

Отчет о проверке на заимствования №1



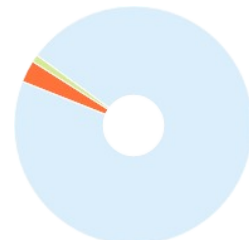
Автор: УНИВЕРИС
Проверяющий: УНИВЕРИС
Организация: Южно-Уральский государственный университет
Отчет предоставлен сервисом «Антиплагиат» - <http://susu.antiplagiat.ru>

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 336552
Начало загрузки: 01.06.2023 16:49:33
Длительность загрузки: 00:00:10
Имя исходного файла:
2023_КЭ-403_ДегтяревВА - Сору.docx
Название документа:
2023_КЭ-403_ДегтяревВА - Сору.docx
Размер текста: 31 кБ
Символов в тексте: 31749
Слов в тексте: 3926
Число предложений: 215

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Начало проверки: 01.06.2023 16:49:44
Длительность проверки: 00:01:31
Комментарии: не указано
Поиск с учетом редактирования: да
Проверенные разделы: основная часть с. 1-26, приложение с. 27-28
Модули поиска: ИПС Адилет, Библиография, Сводная коллекция ЭБС, Интернет Плюс*, Сводная коллекция РГБ, Цитирование, Переводные заимствования (RuEn), Переводные заимствования по eLIBRARY.RU (EnRu), Переводные заимствования по коллекции Гарант: аналитика, Переводные заимствования по коллекции Интернет в английском сегменте, Переводные заимствования по Интернету (EnRu), Переводные заимствования по коллекции Интернет в русском сегменте, Переводные заимствования издательства Wiley, eLIBRARY.RU, СПС ГАРАНТ: аналитика, СПС ГАРАНТ: нормативно-правовая документация, Медицина, Диссертации НББ, Коллекция НБУ, Перефразирования по eLIBRARY.RU, Перефразирования по СПС ГАРАНТ: аналитика, Перефразирования по Интернету, Перефразирования по Интернету (EN), Перефразированные заимствования по коллекции Интернет в английском сегменте, Перефразированные заимствования по коллекции Интернет в русском сегменте, Перефразирования по коллекции издательства Wiley, Патенты СССР, РФ, СНГ, СМИ России и СНГ, Шаблонные фразы, Модуль поиска "ЮУрГУ", Кольцо вузов, Издательство Wiley, Переводные заимствования



СОВПАДЕНИЯ

3,12%

САМОЦИТИРОВАНИЯ

0%

ЦИТИРОВАНИЯ

1,01%

ОРИГИНАЛЬНОСТЬ

95,87%

Совпадения — фрагменты проверяемого текста, полностью или частично сходные с найденными источниками, за исключением фрагментов, которые система отнесла к цитированию или самоцитированию. Показатель «Совпадения» — это доля фрагментов проверяемого текста, отнесенных к совпадениям, в общем объеме текста.

Самоцитирование — фрагменты проверяемого текста, совпадающие или почти совпадающие с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа. Показатель «Самоцитирования» — это доля фрагментов текста, отнесенных к самоцитированию, в общем объеме текста.

Цитирование — фрагменты проверяемого текста, которые не являются авторскими, но которые система отнесла к корректно оформленным. К цитированиям относятся также шаблонные фразы; библиография; фрагменты текста, найденные модулем поиска «СПС Гарант: нормативно-правовая документация». Показатель «Цитирования» — это доля фрагментов проверяемого текста, отнесенных к цитированию, в общем объеме текста.

Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.

Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.

Оригинальный текст — фрагменты проверяемого текста, не обнаруженные ни в одном источнике и не отмеченные ни одним из модулей поиска. Показатель «Оригинальность» — это доля фрагментов проверяемого текста, отнесенных к оригинальному тексту, в общем объеме текста.

«Совпадения», «Цитирования», «Самоцитирования», «Оригинальность» являются отдельными показателями, отображаются в процентах и в сумме дают 100%, что соответствует полному тексту проверяемого документа.

Обращаем Ваше внимание, что система находит текстовые совпадения проверяемого документа с проиндексированными в системе источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности совпадений или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в тексте	Доля в отчете	Источник	Актуален на	Модуль поиска	Блоков в отчете	Блоков в тексте	Комментарии
[01]	1,13%	1,13%	Google изобрела распределённый ИИ... http://nanonewsnet.ru	07 Апр 2017	СМИ России и СНГ	2	2	
[02]	1,09%	0,13%	Google изобрела распределённый ИИ... https://habr.com	01 Июн 2023	Интернет Плюс*	1	4	
[03]	1,01%	1,01%	не указано	29 Сен 2022	Шаблонные фразы	8	8	
[04]	0,96%	0,96%	[Перевод] Дифференциальная приват... https://pcnews.ru	06 Ноя 2020	СМИ России и СНГ	1	1	
[05]	0,89%	0,89%	ГОМОМОРФНОЕ ШИФРОВАНИЕ: ОПР... https://elibrary.ru	21 Дек 2022	eLIBRARY.RU	1	1	
[06]	0,81%	0%	010223551269_рус_В057_Жексенбек_А...	25 Мая 2023	Кольцо вузов	0	2	Источник исключен. Причина: Маленький процент пересечения.
[07]	0,78%	0%	Google изобрела распределённый ИИ... https://se7en.ws	01 Июн 2023	Интернет Плюс*	0	2	Источник исключен. Причина: Маленький процент пересечения.
[08]	0,77%	0%	Дипломная работа на тему: "Модели... https://xn--d1aux.xn--p1ai	23 Мая 2021	Интернет Плюс*	0	4	Источник исключен. Причина: Маленький процент пересечения.

