаемых классов (п. 2.1.1) и модулем формирования итоговой матрицы элементов (классов) узла Worker_Node, содержащей вероятностные модели ненаблюдаемых классов других узлов Worker_Node (п. 2.3). Для хранения получаемых от остальных узлов Worker_Node матриц элементов, а также сохранения собственной матрицы элементов в составе блока FedML-API, наряду с хранилищами обучающих выборок и параметров ЛМО, включено хранилище матриц элементов, использующих структуры данных фреймворка линейной алгебры NumPy [84].

В блоке дифференцированной конфиденциальности (DP-блоке), также реализованном на основе фреймворка обучения FedML-Core, в главе 1 (п. 1.4.2) используется фреймворк PATE, реализующий концепцию APL (независимого частного машинного обучения) на основе ансамбля взаимодействующих узлов Worker_Node. Взаимодействие узлов Worker_Node при этом производится с использованием модуля P2P-взаимодействия (реализация протокола DHT).

4.4 Выбор и обоснование среды имитационного моделирования программного комплекса системы федеративного машинного обучения для задачи многоэлементной классификации в условиях неполноты классов локальных классификаторов

Для оценивания эффективности разработанных в рамках исследования решений (главы 1-3), а также качества предложенной структуры программного комплекса (п. 4.3) требуется проведение экспериментальных исследований.

Очевидно, что выполнение натурного эксперимента в рамках существующих проектов систем с ФМО является труднореализуемым, в силу как проприетарного характера программного обеспечения коммерческих вариантов систем, исключающих возможность включения в их состав предлагаемых функциональных модулей, так и в силу конфиденциальности обрабатываемых ими данных, базирующейся на пользовательских соглашениях, подкрепленных, как Законом о персональных данных [12], так и иными федеральными и ведомственными законодательными актами.

В силу этого, в качестве экспериментальной базы в исследовании предлагается использование подхода на основе имитационного моделирования, при котором в модельном виде будут реализованы важные функциональные особенности реальных систем многоэлементной классификации с ФМО, а также особенности наборов обрабатываемых ими данных.

Особенностям проведения имитационных экспериментов подробно рассмотрены в [13, 14]. В частности, в [13] дается определение моделирующего алгоритма и приводится обобщенный цикл имитационного моделирования. Его модификация, применительно к объекту моделирования и предложенным в главах 1- 3 математическим и алгоритмическим решениям, представлена на рисунке 4.6.

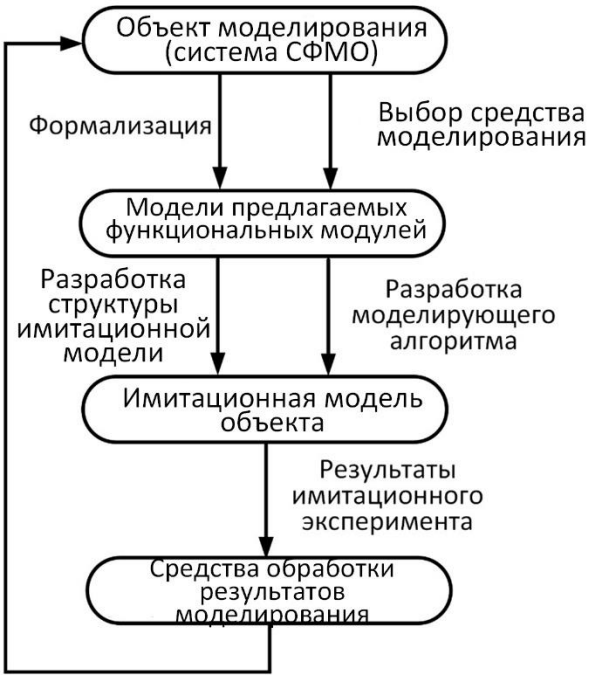

Рисунок 4.6 – Цикл имитационного моделирования применительно к объекту исследования – системе многоэлементной классификации

Анализ существующих средств имитационного моделирования в предметной области существующих систем с ФМО и посвященных им исследовательских проектов [88-93] показал, что большинство из них относятся либо к средствам предназначенным для проведения имитационных экспериментов для специализированных научных (диссертационные исследования, НИР) или коммерческих проектов [88-91], либо являются открытыми платформами СФМО с развитой системой конфигурирования, на базе которых возможна, в том числе, и разработка соответствующих имитационных моделей [92].

Кроме того, было определено, что для решения тестовых задач СФМО, включая их имитационное моделирование, были разработаны следующие наборы данных:

- EMNIST (классификация текстов, библиотека FedNLP); [93]
- CIFAR-100 (классификация изображений, библиотека FedCV); [94]

Также анализ исследовательских источников показал, что в рамках проектов систем с ФМО с открытым исходным кодом активное развитие получила платформа, специализированная для имитационного моделирования методов и алгоритмов в различных предметных областях ФМО. Это проект TensorOpera Federate, являющийся частью облачной инфраструктуры TensorOpera AI cloud [95]. TensorOpera Federate – это платформа машинного обучения, обеспечивающая кроссплатформенное ФМО и а также сбор и обработку соответствующих аналитических данных на основе набора предоставляемых в рамках проекта симуляторов систем с ФМО. Она позволяет проводить МО на основе децентрализованных данных, хранящихся в различных пользовательских хранилищах/периферийных узлах, без необходимости их централизации. TensorOpera Federate включает в себя кроссплатформенный SDK Edge AI, который можно развернуть на вычислительных системах на базе CPU и GPU, а также мобильных устройствах и устройствах IoT. В обобщенном виде состав компонентов

TensorOpera Federate, а также их функциональное назначение представлены на рисунке 4.7.

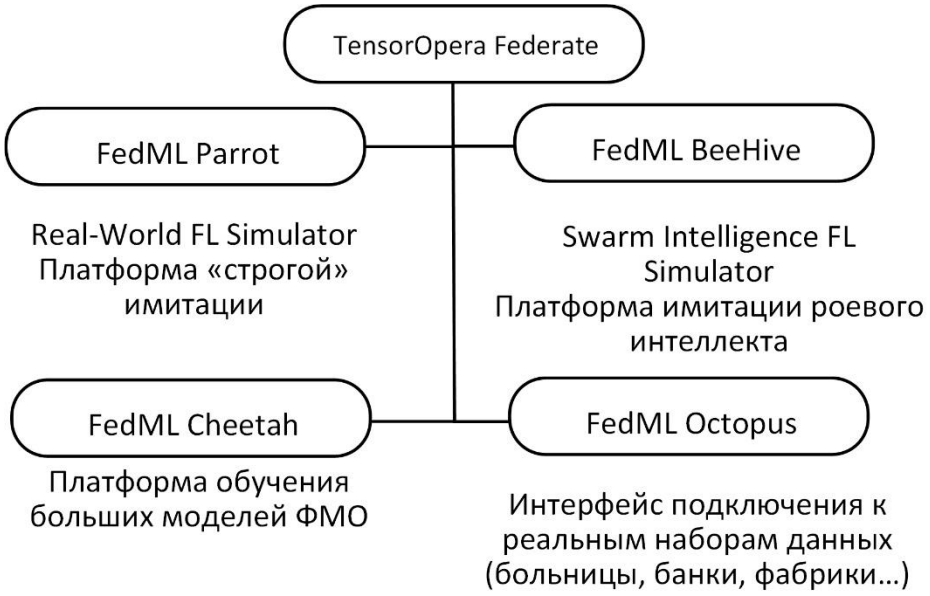

Рисунок 4.7 – Состав и функции компонентов платформы имитационного моделирования TensorOpera Federate

Платформа TensorOpera Federate поддерживает набор API как для работы с библиотекой FedML и централизованными хранилищами данных, так и с библиотекой MLOps, поддерживающей распределенное ФМО. К функциям платформы TensorOpera Federate относятся [95]:

1. TensorOpera Federate Simulation – моделирование реальных СФМО:

- моделирование ФМО на однопроцессорных (многоядерных) вычислительных системах;

- моделирование ФМО на базе систем MPI;

- моделирование ФМО на основе GPU, в частности NCCL (NVIDIA Collective Communications Library).

2. TensorOpera Federate Cross-silo – модель масштабирования ФМО cross-silo (п. 1.1.2) на основе Edge SDK на базе Python для организации процесса обучения между различными учреждениями.

3. TensorOpera Federate Cross-device – модель масштабирования ФМО cross-device (п. 1.1.2) на основе Edge SDK for Android/iOS and embedded Linux для мобильных устройств и устройств IoT.

4. TensorOpera AI – Federate – кроссплатформенный масштабируемый конвейер операций ФМО для систем ИИ на базе библиотеки FedML.

Вариант библиотеки FedML в составе платформы TensorOpera Federate представлен следующими модулями [78, 95]:

- модуль Core – пакет низкоуровневого API. Реализует распределенные вычисления и управление топологией с использованием таких коммуникационных интерфейсов, как MPI, NCCL, MQTT, gRPC, PyTorch RPC. Кроме того, он поддерживает другие низкоуровневые API, связанные с безопасностью и конфиденциальностью. Предназначен для разработки сценариев симуляции процесса ФМО;

- модуль Data – предоставляет разработчикам имитационных моделей ФМО несколько наборов данных и шаблонов настройки по умолчанию;

- модуль Model – набор предразработанных моделей ФМО для библиотеки FedML;

- модуль Device – обеспечивает управление вычислительными ресурсами библиотеки FedML;

- модуль Utils – утилиты общего назначения, используемые другими модулями.

Обобщенная структура формирования в рамках платформы TensorOpera Federate, как локально выполняемых моделей ФМО, так и распределенных моделей ФМО с использованием библиотеки MLOps, представлена на рисунке 4.8.