[09]	0,76%	0%	КРИПТОГРАФИЧЕСКИЕ БИБЛИОТЕКИ ... https://elibrary.ru	20 Дек 2022	eLIBRARY.RU	0	1	
[10]	0,76%	0%	СРАВНЕНИЕ БАЗОВЫХ СВОЙСТВ БИБ... https://elibrary.ru	20 Дек 2022	eLIBRARY.RU	0	1	
[11]	0,76%	0%	Программные продукты и системы: н... http://biblioclub.ru	21 Янв 2020	Сводная коллекция ЭБС	0	1	
[12]	0,76%	0%	СЛЕДУЕТ ЛИ НАМ ОПАСАТЬСЯ ИСКУС... http://elibrary.ru	29 Мар 2020	Перефразирования по eLIBRARY.RU	0	1	
[13]	0,76%	0%	Шудабаев Р.М МД	19 Апр 2023	Кольцо вузов	0	1	
[14]	0,76%	0%	Facebook исследует способы «чтения... https://likeni.ru	08 Апр 2021	СМИ России и СНГ	0	1	
[15]	0,76%	0%	Сборник МММ 2019 https://sarfti.ru	07 Ноя 2022	Перефразированные заимствования по коллекции Интернет в русском сегменте	0	1	
[16]	0,76%	0%	https://izv-tn.tti.sfedu.ru:443/index.php... https://izv-tn.tti.sfedu.ru	12 Мая 2023	Перефразированные заимствования по коллекции Интернет в русском сегменте	0	1	
[17]	0,76%	0%	Full issue in PDF (5.36Mb) (8/19) http://swsys.ru	25 Дек 2014	Перефразированные заимствования по коллекции Интернет в русском сегменте	0	1	
[18]	0,76%	0%	Гомоморфное шифрование http://ru.wikipedia.org	10 Янв 2023	Перефразированные заимствования по коллекции Интернет в русском сегменте	0	1	
[19]	0,74%	0%	не указано http://dspace.susu.ru	08 Ноя 2018	Интернет Плюс*	0	5	Источник исключен. Причина: Маленький процент пересечения.
[20]	0,73%	0%	Цифровые фиатные деньги, гомомор... https://habr.com	27 Мая 2023	Интернет Плюс*	0	2	Источник исключен. Причина: Маленький процент пересечения.
[21]	0,71%	0%	СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОГРАМ... https://elibrary.ru	31 Дек 2022	eLIBRARY.RU	0	1	Источник исключен. Причина: Маленький процент пересечения.
[22]	0,69%	0%	Дифференциальная приватность — а... https://habr.com	28 Окт 2022	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[23]	0,69%	0%	Н. В. Дулина, И. А. Небыков, В. В. Тока... http://dlib.rsl.ru	03 Ноя 2017	Сводная коллекция РГБ	0	1	Источник исключен. Причина: Маленький процент пересечения.
[24]	0,69%	0%	[recovery mode] Анализ данных - осно... http://pcnews.ru	20 Дек 2018	СМИ России и СНГ	0	1	Источник исключен. Причина: Маленький процент пересечения.
[25]	0,67%	0%	Гомоморфное шифрование http://ru.wikipedia.org	10 Янв 2023	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[26]	0,67%	0%	Гомоморфное шифрование — Викип... https://ru.wikipedia.org	05 Июл 2022	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[27]	0,67%	0%	Неразличимость шифротекста — Кар... https://kartaslov.ru	10 Янв 2021	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[28]	0,66%	0%	Новый рубеж цифровых технологий. http://elibrary.ru	15 Янв 2019	Перефразирования по eLIBRARY.RU	0	1	Источник исключен. Причина: Маленький процент пересечения.
[29]	0,66%	0%	Облачные технологии: основные пон... http://swsys-web.ru	04 Дек 2018	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[30]	0,66%	0%	Облачные технологии: основные мод... http://swsys.ru	17 Дек 2022	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[31]	0,66%	0%	Облачные технологии: основные мод... http://swsys.ru	30 Сен 2020	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[32]	0,66%	0%	http://ecat.diit.edu.ua/ft/CloudTech_1.pdf http://ecat.diit.edu.ua	29 Сен 2022	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[33]	0,62%	0%	Построение логистической регресси... https://elibrary.ru	18 Апр 2023	eLIBRARY.RU	0	2	Источник исключен. Причина: Маленький процент пересечения.
[34]	0,6%	0%	https://dspace.susu.ru/xmlui/bitstream... https://dspace.susu.ru	08 Янв 2022	Интернет Плюс*	0	3	Источник исключен. Причина: Маленький процент пересечения.
[35]	0,6%	0%	[Из песочницы] Конспект по методам... https://pcnews.ru	21 Мая 2020	СМИ России и СНГ	0	2	Источник исключен. Причина: Маленький процент пересечения.
[36]	0,58%	0%	Выбросы в данных Анализ и обрабо... https://dmitrymakarov.ru	01 Июн 2023	Интернет Плюс*	0	3	Источник исключен. Причина: Маленький процент пересечения.
[37]	0,55%	0%	http://elibrary.asu.ru/xmlui/bitstream/... http://elibrary.asu.ru	11 Ноя 2022	Интернет Плюс*	0	2	Источник исключен. Причина: Маленький процент пересечения.

[38]	0,53%	0%	Дубинина, Ирина Валерьевна Реализ... http://dlib.rsl.ru	14 Ноя 2022	Сводная коллекция РГБ	0	2	Источник исключен. Причина: Маленький процент пересечения.
[39]	0,52%	0%	Конфиденциальное машинное обуче... https://habr.com	01 Июн 2023	Интернет Плюс*	0	2	Источник исключен. Причина: Маленький процент пересечения.
[40]	0,5%	0%	не указано http://dspace.susu.ru	08 Ноя 2018	Интернет Плюс*	0	3	Источник исключен. Причина: Маленький процент пересечения.
[41]	0,48%	0%	Импульсно-фазовый путевой приёмн... http://diss.natlib.uz	23 Дек 2020	Коллекция НБУ	0	2	Источник исключен. Причина: Маленький процент пересечения.
[42]	0,47%	0%	Система управления проектами терр... https://otherreferats.allbest.ru	03 Дек 2020	Интернет Плюс*	0	4	Источник исключен. Причина: Маленький процент пересечения.
[43]	0,45%	0%	Курсовая работа: Microsoft Excel, его ... https://bestreferat.ru	17 Фев 2023	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[44]	0,45%	0%	Курсовая работа: Программировани... https://itdiplom.ru	22 Мая 2022	Интернет Плюс*	0	3	Источник исключен. Причина: Маленький процент пересечения.
[45]	0,44%	0%	Разработка геоинформационной сис... https://core.ac.uk	24 Мар 2023	Интернет Плюс*	0	3	Источник исключен. Причина: Маленький процент пересечения.
[46]	0,44%	0%	Построение систем машинного обуч... http://studentlibrary.ru	20 Дек 2016	Медицина	0	1	Источник исключен. Причина: Маленький процент пересечения.
[47]	0,43%	0%	Информатика : книга для учащегося : ... http://ibooks.ru	09 Дек 2016	Сводная коллекция ЭБС	0	1	Источник исключен. Причина: Маленький процент пересечения.
[48]	0,43%	0%	"Фотоэлектрические свойства арсени... http://diss.natlib.uz	02 Сен 2014	Коллекция НБУ	0	2	Источник исключен. Причина: Маленький процент пересечения.
[49]	0,43%	0%	"Разработка физико-технических осн... http://diss.natlib.uz	02 Сен 2014	Коллекция НБУ	0	2	Источник исключен. Причина: Маленький процент пересечения.
[50]	0,43%	0%	Прогнозирование отмены брониров... https://questu.ru	03 Мая 2023	Интернет Плюс*	0	2	Источник исключен. Причина: Маленький процент пересечения.
[51]	0,43%	0%	Hyperopt с стратегией TPE всегда отб... https://de-vraag.com	01 Июн 2023	Интернет Плюс*	0	2	Источник исключен. Причина: Маленький процент пересечения.
[52]	0,39%	0%	https://dspace.susu.ru/xmlui/bitstream... https://dspace.susu.ru	04 Июн 2022	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[53]	0,38%	0%	Дементьев.pdf https://kantiana.ru	20 Авг 2019	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[54]	0,38%	0%	Коцюба, Игорь Юрьевич Методы и а... http://dlib.rsl.ru	22 Фев 2019	Сводная коллекция РГБ	0	1	Источник исключен. Причина: Маленький процент пересечения.
[55]	0,36%	0%	Устойчивое управление лесами и дин... http://dep.nlb.by	20 Дек 2016	Диссертации НББ	0	1	Источник исключен. Причина: Маленький процент пересечения.
[56]	0,36%	0%	Сорокин, Леонид Андреевич Информ... http://dlib.rsl.ru	19 Фев 2018	Сводная коллекция РГБ	0	1	Источник исключен. Причина: Маленький процент пересечения.
[57]	0,36%	0%	Научное обоснование основных нап... http://emil.ru	21 Дек 2016	Медицина	0	1	Источник исключен. Причина: Маленький процент пересечения.
[58]	0,36%	0%	scikit-learn: Сохранение и восстано... https://pythobyte.com	01 Июн 2023	Интернет Плюс*	0	2	Источник исключен. Причина: Маленький процент пересечения.
[59]	0,34%	0%	Моделирование модуля сбора данны... http://elibrary.ru	31 Авг 2017	eLIBRARY.RU	0	1	Источник исключен. Причина: Маленький процент пересечения.
[60]	0,32%	0%	2019_090304_BT_ИАИТ_Гуреев_Макси...	14 Июн 2019	Кольцо вузов	0	1	Источник исключен. Причина: Маленький процент пересечения.
[61]	0,32%	0%	Ўзбекистонда маҳаллий давлат ҳоким... http://diss.natlib.uz	16 Сен 2021	Коллекция НБУ	0	1	Источник исключен. Причина: Маленький процент пересечения.
[62]	0,32%	0%	Феноменологический анализ, диагно... http://diss.natlib.uz	12 Фев 2019	Коллекция НБУ	0	1	Источник исключен. Причина: Маленький процент пересечения.
[63]	0,32%	0%	[Перевод] Машинное обучение в Екс... https://pcnews.ru	26 Июл 2021	СМИ России и СНГ	0	1	Источник исключен. Причина: Маленький процент пересечения.
[64]	0,31%	0%	Лингводидактическая концепция фо... http://ibooks.ru	09 Дек 2016	Сводная коллекция ЭБС	0	1	Источник исключен. Причина: Маленький процент пересечения.
[65]	0,31%	0%	Русское слово в лирике XIX века. 1840... http://ibooks.ru	09 Дек 2016	Сводная коллекция ЭБС	0	1	Источник исключен. Причина: Маленький процент пересечения.

[66]	0,31%	0%	Лингводидактическая концепция фо... http://bibliorossica.com	26 Мая 2016	Сводная коллекция ЭБС	0	1	Источник исключен. Причина: Маленький процент пересечения.
[67]	0,31%	0%	https://dspace.tltsu.ru/bitstream/12345... https://dspace.tltsu.ru	20 Окт 2022	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[68]	0,29%	0%	Грудинин, Владимир Алексеевич дисс... http://dlib.rsl.ru	раньше 2011	Сводная коллекция РГБ	0	1	Источник исключен. Причина: Маленький процент пересечения.
[69]	0,26%	0%	Applications of python to evaluate the ... https://doi.org	30 Сен 2018	Издательство Wiley	0	1	Источник исключен. Причина: Маленький процент пересечения.
[70]	0,26%	0%	Applications of python to evaluate the ... https://doi.org	31 Мар 2018	Издательство Wiley	0	1	Источник исключен. Причина: Маленький процент пересечения.
[71]	0,26%	0%	Applications of Python to evaluate envi... https://doi.org	30 Ноя 2017	Издательство Wiley	0	1	Источник исключен. Причина: Маленький процент пересечения.
[72]	0,26%	0%	Study the elastic properties and the ani... https://doi.org	31 Окт 2020	Издательство Wiley	0	1	Источник исключен. Причина: Маленький процент пересечения.
[73]	0,26%	0%	Построение простой модели машинн... https://machinelearningmastery.ru	09 Июн 2022	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[74]	0,26%	0%	https://www.timacad.ru/uploads/files/2... https://timacad.ru	25 Июн 2022	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[75]	0,26%	0%	Семь регрессионных моделей - Русск... https://russianblogs.com	01 Июн 2023	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[76]	0,26%	0%	Разработка методических рекоменда... https://revolution.allbest.ru	19 Апр 2022	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[77]	0,24%	0%	Усовершенствование и анализ матем... http://diss.natlib.uz	29 Сен 2020	Коллекция НБУ	0	1	Источник исключен. Причина: Маленький процент пересечения.
[78]	0,24%	0%	http://comsec.spb.ru/thesis/2018%D0%... http://comsec.spb.ru	15 Авг 2022	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[79]	0,22%	0%	Медико-географический анализ дель... http://diss.natlib.uz	29 Авг 2014	Коллекция НБУ	0	1	Источник исключен. Причина: Маленький процент пересечения.
[80]	0,21%	0%	Трансформация законодательства о к... http://ivo.garant.ru	13 Июл 2019	СПС ГАРАНТ: аналитика	0	1	Источник исключен. Причина: Маленький процент пересечения.
[81]	0,21%	0%	не указано http://ivo.garant.ru	31 Авг 2022	СПС ГАРАНТ: аналитика	0	1	Источник исключен. Причина: Маленький процент пересечения.
[82]	0,21%	0%	Законность и унификация в уголовно... http://ivo.garant.ru	18 Янв 2020	СПС ГАРАНТ: аналитика	0	1	Источник исключен. Причина: Маленький процент пересечения.
[83]	0,21%	0%	Анализ развития регионального мед... http://ivo.garant.ru	04 Июн 2022	СПС ГАРАНТ: аналитика	0	1	Источник исключен. Причина: Маленький процент пересечения.
[84]	0,21%	0%	Приказ Федерального агентства по р... http://ivo.garant.ru	23 Апр 2011	СПС ГАРАНТ: нормативно-правовая документация	0	1	Источник исключен. Причина: Маленький процент пересечения.
[85]	0,21%	0%	https://storage.tusur.ru/files/133547/es... https://storage.tusur.ru	14 Окт 2022	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[86]	0,21%	0%	https://icst.spbstu.ru/userfiles/files/icc/... https://icst.spbstu.ru	31 Мая 2022	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[87]	0,19%	0%	Отчет по лабораторной работе 1 Кла... https://topuch.com	01 Июн 2023	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[88]	0,17%	0%	http://www.pubhealth.spb.ru/EpidD/Bo... http://pubhealth.spb.ru	01 Июн 2023	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[89]	0,16%	0%	Как начать работу с Keras, Deep Learn... https://reg.ru	01 Июн 2023	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[90]	0,16%	0%	https://perm.hse.ru/data/2018/12/15/1... https://perm.hse.ru	18 Мар 2022	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[91]	0,16%	0%	https://perm.hse.ru/data/2018/12/15/1... https://perm.hse.ru	04 Мая 2022	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.
[92]	0,15%	0%	https://aeterna-ufa.ru/sbornik/IN-2022-... https://aeterna-ufa.ru	17 Окт 2022	Интернет Плюс*	0	1	Источник исключен. Причина: Маленький процент пересечения.