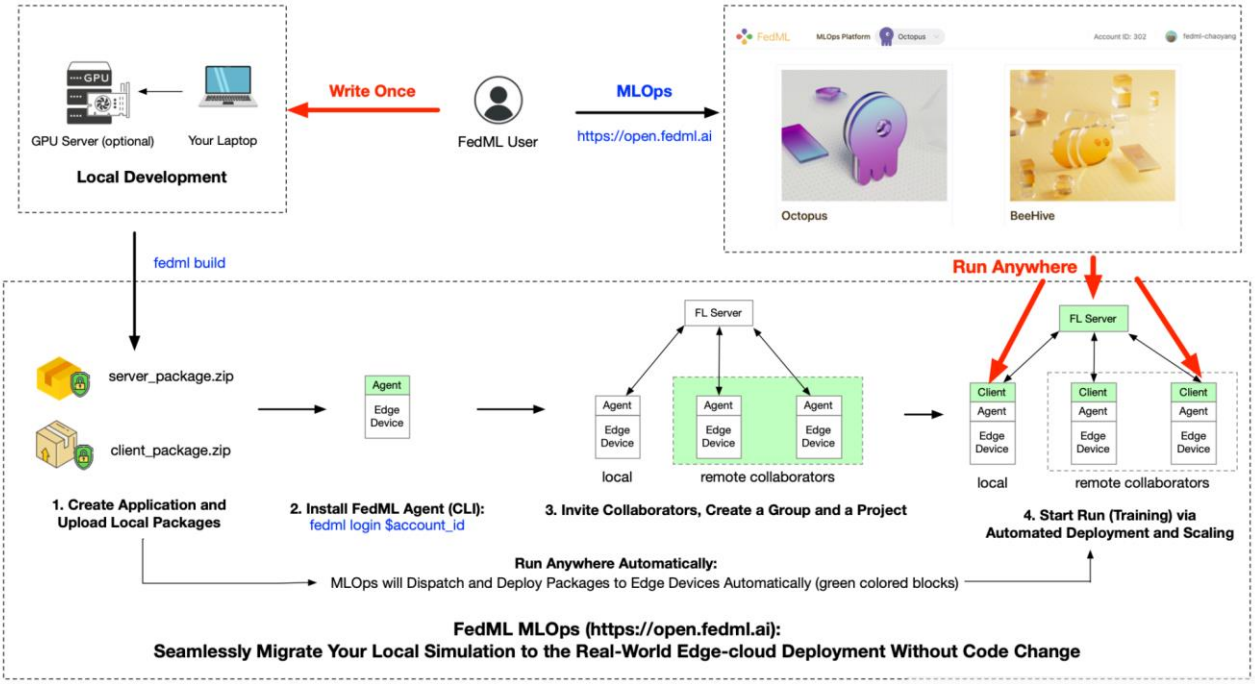

Рисунок 4.8 – Структура формирования моделей ФМО в рамках платформы TensorOpera Federate (источник рисунка - [93])

Обобщенная структура формирования в рамках платформы TensorOpera Federate, как локально выполняемых моделей ФМО, так и распределенных моделей ФМО с использованием библиотеки MLOps, представлена на рисунке 4.8. Поддерживаемые платформой TensorOpera Federate модели ФМО и наборы данных для процесса ФМО представлены в [96].

4.5 Разработка структуры имитационной модели распределенной системы многоэлементной классификации, функционирующей в условиях неполноты классов локальных классификаторов

На основе выбранной в п. 4.4 системы имитационного моделирования TensorOpera Federate с возможностью программной интеграции разработанного программного комплекса узлов СФМО (п. 4.4) была разработана структура имитационной модели, используемой в рамках проводимого исследования. Структурная схема разработанной имитационной модели представлена на рисунке 4.9.

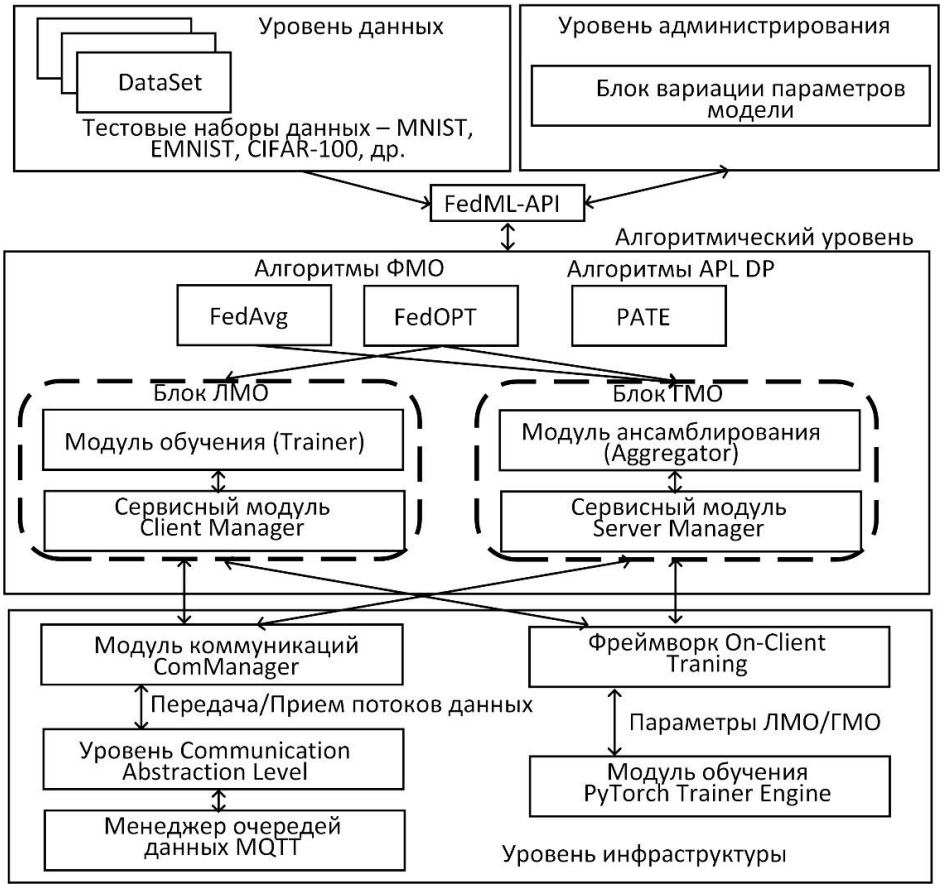

Рисунок 4.9 – Структурная схема имитационной модели системы классификации для задачи многоэлементной классификации в условиях неполноты классов локальных классификаторов

В основе структурной схемы имитационной модели предлагаемого в исследовании варианта СФМО лежат программные компоненты и интерфейсы их взаимодействия, входящие в состав модулей Core, Data, Model и Utils библиотеки FedML платформы TensorOpera Federate, функции которых были рассмотрены в п. 4.5.

Из рисунка 4.9 видно, что разработанная имитационная модель имеет структуру из трех уровней:

1. Уровень инфраструктуры является базовым сервисным уровнем и предназначен для:

- подключения к узлам имитационной модели модулей управления процессом ФМО на базе компонента PyTorch Trainer Engine. В его рамках формируется виртуальная структура СФМО, определяемая исследователем. Кроме того, параметры ЛМО и ГМО, получаемые на узлах Worker_Node и Master_Node, сохраняются в модуле On-Client Trainer;

- формирования схемы P2P взаимодействия узлов Worker_Node и маршрутизации параметров ЛМО и ГМО между узлами Worker_Node и Master_Node. Указанные функции реализуются с использованием модуля коммуникаций ComManager, а реализации протоколов взаимодействия DHT и HTTP реализуются модулями CAL (Communication Abctraction Level) и менеджером очередей данных MQTT.

Модуль ComManager – алгоритмы коммуникации компонентов модели для следующих топологий (рисунок 4.10):

- централизованной (схема Worker_Node-Master_Node). Поддерживаются технологии Fedvg, FedOpt, FedNova, FedGKT;

- децентрализованной (схема Worker_Node-Worker_Node). Поддерживается метод FL;

- вертикальное обучение (схема Worker_Node-Worker_Node, Worker_Node-NAS). Поддерживаются методы FL и FedNAS;

- иерархическое обучение (схема Worker_Node-Worker_Node). Поддерживается метод FL;

- раздельное (split) обучение (схема Worker_Node-Worker_Node). Поддерживается метод FL.

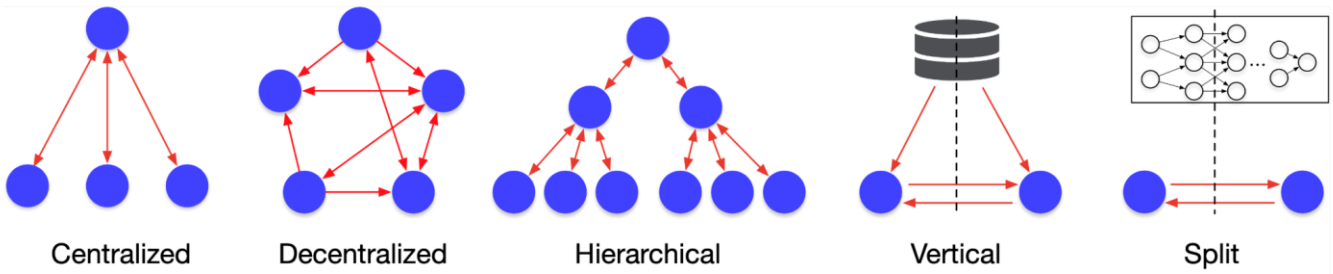

Рисунок 4.10 – Топологии взаимодействия, поддерживаемые модулем ComManager платформы TensorOpera Federate

Для предложенной структурной схемы программного комплекса СФМО (рисунок 4.5) при разработке имитационной модели была выбрана комбинированная схема коммуникаций, реализующая централизованную и децентрализованную

топологии. Реализация децентрализованной топологии базируется на протоколе DHT, реализованном в фреймворке поддержки взаимодействия в мультиагентных системах PADE (Python Agent Development framework) [97], предназначенном для реализации функциональности мультиагентных систем. При этом каждый Worker_Node рассматривается как автономный агент, обменивающийся сообщениями с другими агентами. Пример кода инициализации узла Worker_Node в качестве агента децентрализованной топологии представлен на рисунке 4.11.

```
from pade.misc.utility import display_message
from pade.misc.common import set_ams, start_loop
from pade.core.agent import Agent
from pade.acl.aid import AID


class AgenteHelloWorld(Agent):
    def __init__(self, aid):
        super(AgenteHelloWorld, self).__init__(aid=aid, debug=False)
        display_message(self.aid.localname, 'Hello World!')

if __name__ == '__main__':

    set_ams('localhost', 8000, debug=False)

    agents = list()

    agente_hello = AgenteHelloWorld(AID(name='agente_hello'))
    agente_hello.ams = {'name': 'localhost', 'port': 8000}
    agents.append(agente_hello)

    start_loop(agents, gui=True)
```

Рисунок 4.11 – Пример кода инициализации агента децентрализованной топологии

2. Алгоритмический уровень является основным с точки зрения реализации моделирующего алгоритма. В его рамках развертывается виртуальная структура СФМО, сформированная компонентом PyTorch Trainer Engine уровня инфраструктуры и с помощью модуля FedML Parrot data_loader.py подгружаются выбираемые исследователем алгоритмы ФМО, в частности, выбранная в п. 4.3 модели GMM-классификатора на базе фреймворка Scikit Learn. На их основе в памяти вычислительной системы развертываются имитаторы узлов Worker_Node (на рисунке 4.9 – Блоки ЛМО) и узла Master_Node (на рисунке 4.9 – Блок ГМО).