ВВЕДЕНИЕ

Актуальность

В настоящее время в мире существует огромное количество информации. А также большое количество людей, которые хотят использовать эту информацию в своих научных интересах, для нахождения закономерностей, обучения нейросетей, чтобы получать ответы на важные вопросы.

Машинное обучение и анализ данных уже активно применяется в медицине, финансах, промышленности [1]. Однако эти технологии еще не могут уверенно отвечать на некоторые глобальные и сложные вопросы из-за отсутствия доступа у разработчиков и ученых к большому количеству информации. Основные причины этого выражаются в виде защиты персональных данных, сохранения приватности конкретных данных, а также раздробленности этих данных среди огромного количества организаций.

Эти проблемы можно решить с помощью систем географически-распределенного и конфиденциального машинного обучения.

Постановка задачи

Целью выпускной квалификационной работы является реализация системы для анализа географически-распределенных данных на платформе PySyft. Для достижения поставленной цели необходимо решить следующие задачи:

- 1) выполнить обзор литературы;
- 2) выполнить анализ аналогичных проектов;
- 3) определить функциональные и нефункциональные требования к системе;
- 4) спроектировать систему для анализа географически-распределенных данных на платформе PySyft;
- 5) реализовать систему для анализа географически-распределенных данных на платформе PySyft;

б) провести тестирование системы для анализа географически-распределенных данных на платформе PySyft.

Структура и содержание работы

Работа состоит из введения, четырех глав, заключения, списка литературы и приложения. Объем работы составляет 35 страниц, объем списка литературы – 20 источников.

В первой главе описываются предметная область и аналогичные проекты.

Вторая глава содержит описание теоретической части по теме работы.

Третья глава посвящена определению функциональных и нефункциональных требований к системе и проектированию ее архитектуры.

Четвертая глава содержит в себе подробности и особенности реализации системы для анализа географически-распределенных данных на платформе PySyft.

В пятой главе описывается процесс тестирования работы системы.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Описание предметной области

Целью данной работы является разработка приложения для анализа географически-распределенных данных на платформе PySyft. Анализ данных представляет собой область математики и информатики, которая занимается построением и исследованием наиболее общих математических методов и вычислительных алгоритмов извлечения знаний из экспериментальных данных с целью получения полезной информации и принятия решений [2].

Существует множество методов анализа данных. Анализ статистических данных используется для изучения свойств распределения данных [3]. Классификация данных используется для разделения данных на группы на основе их характеристик. Многомерный анализ данных используется для изучения связей между различными переменными. Регрессионный анализ используется для определения связи между зависимой переменной и одной или несколькими независимыми переменными. Кластерный анализ используется для разделения данных на группы на основе их сходства [4].

Особенностью машинного анализа географически-распределенных данных является то, что информация, которую анализирует система, распределена между различными независимыми устройствами.

Для обеспечения географически-распределенного анализа и конфиденциальности данных современные решения используют такие методы, как федеративное обучение, дифференциальная приватность, гомоморфное шифрование [5].

Федеративное обучение представляет собой метод машинного обучения, который позволяет коллективно обучать алгоритм на нескольких устройствах без централизации всех исходных данных на одном сервере. Системы федеративного обучения совершенствуют единую общую модель.

Источники данных никогда не перемещаются и не объединяются, но каждое устройство вносит свой вклад в обучение и повышение качества общей модели [6].

Дифференциальная приватность – это область, изучающая методы, которые обеспечивают максимально точные результаты статистических запросов в базу данных при минимизации возможности идентификации отдельных записей в ней. Для каждого человека, чьи данные входят в анализируемый набор, дифференциальная приватность гарантирует, что результат анализа на дифференциальную приватность будет практически неотличим вне зависимости от того, есть ли данные этого конкретного человека в наборе или нет [7]. Дифференциальная приватность основана на введении случайности в данные. Чем больше случайности добавляется, тем сильнее сохраняется приватность, однако получаются более неточные результаты. Также на точность результатов влияет размер выборки, чем больше выборка, тем точнее результаты [8].

Гомоморфное шифрование – это форма шифрования, позволяющая производить определённые математические действия с зашифрованным текстом и получать зашифрованный результат, который соответствует результату операций, выполненных с открытым текстом [9]. Использование гомоморфного шифрования открывает множество перспектив при обработке конфиденциальных данных в среде, участники которой не доверяют друг другу. Оно позволяет осуществлять индексацию, фильтрацию спама, обработку платежей и другие действия без расшифровки самих сообщений и может применяться в облачных вычислениях, децентрализованных системах, электронном голосовании [10].