Из рисунка видно, что блок ЛМО содержит следующие модули:

- Trainer (модуль обучения) – базовый, с точки зрения процесса ФМО;

- Client Manager – сервисный модуль управления загрузкой тестовых наборов данных и подключения дополнительных функций.

Отличительной особенностью блока ЛМО, реализованного в рамках настоящего исследования, является динамическое подключение модулем Client Manager к множеству модулей Trainer через интерфейс FedML API программных компонентов фреймворка PATE, реализующего функции дифференцированной конфиденциальности на основе концепции APL (п. 1.3.3).

В свою очередь блок формирования итоговой МО содержит следующие модули:

- Aggregator – модуль ансамблирования параметров ЛМО, поступающих от узлов Worker_Node в параметры ГМО на основе алгоритма FedAvg (рисунок 1.9);

- Server Manager – сервисный модуль управления загрузкой параметров ЛМО и подключения дополнительных функций.

Параметры модулей Client Manager и Server Manager, используемые в процессе моделирования приведены в таблице 4.2.

Таблица 4.2

Параметры модулей Client Manager и Server Manager

Параметр	Описание параметра
client_number	Общее количество клиентских узлов (Worker_Node)
train_data_num	Общее количество обучающих выборок
test_data_num	Общее количество тестовых выборок
train_data_global	Глобальный тренировочный набор данных в формате PyTorch_dataloader
test_data_global	Глобальный тестовый набор данных в формате PyTorch_dataloader
train_data_local_dict	Словарь для индексации загрузчика данных для каждого клиента в формате key/value. Значение key — индекс узла Worker_Node, а значение value — локальные данные узла Worker_Node в формате PyTorch_dataloader

3. Уровень данных. Поддерживает множество тестовых наборов данных, используемых модулем обучения Trainer. По умолчанию подключены следующие наборы данных: MNIST, EMNIST (изображения 28x28 – рукописный текст) – используются для задач классификации и регрессии [93], CIFAR100 (изображения 32x32, группировка 100 классов) – используются для задач многоклассовой классификации [94].

4. Уровень администрирования. Представлен панелью управления (DashBoard) (рисунок 4.12), обеспечивающей функции варьирования параметров модели.

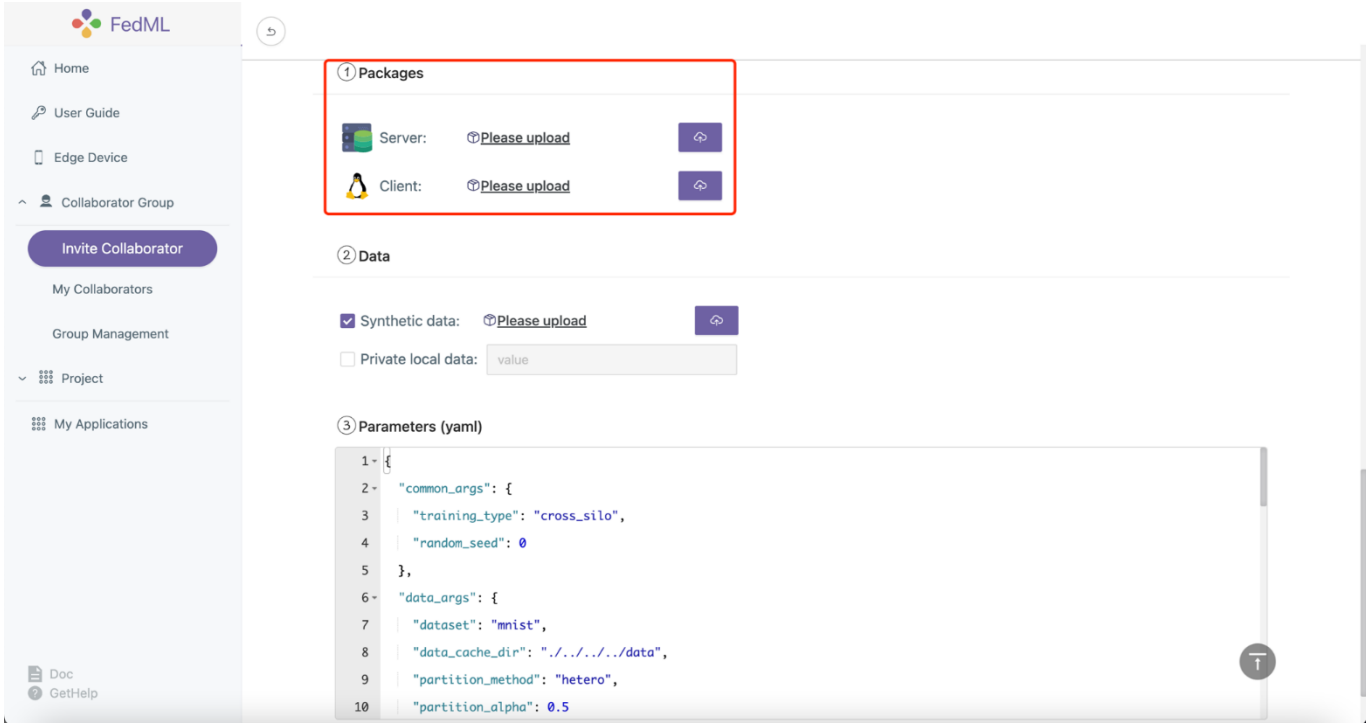

Рисунок 4.12 – Пример интерфейса панели управления (уровень администрирования имитационной модели)

В соответствии с алгоритмом функционирования модуля FedML Parrot для представленных выше компонентов алгоритмического уровня выполняются следующие функции этапов ФМО:

- init FedML_framework – инициализация фреймворка FedML и получение настройки параметров (args) каждого компонента;

- init device – инициализация запуска компонентов модели в соответствии с параметрами, связанными с настройками (device);

- load data – загрузка набора данных information() и размерность task_output() в соответствии с параметрами, относящимися к данным. Например, текущий набор данных является задачей многоклассовой классификации с 10 классами, то есть параметр размерность будет 10.datasetoutput_dim;

- load model – загрузка инициализированной модели в соответствии с параметрами (model);

- start training – инициализация и запуски run() имитационной модели с параметрами Simulator(args, device, dataset, model).

4.6 Разработка моделирующего алгоритма предложенной имитационной модели разрабатываемой системы

Поскольку по своей природе имитационная модель – модель прогонного типа, то, как отмечается в [85], она имеет свойства, присущие алгоритмическим конструкциям, а именно: детерминированность, конечность, массовость и результативность.

Таким образом, процесс имитационного моделирования для предлагаемой имитационной модели (п. 4.7) определяется структурой, именуемой моделирующий алгоритм - структура и правила взаимодействия элементов имитационной модели при их реализации на ЭВМ. Основы разработки моделирующих алгоритмов представлены в [13, 14]. важной задачей процесса имитационного моделирования является разработка моделирующего алгоритма.

В основе моделирующего алгоритма лежит цикл прогона модели (основной цикл), использующий значения входных параметров элементов модели, участвующих в прогоне.

Выход из основного цикла моделирующего алгоритма происходит после выполнения одного из условий:

- завершение определенного исследователем значения модельного времени;

- завершения обработки заданного числа значений параметров имитационной модели.

Для предложенной имитационной модели невозможно выбрать детерминированные значения модельного времени каждого прогона, поскольку эти значения зависят от случайных факторов, влияющих на продолжительность раундов машинного обучения в комбинированной централизованно-децентрализованной схеме. Таким образом, останов основного цикла моделирующего алгоритм будет выполняться на основе значений счетчиков параметров компонентов имитационной модели (цикл for со счетчиками итераций).

Обобщенная схема разработанного моделирующего алгоритма представлена на рисунке 4.13.

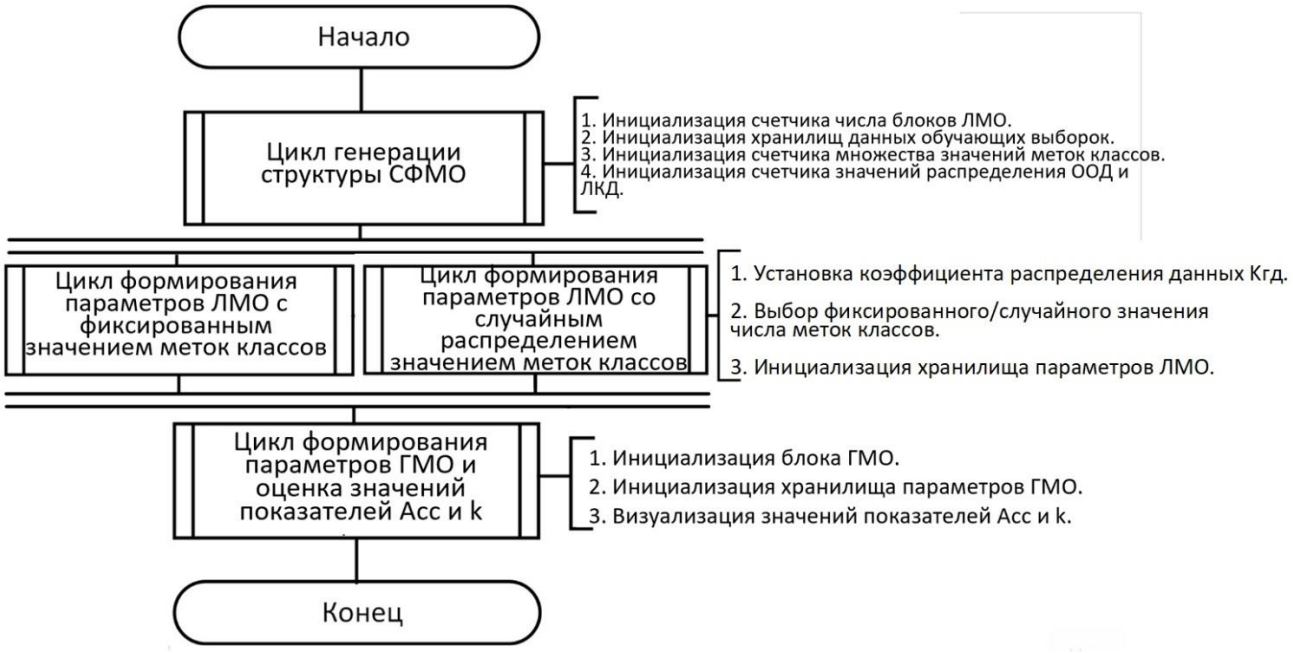

Рисунок 4.13 – Обобщенная схема моделирующего алгоритма

Из рисунка 4.13 видно, что разработанный моделирующий алгоритм состоит из четырех предопределенных процессов, формирующих один прогон модельного эксперимента.

Первый предопределенный процесс формирует множество блоков ЛМО и инициализирует их начальные параметры, такие как мощность множества элементов обучающей выборки.

Второй и третий предопределенные процессы реализуют идентичные циклы раунда обучения ЛМО в условиях неполноты классов и могут выполняться

параллельно. Отличием второго предопределенного процесса от третьего является получение матрицы классов при фиксированном значении классов c=3 в каждом из n блоков ЛМО, в то время как в третьем предопределенном процессе количество классов в каждом блоке ЛМО формируется выбором случайного значения множества C мощностью 5.

Схема второго и третьего предопределенных процессов представлена на рисунке 4.14.

Четвертый предопределенный процесс реализует функции блока ГМО (агрегация параметров множества ЛМО, обучение ГМО, оценивание ГМО по показателям точности (Accuracy) и каппа Коэна (k)) и блока администрирования, реализующего визуализацию результатов оценивания.

Таким образом, в рамках одного прогона моделирующего алгоритма выполняется варьирование следующих параметров:

- количество блоков ЛМО в составе СФМО;

- коэффициент $K_{ГД}$ – распределения данных обучающей выборки каждого блока ЛМО на ООД и ЛКД;

- количество меток классов в каждом блоке ЛМО (фиксированное или случайно выбранное из множества меток классов).

Указанные параметры позволяют в рамках одного прогона сформировать различные условия функционирования СФМО, оказывающие влияние на качество сформированной ГМО. В первую очередь к таким условиям следует отнести объем обрабатываемых каждым блоком ЛМО данных ЛКД и количество классов, используемых классификатором блока ЛМО.

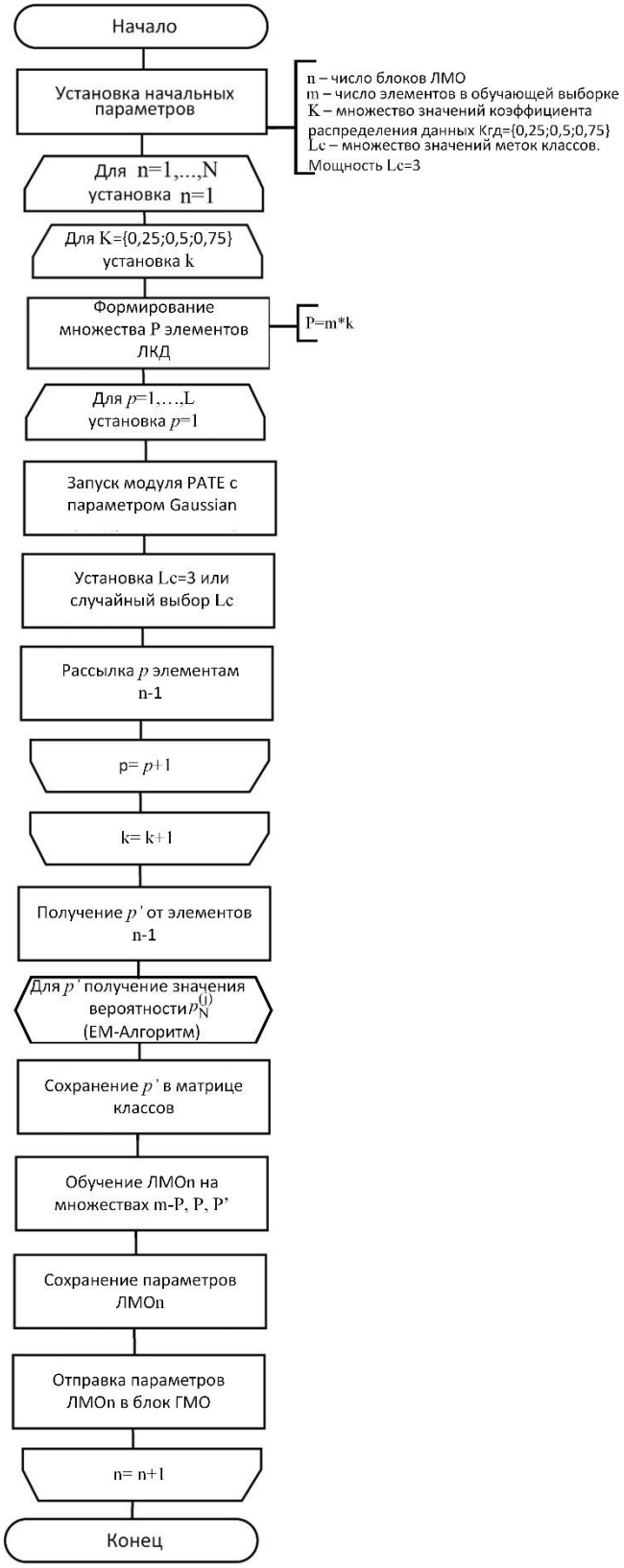

Рисунок 4.14 – Схема моделирующего алгоритма блока ЛМО (предопределенные процессы 2 и 3 на рисунке 4.13).

4.7 Методика расчета числа имитационных прогонов

Одной из необходимых задач, требующих решения при разработке плана имитационного эксперимента является обоснование вида и рационального числа имитационных прогонов.

Вид имитационного прогона определяет характер рассмотрения модельного времени в ходе проводимого имитационного эксперимента. Как следует из [87], в имитационном моделировании распространено использование следующих видов прогонов:

- линейные – два или более повторяющихся имитационных прогона с единым масштабом модельного времени;

- масштабируемые линейные – два или более имитационных прогона с положительным или отрицательным масштабированием шкалы линейного времени каждого последующего прогона;

- прогоны на основе подинтервалов – разделение множества прогонов на группы с различным масштабом модельного времени и расчет среднего значения показателей для каждой группы.

В рамках проводимого исследования целесообразно выбрать вид линейных прогонов, поскольку целью модельного эксперимента является обоснование пригодности предлагаемых решений, а не определение оптимальных условий их использования.

В общем случае задача определения необходимого количества прогонов является задачей математической статистики и состоит в определении необходимого для выполнения модельного эксперимента объема выборки N – количества прогонов моделирующего алгоритма.

Для решения задачи выбора рационального числа имитационных прогонов в исследовании рассматривались подходы, представленные в [13]. К ним относятся:

1. Метод последовательных испытаний, основанный на плановом увеличении количества пробных прогонов и их остановом при достижении требуемой

статистической точности эксперимента. Недостатком подобного подхода является высокая временная и вычислительная ресурсоемкость;

2. Метод квантилей, представленный следующими этапами:

- эмпирическое определение требуемого уровня точности (например, ±5% от среднего значения);

- эмпирическое определение требуемой надежность полученного результата (например, 95%);

- табличное нахождение z-статистики, соответствующей заданной надежности – показателя, определяющего на сколько стандартных отклонений конкретное значение данных отклоняется от среднего значения набора данных, выраженного в стандартных отклонениях;

- расчет числа имитационных прогонов по следующему выражению:

$$N=\left(\frac{Z\cdot\sigma}{E}\right)^2, \tag{71}$$

где σ – дисперсия, а E – допустимая погрешность.

- последовательный анализ получаемой после каждого нового прогона в пробной последовательности;

- останов пробных прогонов при стабилизации значений статистики.

На основе использования метода квантилей при проведении имитационного эксперимента было принято решение о числе прогонов $N=5$.

4.8 Выполнение имитационного эксперимента и получение сравнительной оценки предложенного решения

В рамках разработанной имитационной модели (п. 4.6) на основе разработанного моделирующего алгоритма (п. 4.6) был спланирован сравнительный имитационный эксперимент для оценивания результата многоэлементной классификации в условиях неполноты классов локальных классификаторов (наличие

в обучающей и тестовой выборках данных ЛКД) системы многоэлементной классификации со следующими структурами:

- традиционная система классификации централизованного типа (рисунок 1.7);

- предложенная в ходе исследования система многоэлементной классификации гибридного типа с поддержкой механизма дифференцированной конфиденциальности (п. 4.2).

Входные параметры для каждого имитационного прогона определялись в соответствии с разработанным моделирующим алгоритмом (п. 4.6).

Число имитационных прогонов было рассчитано в п. 4.7.

Количество блоков ЛКД в каждой из сравниваемых альтернатив варьировалось от 2 до 5.

В качестве обучающей и тестовой выборок использовалось подмножество изображений набора данных CIFAR100 (элементы одежды) [94], сгруппированных, соответственно, в подмножества с 3, 5 и 10 классами.

Параметры множества вероятностных GMM-моделей кандидатов (п. 2.1.1) $\{GMM_0, GMM_1, ..., GMM_9\}$ для десяти классов изображений представлено в таблице 4.3.