1.2. Анализ существующих платформ распределенного анализа данных

1.2.1. PySyft

В 2019 году была создана библиотека PySyft сообществом OpenMined. Это люди, объединенные темой конфиденциальности в машинном обучении. PySyft представляет собой обертку над PyTorch, Tensorflow или Keras для приватного машинного обучения [8].

Основная задача, стоящая перед сообществом OpenMined, заключалась в том, чтобы создать программное обеспечение, которое бы позволяла одному человеку получать ответы на свои вопросы, используя данные, принадлежащие другому человеку без необходимости просмотра и создания копии этих данных [11].

Проект предоставляет удаленный вызов процедур, что позволяет разработчику отправлять модель нейронной сети к пользователям, где она локально обучается на их данных, после чего возвращается с обновленными весами обратно разработчику (рисунок 1). Данный процесс может происходить одновременно на разных устройствах, тем самым происходит параллельное обучение модели нейронной сети[8].

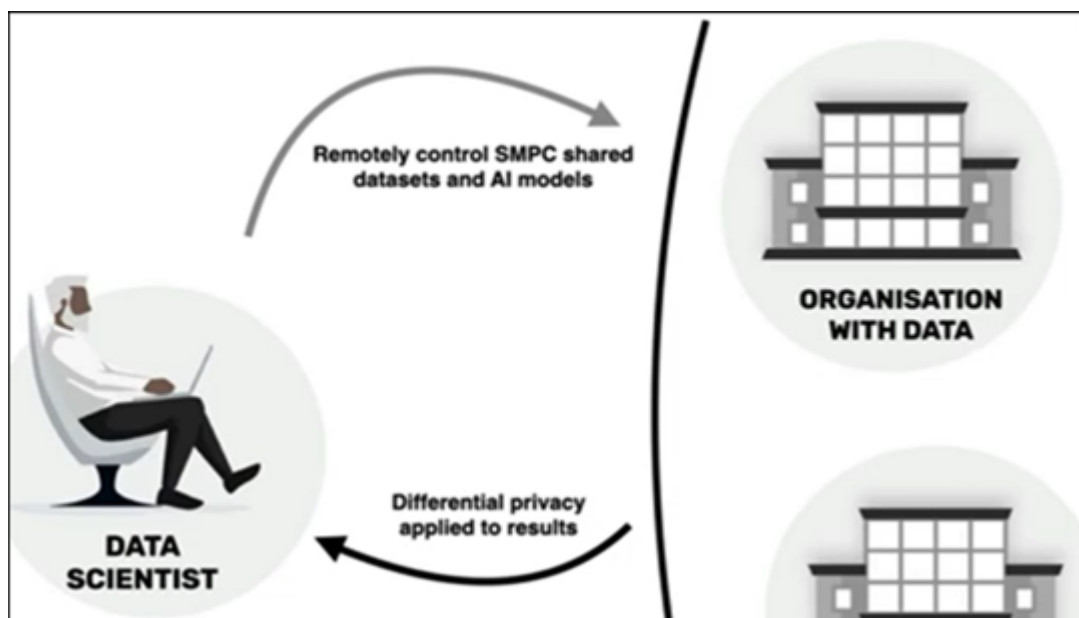


Рисунок 1 – Взаимодействие разработчика с пользователями [12]

Также ключевой особенностью PySyft является использование дифференциальной приватности [11]. По измененным весам модели нейронной сети можно догадаться, какие данные были у пользователя. Чтобы это предотвратить, к данным, которые хранятся на вычислительной машине пользователя, добавляется шум. Дифференциальная приватность представляет собой методы, которые описывают добавление шума.

1.2.2. TensorFlow Federated

Компания Google давно занимается сбором некоторой информации с устройств пользователей в единое защищенное хранилище, на котором тренируют свои нейросети. В 2017 году в Google Research был предложен инновационный подход под названием федеративное машинное обучение. Он позволяет всем устройствам, которые участвуют в машинном обучении, делить на всех единую модель для прогнозирования, но при этом не делиться первичными данными для обучения модели. Система федеративного обучения работает по принципу совершенствования единой общей модели нейросети [13].

Для проверки системы федеративного обучения на больших объемах данных компанией Google был реализован этот алгоритм в мобильном приложении клавиатуры Gboard [13]. Основной задачей являлось прогнозирование слов и выражений, которые пользователь предположительно использовал бы следующими во время печатания текста. Система федеративного обучения не отправляет текст, который печатает пользователь, на сервер компании Google, она отправляет На устройстве каждого пользователя производится анализ текста, который он использовал. Затем результаты анализа отправляются в компанию Google, где они будут объединены с другими результатами анализа для улучшения общей модели набора текста. Тем самым, каждый пользователь улучшает опыт использования Gboard каждому пользователю.

Компанией Google была создана платформа TensorFlow Federated с открытым исходным кодом для машинного обучения на децентрализованных данных. Архитектура данной платформы представлена на рисунке 2. TensorFlow Federated был разработан для облегчения исследований и экспериментов с федеративным обучением [14].

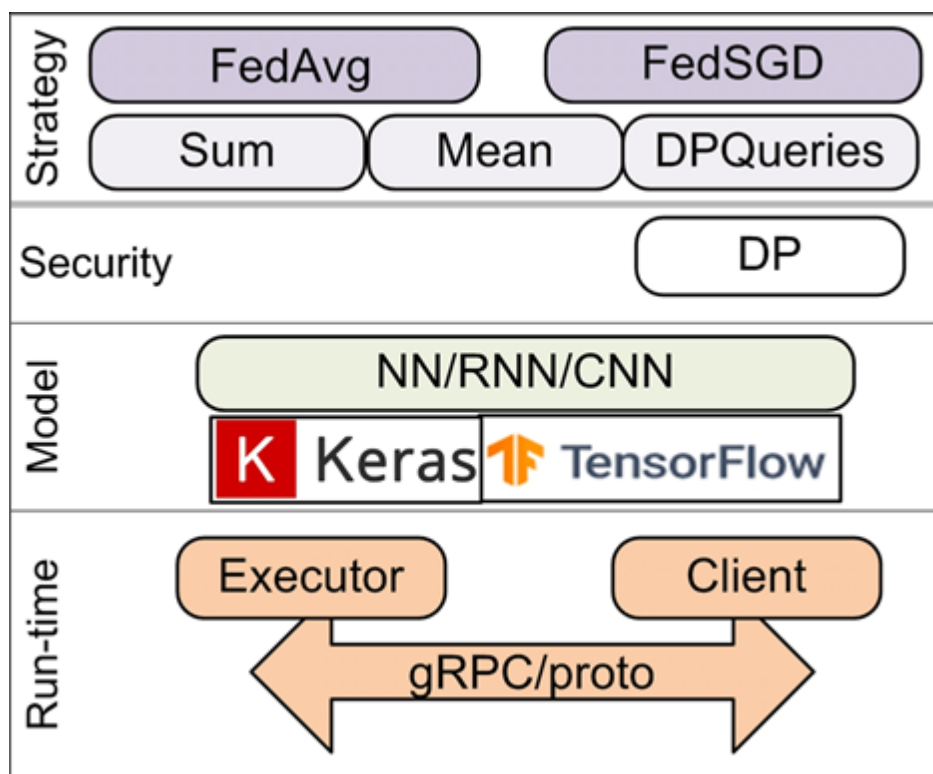


Рисунок 2 – Архитектура TensorFlow Federated [14]