Таблица 4.3

Параметры множества вероятностных GMM-моделей кандидатов классов изображений

Идентификатокласса	Изображение	Параметры GMM-модели			
		C_gn - число используемых компонентов	Тип матрицы ковариации	Коэфф. неотрицательной регуляризации	Рассчитанный BIC
Class0	Футболка	2	полная	0,001	218599,8304
Class1	Плавки	2	полная	0,001	92074,9725
Class2	Свитер	5	полная	0,00001	215139,5721
Class3	Платье	3	полная	0,001	218177,7049
Class4	Пиджак	3	полная	0,01	209474,0943
Class5	Шорты	2	связанная	0,001	271939,6885
Class6	Тапочки	5	полная	0,0001	225819,9335
Class7	Кроссовки	2	полная	0,01	166274,3903
Class8	Туфли муж.	5	полная	0,001	293328,9243
Class9	Туфли жен.	3	полная	0,01	226256,1887

Выполнение спланированного сравнительного имитационного эксперимента преследовало определение следующих зависимостей:

1. Сравнительное оценивание традиционной централизованной схемы с предложенной в ходе проведения исследования схемой по качеству многоэлементной классификации в следующих модельных условиях:

- последовательное увеличение числа блоков ЛМО с 10 до 40;

- последовательное изменение для каждого блока ЛМО коэффициента Кгд – распределения в обучающей и тестовой выборках данных ООД и ЛКД (рисунок 4.14).

Фиксированными параметрами являлись:

- подмножество CIFAR100 с изображениями, сгруппированными в 3 класса (Lc=3).

Табличное представление полученных в ходе имитационных прогонов данных о значении показателей Accuracy (ACC) и Каппа Коэна (k) (п. 1.5) для 5 прогонов моделирующего алгоритма приведено в Приложении 2. В каждом прогоне рассчитывалось среднее значение полученных данных, а также их дисперсия.

2. Сравнительное оценивание традиционной централизованной схемы с предложенной в ходе проведения исследования схемой по качеству многоэлементной классификации в следующих модельных условиях:

- последовательное увеличение числа блоков ЛМО с 10 до 40 с шагом 10 блоков;

- последовательное изменение для каждого блока ЛМО коэффициента Кгд – распределения в обучающей и тестовой выборках данных ООД и ЛКД (рисунок 4.14).

- случайный выбор подмножеств CIFAR100 с изображениями, сгруппированными в 3, 5 и 10 классов соответственно (Lc=variable).

Табличное представление полученных в ходе имитационных прогонов данных о значении показателей Accuracy (ACC) и Каппа Коэна (k) для 5 прогонов моделирующего алгоритма приведено в Приложении 3. В каждом прогоне рассчитывалось среднее значение полученных данных, а также их дисперсия.

Результаты их обработки для централизованной и предлагаемой схем представлены на рисунках 4.15-4.16 соответственно.

Из рисунков видно, что в условиях неполноты классов блоков ЛМО значение точности для централизованной схемы не превышает 0,5, а Каппа Коэна не превышает 0,52. При этом:

- переменное число меток классов в блоках ЛМО, определяемых Lc=var, снижает значение точности до 0, 33 (в среднем) и Каппа Коэна до 0,15 (в среднем);

- увеличение числа блоков ЛМО, включенных в централизованную схему СФМО дает незначительный прирост качества многоэлементной классификации, как с фиксированным, так и с переменным числом меток Lc – в среднем на 0,2 для показателей точности и Каппа Коэна.

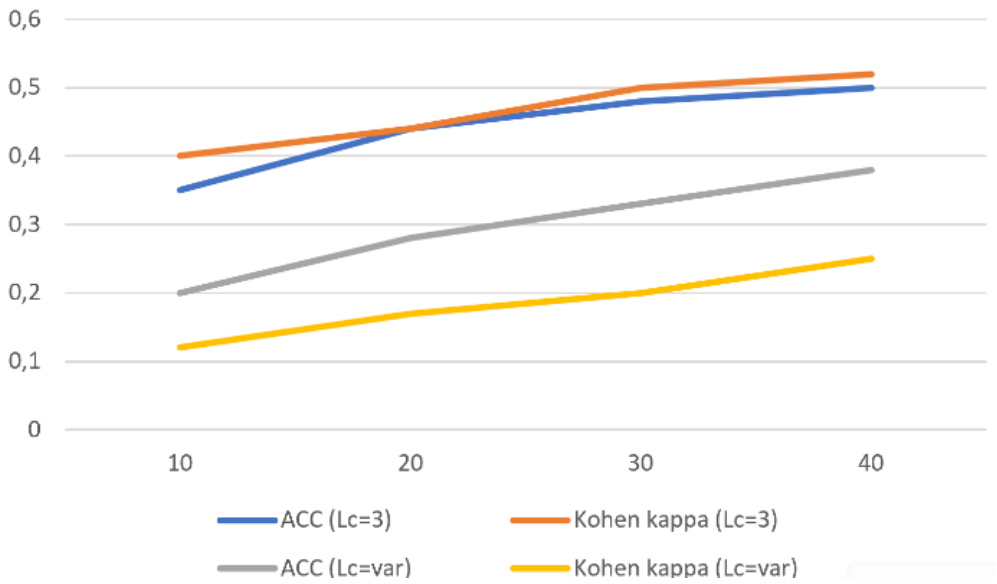

Рисунок 4.15 – Качество многоэлементной классификации с фиксированным (Lc=3) переменным (Lc=var) числом меток классов Lc=3 по показателям точности (ACC) и Каппа Коэна (k) для централизованной схемы

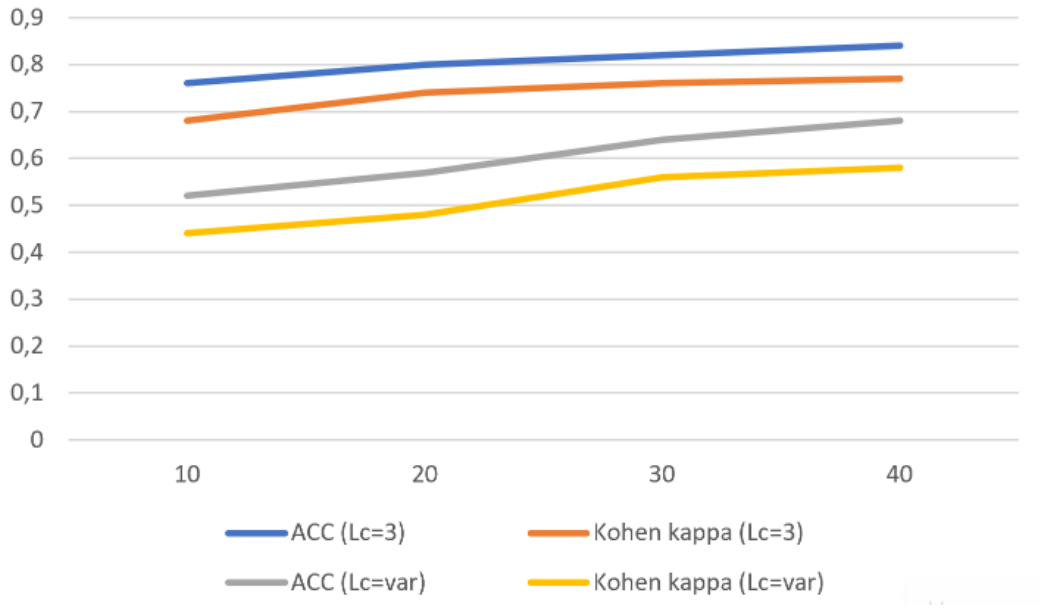

Рисунок 4.16 – Качество многоэлементной классификации с фиксированным (Lc=3) переменным (Lc=var) числом меток классов Lc=3 по показателям точности (ACC) и Каппа Коэна (k) для разработанной схемы

Для предлагаемой схемы максимальное значение точности для централизованной схемы СФМО равно 0,84, а Каппа Коэна равно 0,77. При этом:

- переменное число меток классов в блоках ЛМО, определяемых Lc=var, не оказывает существенного влияния на качество многоэлементной классификации, снижая значения точности и Каппа Коэна в среднем на 0,1;

- однако, увеличение числа блоков ЛМО, включенных в централизованную схему системы много элементной классификации с ФМО также, как и для централизованной схемы, дает незначительный прирост качества многоэлементной классификации, как с фиксированным, так и с переменным числом меток Lc – в среднем на 0,2 для показателей точности и Каппа Коэна, что демонстрирует инвариантность процесса многоэлементной классификации к мощности множества блоков ЛМО, входящих в состав системы с ФМО.

4.9 Выводы по главе

В главе предложена структура программного комплекса распределенной системы многоэлементной классификации, обеспечивающего поддержку ее решения в условиях неполноты классов локальных классификаторов.

В отличие от традиционных схем структура программного комплекса является гибридной: формирование итоговой модели классификатора (ГМО) реализуется по традиционной централизованной схеме, а решение проблемы обучения множества моделей локальных классификаторов (ЛМО) на гибридных (общедоступных и локально-конфиденциальных) данных реализуется по децентрализованной схеме, основанной на модификации метода модельно-независимого машинного обучения.

В главе подробно описана разработанная структура системы, а также обоснован выбор фреймворков для программной реализации ее компонентов и разработан вариант их программного обеспечения.

В качестве оценки эффективности предлагаемых решений представлены результаты сравнительного имитационного эксперимента по оцениванию качества многоэлементной классификации, реализованного на базе выбранной платформы TensorOpera Federate.

Таким образом, в главе представлена структура программно-реализованной системы многоэлементной классификации, обеспечивающей формирование итоговой модели классификатора в условиях неполноты классов подмножества локальных классификаторов, входящих в ее состав. Результаты имитационного эксперимента подтверждают достижение цели исследования. На реализованные элементы разработанного ПО специальных модулей локального и глобального классификаторов, получено свидетельство о регистрации программы для ЭВМ в реестре ФИПС.

ЗАКЛЮЧЕНИЕ

Диссертационная работа направлена на разработку средств математического и программного обеспечения децентрализованного обмена информацией в распределенных системах многоэлементной (многоклассовой) классификации с федеративным машинным обучением, функционирующих в условиях неполноты классов их локальных классификаторов.

Научная задача, решенная в диссертации, может быть классифицирована как задача применения известных научных методов в новой предметной области.

Достоверность и обоснованность полученных результатов подтверждается научно организованными экспериментами, корректным применением известных методов исследования, адекватных природе изучаемых процессов и явлений, непротиворечивостью и воспроизводимостью результатов, полученных в процессе сравнительного анализа вычислений и натурных экспериментов.

В процессе выполнения диссертационного исследования получены следующие основные результаты:

1. Разработана модель классификатора распределенной системы многоэлементной классификации, учитывающая неполноту локальных элементных матриц и основанную на представлении меток ненаблюдаемых классов распределением плотности вероятности.

2. Разработан алгоритм получения значений оценок вероятностной функции ненаблюдаемых классов локальных классификаторов, основанный на методе расчета параметров статистической вероятностной модели со смешанными распределениями.

3. Разработан алгоритм децентрализованного обмена данными узлов распределенной системы многоэлементной классификации, функционирующей в условиях неполноты локальных элементных матриц классов, обеспечивающий получение полной элементной матрицы с учетом потенциально ненаблюдаемых классов.

4. Модифицирована существующая архитектура распределенной системы многоэлементной классификации и реализующий ее программный комплекс, обеспечивающие формирование итоговой модели обучения классификатора для условий неполноты локальных элементных матриц классов.

СПИСОК ТЕРМИНОВ, СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

В настоящей работе применяются следующие сокращения:

AVA – All-Vs-All – Все против всех
CV – Computer Vision – Компьютерное зрение
DP – Differentially Privacy – Дифференцированная приватность
DP-SGD – Differentially Private Stochastic Gradient Descent – дифференцированно приватный стохастический градиентный спуск
FL – Federative Learning – Федеративное обучение
HFL – Horizontal FL – Горизонтальное федеративное обучение
GAN – Generative adversarial networks – генеративно-состязательная сеть
GPT – Generative pre-trained transformer – нейронная языковая модель
IID – Independent and Identically Distributed Data – независимые и одинаково распределенные данные
IoT – Internet of Things – Интернет вещей
NMPU – Neural-Morphing Processing Unit – Нейро-морфный процессор
MAP – Maximum A posteriori Probability – Максимальная апостериорная вероятность
OVA – One-Vs-All – Один против всех
OVO – One-Vs-One – Один против одного
SGD – стохастический градиентный спуск
SMPC – Secure Multi-Party Computation – безопасное многовариантное вычисление
TPU – Tensor Processor Unit – Тензорный процессорный блок
VFL – Vertical FL – Вертикальное федеративное обучение
VPU – Vision Processor Unit – Блок визуального процессора
БОУ – базовые обучаемые устройства
ВИС – взаимное информационное согласование
ГМО – глобальная МО
ИИ – искусственный интеллект
КТ – компьютерная томография
ЛМО – локальная ММО
МО – машинное обучение
ММО – модель машинного обучения
МРТ – магнитно-резонансной томография
СХД – система хранения данных
СФМО – система федеративного машинного обучения
ФМО – федеративное МО
ФО – федеративное обучение
ЦОД – центр обработки данных

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. LeCun, Y., Bengio, Y., Hinton, G. Deep learning // Nature, no. 5(21), 2015, pp. 436-472.

2. Sun, C., Shrivastava, A., Singh, S. & Gupta, A. Revisiting unreasonable effectiveness of data in deep learning era // In Proceedings of the IEEE international conference on computer vision, 2017, v.3, pp. 843–852 (IEEE, 2017).

3. Wang, F., Casalino, L. P. & Khullar, D. Deep learning in medicine—promise, progress, and challenges // JAMA Intern. Med. 179, 2019. pp. 293–294.

4. Blanchard, O., Vines, D., Wills, S. On the future of macroeconomic models: Rebuilding macroeconomic theory // Oxford Review of Economic Policy. 2018. no. 2, pp. 43-54.

5. Vu-Quoc, L., Humer, A. Deep learning applied to computational mechanics: A comprehensive review, state of the art, and the classics // Computer Modeling in Engineering & Sciences, 2023. v. 37, no. 2, pp. 1069-1343.

6. Pires, I., Faisal, H, Garsia, N, Lameski, P. Homogeneous Data Normalization and Deep Learning: A Case Study in Human Activity Classification // Future Internet, 12(194). 2020. pp. 2-14.

7. Намиот Д. Е. Введение в атаки отравлением на модели машинного обучения // International Journal of Open Information Technologies. Т. 11. №. 3. 2023. С. 58-68.

8. Mitra, M. Neural processor in artificial intelligence advancement // Journal of Autonomous Intelligence, 1(1):2, 2018. pp. 54-67.

9. Youvan, D. Colossus Unveiled: The World's Most Powerful AI Supercomputer and Its Implications for the Future of AI Development // preprint Researchgate.net, Oct. 2024. P. 15. DOI: 10.13140/RG.2.2.24131.82726.

10. Van Panhuis, W. G. A systematic review of barriers to data sharing in public Health // BMC Public Health, no. 14, 2014. pp. 1112-1144.

11. Schwarz, C. G. Identification of anonymous mri research participants with face-recognition software // N. Engl. J. Med. No. 381, 2019. pp. 1684–1686.

12. Chai, Z., Fayyaz, H., Fayyaz, Z., Anwar, A., Zhou, Y., Baracaldo, N. Towards taming the resource and data heterogeneity in federated learning // In USENIX conference on operational machine learning, v. 1, 2019. pp. 19–21.

13. Li, T., Sahu, A. K., Talwalkar, A. & Smith, V. Federated learning: Challenges, methods, and future directions // IEEE Signal Processing Magazine, no. 37, 2020. pp. 50–60.

14. Yang, Q., Liu, Y., Chen, T. & Tong, Y. Federated machine learning: concept and applications // ACM Trans. Intell. Syst. Technol. (TIST), no. 10, 2019. pp. 57-68.

15. Patra, B., Tamrakar, A., Sharma, R. Edge computing: evolution, challenges, and future directions // Turkish Journal of Computer and Mathematics Education (TURCOMAT), no. 10(1), 2019. pp. 741-745.

16. Konečny, J., McMahan, H. B., Ramage, D., & Richtárik, P. (2016). Federated optimization: Distributed machine learning for on-device intelligence // arXiv preprint. 2016, arXiv:1610.02527.

17. Banabilah, S., Aloqualy, M., Asayed, E., Malik, N., Jararweh, Y. Federated learning review: Fundamentals, enabling technologies, and future applications // Information Processing and Management, no. 59, 2022. pp. 46-70.

18. Federation Learning Solution Market // [Электронный ресурс]. – Режим доступа: https://www.grandviewresearch.com/industry-analysis/federated-learning-market-report. – Дата доступа: 28.10.2019.

19. Federation Learning Market Size by Application // [Электронный ресурс]. – Режим доступа: https://www.maximizemarketresearch.com/market-report/global-federated-learning-solutions-market/96614/. – Дата доступа: 15.02.2020.

20. Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., Liu, X., He, B. A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection // arXiv preprint. 2021, arXiv:1907.09693v7.

21. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs // IEEE transactions on pattern analysis and machine intelligence 40(4), 2018. pp. 834–848.

22. Yurdem, B., Kuzlu, M., Gullu, M., Katak, F., Tabassum, M. Federated learning: Overview, strategies, applications, tools and future directions // Helion, no. 10, 2024. pp. 124-148.

23. Ding, Y., Wu, C., Tang, S., Lu, C. Federated Submodel Averaging // arXiv preprint. 2021, arXiv:2109.07704v1.

24. Li, Q., Wen, Z., He, B. Practical Federated Gradient Boosting Decision Trees // arXiv preprint. 2019, arXiv:1911.04206v2.

25. Намиот, Д.Е., Ильюшин, Е.А. Мониторинг сдвига данных в моделях машинного обучения // International Journal of Open Information Technologies ISSN: 2307-8162 vol. 10, no. 12, 2022. pp. 84-94.

26. Bonawitz, K. Towards federated learning at scale: System design // Proceedings
of Machine Learning and Systems, no. 1, 2019. pp. 374-388.

27. Запечников, С.В. Модели и алгоритмы конфиденциального машинного обучения // Безопасность информационных технологий, Том 27, № 1, 2020. С. 51-68.

28. Liu, Y., Kang, Y., Xing, C., Chen, T., Yang, Q. A secure federated transfer learning framework // IEEE Intelligent Systems, 35 (4), 2020. pp. 70-82.

29. Mohassel, P., Zhang, Y. Secureml: A system for scalable privacy-preserving
machine learning // IEEE symposium on security and privacy (SP), vol. 1(2), 2017. pp. 19-38.

30. R. C. Geyer, T. Klein, and M. Nabi, Differentially private federated learning: A client level perspective, arXiv preprint arXiv:1712.07557, 2017.

31. Дифференциальная приватность в машинном обучении // [Электронный ресурс]. – Режим доступа: https://habr.com/en/companies/otus/articles/788332/. – Дата доступа: 15.03.2024.

32. Введение в машинное обучение // [Электронный ресурс]. – Режим доступа: https://habr.com/en/articles/448892/. – Дата доступа: 07.10.2023.

33. Liu, K.H., Xu, C.G. A genetic programming-based approach to the classification
of multiclass microarray datasets // Bioinformatics 25 (3), 2009. pp. 331–337.

34. Torralba, A., Murphy, K.P., Freeman, W.T. Sharing visual features for multiclass
and multiview object detection // IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (5), 2007. pp. 854–869.

35. Hong, J.H., Min, J.K., Cho, U.K., Cho, S.B. Fingerprint classification using one-vs-all support vector machines dynamically ordered with Naive Bayes classifiers // Pattern Recognition 41 (2), 2008. pp. 662–671.

36. Aran, O., Akarun, L. A multi-class classification strategy for fisher scores: application to signer independent sign language recognition // Pattern Recognition 43 (5), 2010. pp. 1776–1788.

37. Anand, A.P., Suganthan, N. Multiclass cancer classification by support vector machines with class-wise optimized genes and probability estimates, Journal of Theoretical Biology 259 (3), 2009. pp. 533–540.

38. Guler, I., Ubeyli, E.D. Multiclass support vector machines for EEG-signals classification // IEEE Transactions on Information Technology in Biomedicine 11 (2), 2007. pp. 117–126.