Выводы по первой главе

В данной главе был проведен обзор предметной области и анализ существующих решений и проектов в области машинного анализа географически-распределенных данных.

2. ПРОЕКТИРОВАНИЕ

3

Целью данной работы является разработка системы для анализа географически-распределенных данных на платформе PySyft. Система представляет собой децентрализованное приложение, которое будет предоставлять одноранговую сеть для владельцев данных и аналитиков данных.

Владелец данных с помощью системы сможет создавать узлы внутри сети, загружать данные и управлять созданными узлами.

Аналитик данных с помощью приложения сможет подключаться к различным узлам внутри сети и проводить аналитические операции на основе данных, расположенным на этих узлах.

В рамках данной работы будет создано по одному узлу с двух разных устройств, которые соединены между собой локальной сетью. На каждый узел будут загружены данные. По локальной сети к созданным узлам будет подключаться устройство, на котором будет происходить процесс получения данных с подключенных узлов и их анализа.

2.1. Функциональные требования к системе

Можно выделить следующий набор функциональных требований к системе.

1. Система должна предоставлять владельцу данных возможность запустить узел в одноранговой сети.
2. Система должна предоставлять владельцу данных возможность отключить созданный им узел в одноранговой сети.
3. Система должна предоставлять владельцу данных возможность загрузить данные на созданный им узел в одноранговой сети.
4. Система должна предоставлять аналитику данных возможность использовать данные, размещенные на любых других узлах одноранговой сети.

2.2. Нефункциональные требования к системе

Можно выделить следующие нефункциональные требования к системе.

1. Система должна образовывать одну целостную одноранговую сеть.
2. Система должна обеспечивать аналитику данных возможность получать данные из нескольких узлов одновременно.
3. Система должна быть написана на языке программирования Python.
4. Система должна быть разработана с использованием таких инструментов, как: PySyft, PyGrid, HaGrid.

2.3. Диаграмма вариантов использования системы

Для проектирования системы был использован язык графического описания для объектного моделирования UML.

На рисунке 3 представлена диаграмма вариантов использования.



Рисунок 3 – Диаграмма вариантов использования системы для анализа географически-распределенных данных

В системе определены следующие виды акторов.

1. *Владелец данных* – это пользователь приложения, который может запустить и отключить узел в одноранговой сети для размещения на него определенных данных.

2. *Аналитик данных* – пользователь приложения, который может использовать размещенные на узлах данные для аналитических операций.

Актору «Владелец данных» доступны следующие варианты использования системы.

1. Владелец данных может запустить узел для дальнейшего размещения на нем данных.

2. Владелец данных может отключить узел, после этого, все данные, которые были размещены на узле, станут недоступными для использования аналитиками данных.

3. Владелец данных может загрузить данные на созданный им узел, которые в дальнейшем будут аннотированы дифференциальной приватностью и станут доступны аналитикам данных.

4. Владелец данных может создать нового пользователя, который будет иметь доступ к созданному владельцем узлу.

5. Владелец данных может авторизоваться в созданном им узле. Перед авторизацией владелец данных подключается к узлу.

Актору «Аналитик данных» доступны следующие варианты использования системы.

1. Аналитик данных может получить данные, которые размещены на узле.

2. Аналитик данных может авторизоваться в узле, к которому он имеет доступ. Перед авторизацией аналитик данных подключается к узлу.

2.4. Диаграмма компонентов

На рисунке 4 представлена диаграмма компонентов системы.

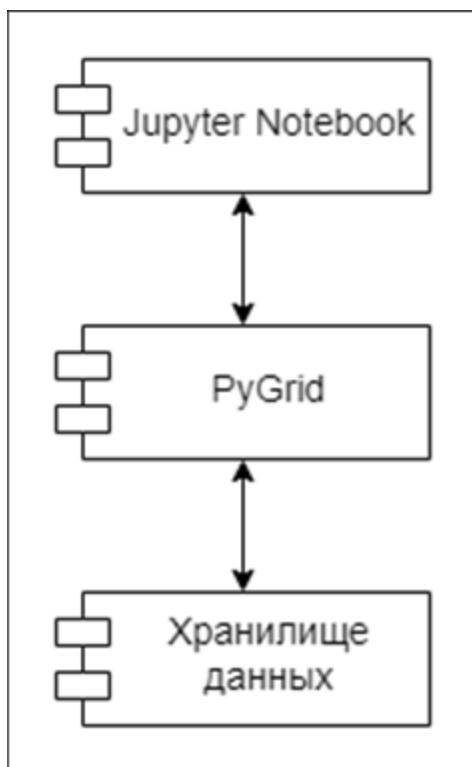


Рисунок 4 – Диаграмма компонентов системы

Система состоит из следующих компонентов.

1. Jupyter NoteBook – интерфейс, с которым взаимодействуют пользователи.
2. PyGrid – серверный компонент системы, который обрабатывает все запросы пользователей, взаимодействует с хранилищем данных узла.
3. Хранилище данных – компонент системы для хранения данных. Для каждого узла существует собственный компонент хранилища данных, в котором хранится информация о пользователях узла, данные, которые загружает владелец узла.

2.5. Диаграммы деятельности

При проектировании системы были составлены диаграммы деятельности для прецедентов «Загрузить данные» и «Получить данные».

Для прецедента «Загрузить данные» было выделено 3 компонента, взаимодействующие между собой, такие как «Владелец данных», «PyGrid», «Хранилище данных».

Для прецедента «Получить данные» было выделено 3 компонента, взаимодействующие между собой, такие как «Аналитик данных», «PyGrid», «Хранилище данных».