39. Furnkranz, J. Round robin classification // Journal of Machine Learning Research 2, 2002. pp. 721–747.

40. Knerr, S., Personnaz, L., Dreyfus, G. Single-layer learning revisited: a stepwise procedure for building and training a neural network // F. Fogelman Soulie´, J. He´ rault (Eds.), Neurocomputing: Algorithms, Architectures and Applications. ASI Series, vol. F68, 1990. pp. 41–50.

41. Anand, R., Mehrotra, K., Mohan, C.K., Ranka, S. Efficient classification for multiclass problems using modular neural networks // IEEE Transactions on Neural Networks 6 (1), 1995. pp. 117–124.

42. Furnkranz, J., Hullermeier, E., Vanderlooy, S. Binary decomposition methods for multipartite ranking // Machine Learning and Knowledge Discovery I Databases. Lecture Notes in Computer Science, vol. 5781(1), 2006. pp. 359–374.

43. Sokolova, M., Japkowicz, N., Szpakowicz, S. Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation // Australian

Conference on Artificial Intelligence. Lecture Notes in Computer Science, vol. 4304, 2006. pp. 1015–1021.

44. Ferri, C., Hernandez-Orallo, J., Modroiu, R. An experimental comparison of performance measures for classification // Pattern Recognition Letters 30 (1), 2009. pp. 27–38.

45. Landgrebe, T., Duin, R. Efficient multiclass ROC approximation by decomposition via confusion matrix perturbation analysis // IEEE Transactions on Pattern Analysis and Machine Intelligence, no. 30 (5), 2008. pp. 810–822.

46. Cohen, J. A coefficient of agreement for nominal scales // Educational and Psychological Measurement, 20 (1), 1960. pp. 37–46.

47. Mikhalev P.A., Kutsakin M.A., Mironov O.Yu: On the need for parametric optimization of systems with federated machine learning // XXVIII International Open Conference "Modern informatization problems in simulation and social technologies" (MIP-2023’SCT), Volume 1, pp. 37-41 (January 2023).

48. Differential Privacy for Privacy-Preserving Data Analysis: An Introduction to our Blog Series // [Электронный ресурс]. – Режим доступа: https://www.nist.gov/blogs/cybersecurity-insights/differential-privacy-privacy-preserving-data-analysis-introduction-our – Дата доступа: 08.03.2023.

49. Song, S., Chaudhuri, K., Sarwate, A. Stochastic gradient descent with differentially private updates // Journal of Privacy and Confidentiality, vol. 1, no. 2, pp. 135–154, 2009.

50. Bassily, R., Smith, A., Thakurta, A. Private Empirical Risk Minimization: Efficient Algorithms and Tight Error Bounds // IEEE 55th Annual Symposium on Foundations of Computer Science. 2014, pp. 17-28.

51. Abadi, M., Chu, A., Goodfellow, I., McMahan, H., Mironov, I., Talwar, K., Zhang, L. Deep Learning with Differential Privacy // ACM SIGSAC Conference on Computer and Communications Security, 2016. pp. 308-318.

52. Friedman, J.H. Another approach to polychotomous classification // [Электронный ресурс]. – Режим доступа: http://www-stat.stanford.edu/~jhf/ftp/poly.ps.Z. – Дата доступа: 12.11.2024.

53. Platt, J.C., Cristianini, N., Shawe-taylor J. Large margin dags for multiclass classification // Advances in Neural Information Processing Systems, MIT Press, 2000, pp. 547–553.

54. Huhn, J.C., Hullermeier, E. FR3: a fuzzy rule learner for inducing reliable classifiers // IEEE Transactions on Fuzzy Systems 17 (1), 2009. pp. 138–149.

55. Fei, B., Liu, J. Binary tree of SVM: a new fast multiclass training and classification algorithm // IEEE Transactions on Neural Networks 17 (3), 2006. pp. 696–704.

56. Михалев П.А., Куцакин М.А., Ветров И.И. Подход к моделированию многоклассового классификатора системы федеративного машинного обучения, функционирующего в условиях неполноты классов локальных классификаторов // Системы управления и информационные технологии. -Воронеж: Издательство «Научная книга», №4(98). 2024, с. 26-32.

57. Mikhalev P.A., Kutsakin M.A., Mironov O.Yu. An approach to the ensembling of models of local classifiers under conditions of incompleteness of classes in systems with federated learning // XXIX-th International Open Science Conference "Modern informatization problems in simulation and social technologies" (MIP-2024’SCT). 2024, pp. 36-40.

58. Ижболдин О.В., Курляндчик Л.А. Неравенство Йенсена // Научно-популярный физико-математический журнал «Квант», №4, 2000. с. 7-10.

59. Апраушева Н. Н., Сорокин С. В. Заметки о гауссовых смесях // Издательство ВЦ РАН, 2015. С. 145.

60. Берзинь А. У. Применение модели гауссовой смеси в лингвометрических задачах // Mining journal of KSMU n.a. academic U. Asanaliev №1, 2021. с. 118-126.

61. А.В. Кугаевских, Д.И. Муромцев, О.В. Кирсанова. Классические методы машинного обучения. – СПб: Университет ИТМО, 2022. – 53 с.

62. Ledesma, R., Valero-Mora, P., Macbeth, G. The Scree Test and the Number of Factors: a Dynamic Graphics Approach // The Spanish Journal of Psychology, No. 2, 2015. pp. 1-10.

63. Categorical Cross-Entropy in Multi-Class Classification // [Электронный ресурс]. – Режим доступа: https://www.geeksforgeeks.org/deep-learning/categorical-cross-entropy-in-multi-class-classification/ – Дата доступа: 12.06.2025.

64. 8. Papernot, N, Abadi, M., Erlingsson, U., Goodfellow, I., Talwar, K. Semi-supervised knowledge transfer for deep learning from private data // The International Conference on Learning Representations (ICLR). 2017, pp. 12-28.

65. Mayakuntla, P., Ganguli, A., Smyl, D. Gaussian Mixture Model-Based Classification of Corrosion Severity in Concrete Structures Using Ultrasonic Imaging // Journal of Nondestructive Evaluation, 42:28. 2023, pp. 27-47.

66. Pribil, J., Pribilova, A., Matausek, J. Experiment with Evaluation of Quality of the Synthetic Speech by the GMM Classifier // IEICE Transactions on Information and Systems E93D(12), 2010, pp. 3368–3376.

67. Zhu, X., Wu, J., Cheng, Y., Wang, Y. GMM-Based Classification Method for Continuous Prediction in Brain-Computer Interface // 18th International Conference on Pattern Recognition (ICPR 2006), 2006, pp. 20-24.

68. Panic, B., Klemenc, J., Nagode, M. Gaussian Mixture Model Based Classification Revisited: Application to the Bearing Fault Classification // Strojniški vestnik - Journal of Mechanical Engineering 66(2020)4, pp. 215-226.

69. Щербаков О.В., Жданов И.Н., Лушин Я.А. Сверточный автоэнкодер как генеративная модель изображений для задач выделения признаков и восстановления изображений в утерянных областях // Journal of Optical Technology, Т. 82. № 8, 2015. С. 48–53.

70. Построение SIFT дескрипторов и задача сопоставления изображений // [Электронный ресурс]. – Режим доступа: https://habr.com/ru/articles/106302/ – Дата доступа: 14.05.2024.

71. Introdution to SIFT [Электронный ресурс]. – Режим доступа: https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html – Дата доступа: 14.05.2024.

72. Еськов С.С. Специальное математическое и программное обеспечение взаимного информационного согласования в системах распределенного реестра.

[Электронный ресурс]. – Режим доступа: https://cchgeu.ru/science/dissertatsionnye-sovety/dissertatsionnyy-sovet-d-212-037-13/soiskateli/eskov.php – Дата доступа: 18.08.2024.

73. Штайн, К., Ривест, Р. Алгоритмы. Построение и анализ // Издательство «Вильямс», 2019. С. 1328.

74. Liu, Y., Fan, T., Chen, T., Hu, Q., Yang, Q. FATE: An Industrial Grade Platform for Collaborative Learning With Data Protection // Journal of Machine Learning Research, no. 22, 2021. pp. 1-6.

75. Solanki, T., Rai, B., Sharma, S. Federated Learning Using Tensor Flow // Federated Learning for IoT Applications, Springer, 2022. pp. 19-29.

76. Ziller, A., Trask, A., Lopardo, A., Shymkow, B. PySyft: A Library for Easy Federated Learning // Federated Learning Systems: Towards Next-Generation AI, Springer, 2021. pp. 111-139.

77. PaddlePaddle/PaddleFL: Federated Deep Learning // [Электронный ресурс]. Режим доступа: https://github.com/PaddlePaddle/PaddleFL. – Дата доступа: 08.03.2025.

78. He, C., Li, S., So, J., Zhang, M. FedML: A Research Library and Benchmark for Federated Machine Learning // arXiv preprint. 2020, arXiv:2007.13518v4.

79. Nissim, K., Raskhodnikova, S. Smith, A. Smooth sensitivity and sampling in private data analysis // STOC '07: Thirty-ninth annual ACM symposium on Theory of computing. 2007, pp. 75-84.

80. Bassily, R., Thakkar, O., Thakurta, A. Model-Agnostic Private Learning // NeurIPS 2018 - 32nd Conference on Neural Information Processing Systems. 2018, pp. 64-75.

81. Papernot, N, Abadi, M., Erlingsson, U., Goodfellow, I., Talwar, K. Semi-supervised knowledge transfer for deep learning from private data // The International Conference on Learning Representations (ICLR). 2017, pp. 12-28.

88. pate_2018 // [Электронный ресурс]. – Режим доступа: https://github.com/tensorflow/privacy/tree/master/research/pate_2018#readme – Дата доступа: 12.06.2025.

89. PATE // [Электронный ресурс]. – Режим доступа: https://github.com/kamathhrishi/PATE/blob/master/README.md – Дата доступа: 16.06.2025.

90. Титов А.Н., Тазиева Р.Ф. Основы работы с библиотекой NumPy: учебно-методическое пособие / Минобрнауки России, Казан. нац. исслед. технол. ун-т. – Казань : Изд-во КНИТУ, 2024. –112 с.

91. Федеральный закон "О персональных данных" от 27.07.2006 N 152-Ф // [Электронный ресурс]. – Режим доступа: https://www.consultant.ru/document/cons_doc_LAW_61801/ (дата обращения 23.06.2025).