На рисунке 5 представлена диаграмма деятельности прецедента «Загрузить данные».

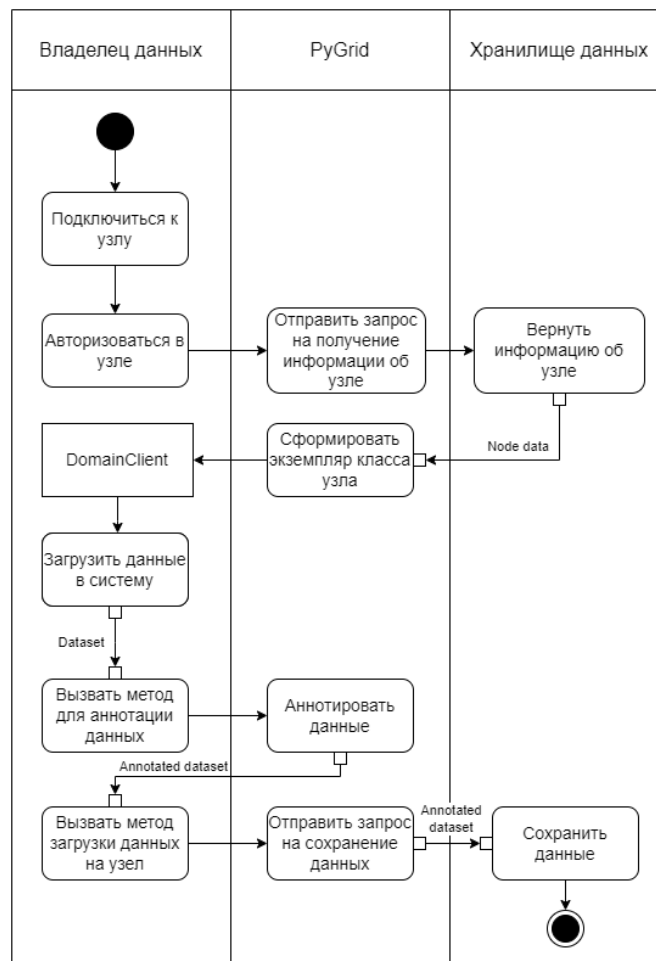


Рисунок 5 – Диаграмма деятельности прецедента

На рисунке 6 представлена диаграмма деятельности прецедента «Получить данные».

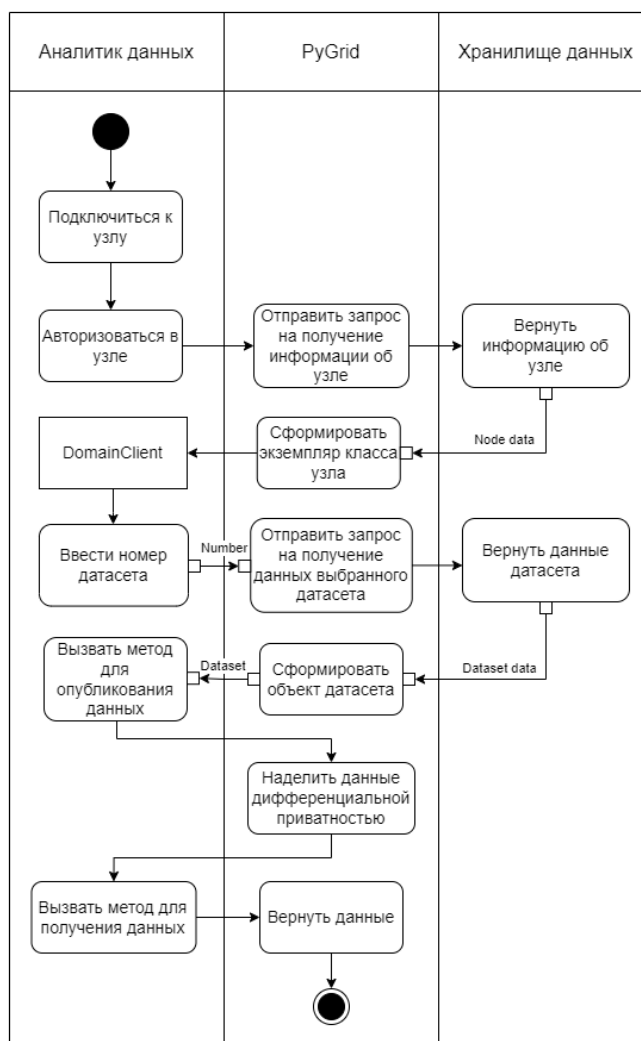


Рисунок 6 – Диаграмма деятельности прецедента

Выводы по второй главе

В данной главе были выделены функциональные и нефункциональные требования к системе, составлена диаграмма вариантов использования системы и разработаны диаграммы компонентов системы.

Узел домена занимает порт 8081. При развертывании узла домена создаются и запускаются «Docker» контейнеры. Данный процесс показан на рисунке 9.

```

Launching a PyGrid Domain node on port 8081!

- NAME: local_node
- RELEASE: production
- ARCH: linux/amd64
- TYPE: domain
- DOCKER_TAG: latest
- HAGRID_VERSION: 0.2.134
- PORT: 8081
- DOCKER COMPOSE: v2.15.1

" Launching Containers
  ✓ Pulling [9 / 9] _____ 100.00%
  ✓ Launching [11 / 11] _____ 100.00%

" Checking node API
  ✓ local_node Domain Containers Created
  ✓ Backend
  ✓ Startup Complete

```

PyGrid	Info	600
UI (beta)	http://localhost:8081/login	✓
api	http://localhost:8081/api/v1/openapi.json	✗

Рисунок 9 – Запуска узла домена владельца данных

3.2. Постановка прикладной задачи

В качестве примера работоспособности системы в рамках данной работы будет решена задачи классификации.

Цель прикладной задачи заключается в том, чтобы на основе данных о ядрах клеток опухоли молочной железы выявить, злокачественной или доброкачественной является клетка опухоли.

Для решения данной задачи будет реализована модель дерева решений.

Данные о клетках молочной железы были взяты с сайта Kaggle [20]. Это набор данных, представляющий собой таблицу с 569 строками и 31

столбцом, каждый из этих столбцов обозначает определенную характеристику ядра клетки молочной железы и имеет вещественное значение, а также столбец, который обозначает тип клетки. Для первого узла были выделены первые 250 строк набора данных, для второго узла следующие 250 строк набора данных.