92. Советов Б.Я, Яковлев С.А. Моделирование систем. Учебник для академического бакалавриата. 7-е изд. / –М: Издательство Юрайт, 2015. — 344 с.

93. Кораблев Ю.А. Имитационное моделирование : учебник. / –М: Издательство КНОРУС, 2017. — 146 с.

94. Casaroli, A. FELES: a Federated Learning Simulator. M.Sc. Thesis // Politecnico Milano 1883, 2021. P. 221.

95. Peter Kairouz et al. “Advances and Open Problems in Federated Learning”. In: CoRR abs/1912.04977 (2019). arXiv: 1912.04977.

96. Latif U. Khan et al. “Dispersed Federated Learning: Vision, Taxonomy, and Future Directions”. In: CoRR abs/2008.05189 (2020). arXiv:2008.05189.

97. Tian Li et al. “Federated Learning: Challenges, Methods, and Future Directions”. In: CoRR abs/1908.07873 (2019). arXiv: 1908.07873.

98. G. Anthony Reina et al. “OpenFL: An open-source framework for Federated Learning”. In: CoRR abs/2105.06413 (2021). arXiv: 2105.06413.

99. The Mnist database. [Электронный ресурс]. – Режим доступа: http://yann.lecun.com/exdb/mnist/ (дата обращения 2.07.2025).

100. Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. [Электронный ресурс]. – Режим доступа: http://www.cs.toronto.edu/~kriz/cifar.html (дата обращения 17.07.2025).

101. What is TensorOpera® Federate. [Электронный ресурс]. – Режим доступа https://docs.tensoropera.ai/federate (дата обращения 19.07.2025).

102. Datasets and Models. [Электронный ресурс]. – Режим доступа https://docs.tensoropera.ai/federate/datasets-and-models (дата обращения 19.07.2025).

103. Python Agent Development framework. [Электронный ресурс]. – Режим доступа https://pade.readthedocs.io/en/latest/ (дата обращения 20.07.2025).

104. Gaussian Mixture Models Clustering – Explained. [Электронный ресурс]. – Режим доступа https://www.kaggle.com/code/vipulgandhi/gaussian-mixture-models-clustering-explained (дата обращения 23.07.2025).

105. GMM-from-scratch. [Электронный ресурс]. – Режим доступа https://github.com/Ransaka/GMM-from-scratch (дата обращения 25.07.2025).

106. Density Estimation for a Gaussian mixture. [Электронный ресурс]. – Режим доступа https://scikit-learn.org/stable/auto_examples/mixture/plot_gmm_pdf.html (дата обращения 27.07.2025).

ПРИЛОЖЕНИЕ 1

Сравнительный анализ существующих проектов

	Реализация	Разделение данных	Модель	Конфид-сть	Архитектура
FedAvg	Реализованные алгоритмы	Горизонтальное	NN	/	Централиз-я
FedSVRG			LM		
FedProx			LM, NN		
SCAFFOLD			LM, NN		
FedNova			NN		
Per-FedAvg			NN		
pFedMe			LM, NN		
IAPGD, AL2SGD+			LM		
IFCA			LM, NN		
Agnostic FL			LM, NN		
FedRobust			NN		
FedDF			NN		
FedBCD		Вертик-е			
PNFM		Горизонтальное	NN		
FedMA					
SplitNN		Вертик-е			
Tree-based FL		Горизонтальное	DT	DP	Децентрал-я
SimFL					
FedXGB				APL	
FedForest					
SecureBoost		Вертик-е			
Ridge Regression FL		Горизонтальное	LM		
PPRR					
Linear Regression FL		Вертик-е			
Logistic Regression FL					
Federated MTL					Централиз-я
Federated Meta-Learning			NN		
Personalized FedAvg					
LFRL				/	
FBO			LM		
Structure Updates	Реализованные алгоритмы	Горизонтальное	NN		
Multi-Objective FL					
On-Device ML					
Sparse Ternary Compression					
DPASGD					Децентрал-я
Client-Level DP FL				DP	
FL-LSTM					
Local DP FL			LM, NN		
Secure Aggregation FL			NN	APL	
Hybrid FL			LM, DT, NN	Гибрид	
Backdoor FL			NN		
Adversarial Lens					
Distributed Backdoor					
Image Reconstruction					
RSA			LM		
Model Poison			LM, NN		
q-FedAvg			LM, NN	/	
BlockFL			LM		Централиз-я
Reputation FL					
FedCS	Реализованные алгоритмы		NN		
DRL-MEC					
Resource-Constrained MEC			LM, NN		
FedGKT			NN		
FedCF			LM		
FedMF					
FedRecSys			LM, NN	APL	
FL Keyboard			NN		
Fraud detection			NN		
FedML	Реализованные алгоритмы	Гибрид	LM, NN	/	Гибрид
FedEval		Горизонтальное	NN		
OARF			NN	Гибрид	Централиз-я
Edge AIBench			/		Централиз-я
PerfEval			NN	/	Централиз-я
FedReID					
semi-supervised benchmark					
non-IID benchmark					/
LEAF			/		Централиз-я
Street Dataset					/

ПРИЛОЖЕНИЕ 2

Значение показателя для фиксированных классов

Таблица 2.1 – Значение показателя Accuracy для 3-х фиксированных классов (Lc=3) и последовательным изменением коэффициента Kгд для централизованной схемы

N прогона	10 бл. ЛМО	20 бл. ЛМО	30 бл. ЛМО	40 бл. ЛМО
1	0,73	0,78	0,8	0,81
2	0,71	0,8	0,82	0,84
3	0,77	0,82	0,83	0,85
4	0,72	0,8	0,81	0,83
5	0,76	0,79	0,82	0,84
Среднее	0,76	0,8	0,82	0,84
Дисп.	0,00067	0,00022	0,00013	0,00023

Таблица 2.2 – Значение показателя kappa для 3-х фиксированных классов (Lc=3) и последовательным изменением коэффициента Kгд для централизованной схемы

N прогона	10 бл. ЛМО	20 бл. ЛМО	30 бл. ЛМО	40 бл. ЛМО
1	0,69	0,73	0,75	0,79
2	0,71	0,74	0,77	0,75
3	0,68	0,76	0,73	0,76
4	0,68	0,72	0,76	0,74
5	0,69	0,74	0,75	0,78
Среднее	0,68	0,74	0,76	0,77
Дисп.	0,00015	0,00022	0,00022	0,00043

Таблица 2.3 – Значение показателя Accuracy для 3-х фиксированных классов (Lc=3) и последовательным изменением коэффициента Kгд для предлагаемой схемы

N прогона	10 бл. ЛМО	20 бл. ЛМО	30 бл. ЛМО	40 бл. ЛМО
1	0,37	0,42	0,45	0,49
2	0,34	0,44	0,47	0,51
3	0,33	0,46	0,49	0,48
4	0,35	0,42	0,44	0,47
5	0,36	0,45	0,46	0,52
Среднее	0,35	0,44	0,48	0,5

Дисп.	0,00025	0,00032	0,00037	0,00043

Таблица 2.4 – Значение показателя kappa для 3-х фиксированных классов (Lc=3) и последовательным изменением коэффициента Kгд для предлагаемой схемы

N прогона	10 бл. ЛМО	20 бл. ЛМО	30 бл. ЛМО	40 бл. ЛМО
1	0,38	0,41	0,53	0,49
2	0,42	0,38	0,51	0,53
3	0,4	0,46	0,48	0,51
4	0,39	0,43	0,5	0,5
5	0,41	0,41	0,49	0,52
Среднее	0,4	0,44	0,5	0,52
Дисп.	0,00025	0,00087	0,00037	0,00025

ПРИЛОЖЕНИЕ 3

Табличное представление значений показателей Accuracy

Таблица 3.1 – Значение показателя Accuracy для 3, 5 и 10 классов (Lc=variable) и последовательным изменением коэффициента Kгд для централизованной схемы

N прогона	10 бл. ЛМО	20 бл. ЛМО	30 бл. ЛМО	40 бл. ЛМО
1	0,47	0,5	0,54	0,63
2	0,49	0,49	0,56	0,6
3	0,54	0,54	0,5	0,59
4	0,51	0,58	0,65	0,64
5	0,55	0,57	0,65	0,69
Среднее	0,52	0,57	0,64	0,68
Дисп.	0,00112	0,00163	0,00455	0,00155

Таблица 3.2 – Значение показателя kappa для 3, 5 и 10 классов (Lc=variable) и последовательным изменением коэффициента Kгд для централизованной схемы

N прогона	10 бл. ЛМО	20 бл. ЛМО	30 бл. ЛМО	40 бл. ЛМО
1	0,39	0,42	0,52	0,57
2	0,43	0,45	0,6	0,53
3	0,4	0,5	0,58	0,62
4	0,45	0,47	0,53	0,61
5	0,42	0,48	0,54	0,54
Среднее	0,44	0,48	0,56	0,58
Дисп.	0,00057	0,00093	0,00118	0,00163

Таблица 3.3 – Значение показателя Accuracy для 3, 5 и 10 классов (Lc=variable) и последовательным изменением коэффициента Kгд для предлагаемой схемы

N прогона	10 бл. ЛМО	20 бл. ЛМО	30 бл. ЛМО	40 бл. ЛМО
1	0,17	0,23	0,29	0,34
2	0,14	0,2	0,32	0,35
3	0,2	0,25	0,31	0,39
4	0,19	0,27	0,28	0,37
5	0,21	0,29	0,32	0,4
Среднее	0,2	0,28	0,3	0,38

Дисп.	0,00077	0,00122	0,00033	0,00065

Таблица 3.4 – Значение показателя kappa для 3, 5 и 10 классов (Lc=variable) и последовательным изменением коэффициента Kгд для предлагаемой схемы

N прогона	10 бл. ЛМО	20 бл. ЛМО	30 бл. ЛМО	40 бл. ЛМО
1	0,09	0,14	0,16	0,17
2	0,11	0,16	0,18	0,23
3	0,1	0,15	0,19	0,21
4	0,13	0,2	0,22	0,25
5	0,13	0,21	0,21	0,27
Среднее	0,12	0,17	0,2	0,25
Дисп.	0,00032	0,00097	0,00057	0,00148