3.3. Процесс загрузки данных на узел

Процесс подключения к узлу в качестве владельца данных продемонстрирован в листинге 1.

Листинг 1 – Подключение к узлу

```
try:
    domain_client = sy.login(
        port=8081,
        email="info@openmined.org",
        password="20052005"
    )
except Exception as e:
    print("Unable to login. Please check your domain is up with `!hagrid check localhost:8081 --timeout=120`")
```

Исходные данные были обработаны. Был удален пустой столбец «Unnamed». Исходный набор данных был разделен на два набора данных, так как для решения задачи нужно наделять дифференциальной приватностью все столбцы, кроме столбца, обозначающего тип клетки.

Данные были аннотированы с дифференциальной приватностью с помощью метода `annotate_with_dp_metadata` класса `Tensor`. В параметрах данного метода были указаны нижняя и верхняя границы предельно допустимых значений для конкретного набора данных. Для каждого набора данных эти значения будут различаться, например, на основе всего набора данных, для данных о среднем расстоянии от центра до точек на периметре ядра клетки (`radius_mean`) нижняя граница равняется 0, а верхняя граница 30. Также были выделены субъекты набора данных, то есть сущности, информация о которых не будет никому предоставляться. Для данного

набора данных субъектом является идентификатор (id), то есть у любой строки таблицы будет отсутствовать идентификатор (id).

Процесс аннотации представлен в листинге 2.

Листинг 2 – Аннотирование данных

```
final_dataset = dict()

data_subjects = sy.DataSubjectArray.from_objs(dataset1["id"])

final_dataset["radius_mean"] = sy.Tensor(dataset1["radius_mean"]).annotate_with_dp_metadata(
    lower_bound=0, upper_bound=30, data_subject=data_subjects
)
```

Аналогично были аннотированы все столбцы набора данных. Все аннотированные столбцы добавляются в словарь final_dataset. Исходный код представлен в приложении.

Для загрузки данных на узел был использован метод load_dataset класса DomainClient (листинг 3). В параметрах метода были указаны данные, которые будут храниться в датасете, его имя и описание.

Листинг 3 – Загрузка данных характеристик

```
domain_client.load_dataset
(
    name="testDataset_v3",
    assets=final_dataset,
    description="Our dataset contains real-valued features for cell nucleus of a breast mass. There are 31
columns and 250 rows in our dataset."
)
```

Также на узел был загружен второй набор данных, обозначающий тип клетки (листинг 4).

Листинг 4 – Загрузка данных типа клетки

```
domain_client.load_dataset
(
    name="testDataset_v3_diagnosis",
    assets={
        "diagnosis": diagnosis,
    },
    description="Dataset contains diagnosis result. There are 1 column and 250 rows in our dataset."
)
```

Для того, чтобы проверить загрузку данных на узел, выведем объект datasets класса DomainClient. Вывод объекта представлен на рисунке 10.

Idx	Name	Description	Assets	Id
[0]	testDataset_v3	Our dataset contains test data. There are 31 columns and 250 rows in our dataset.	["radius_mean"] -> Tensor ["texture_mean"] -> Tensor ["perimeter_mean"] -> Tensor ...	a4614973-9288-4a94-9019-cb06f42d7bdf
[1]	testDataset_v3_diagnosis	Dataset contains diagnosis result. There are 1 column and 250 rows in our dataset.	["diagnosis"] -> Tensor	a62e69d7-0ecb-4e49-a168-e2c94764f050

Рисунок 10 – Загруженные на узел данные

Также с профиля владельца данных был создан новый пользователь с ролью аналитика данных при помощи метода `create` объекта `users`. В параметрах метода были указаны имя пользователя, почта, пароль и количество бюджета конфиденциальности. Вызов метода представлен в листинге 5.

Листинг 5 – Создание нового пользователя

```
domain_client.users.create(
    **{
        "name": "Vladimir Fake",
        "email": "vova.degtyarev.01@mail.ru",
        "password": "20012002",
        "budget": 5000000
    }
)
```

Для того, чтобы посмотреть всех пользователей узла, выведем объект `users` (рисунок 11). В рамках данного узла существуют два пользователя. Пользователь Jane Doe является владельцем данных, а пользователь Vladimir Fake является аналитиком данных.

	id	email	name	budget	verify_key	role	added_by	website	institution
0	1	info@openmined.org	Jane Doe	5.55	950ae3021cdc01ebbf27f6d20cd7b7b72cfcd8a98ac24a...	Owner	None	None	None
1	3	vova.degtyarev.01@mail.ru	Vladimir Fake	5000000.00	9fb07148374b827d6d64854dd3bea5956e9893d0bdc0f2...	Data Scientist	Jane Doe		

Рисунок 11 – Вывод объекта `users`

Аналогичным образом были загружены данные на узел на втором устройстве.

3.4. Процесс получения данных из узла

Процесс подключения к первому узлу в качестве аналитика данных продемонстрирован в листинге 6.

Листинг 6 – Подключение к первому узлу

```
try:
    domain = sy.login(
        email="vova.degtyarev.01@mail.ru",
        password="20012002",
        url="192.168.9.251",
        port=8081
    )
except Exception as e:
    print("Unable to login. Please check the domain you are connecting to")
```

Информация о том, какие данные хранятся на подключенном узле находится в объекте `datasets` экземпляра `domain`. Вывод объекта представлен на рисунке 12.

Idx	Name	Description	Assets	Id
[0]	testDataset_v3	Our dataset contains test data. There are 31 columns and 250 rows in our dataset.	["radius_mean"] -> Tensor ["texture_mean"] -> Tensor ["perimeter_mean"] -> Tensor ...	a4614973-9288-4a94-9019-cb06f42d7bdf
[1]	testDataset_v3_diagnosis	Dataset contains diagnosis result. There are 1 column and 250 rows in our dataset.	["diagnosis"] -> Tensor	a62e69d7-0ecb-4e49-a168-e2c94764f050

Рисунок 12 – Вывод объекта `datasets`

Аналитик данных может вывести любой размещенный на узле объект данных для получения демонстрационных данных. Например, вывод объекта, представляющего собой тензор `radius_mean` датасета с идентификатором 0 представлен на рисунке 13.

```

PointerId: 0a8e7de1e4343219f0a73de11145452
Status: Ready
Representation: array([17.94603224, 22.60630224, 19.34882596, 25.08286159, 4.65265277,
29.00875206, 8.04870366, 28.11085001, 4.83519571, 6.1352799 ,
23.97794796, 8.7051875 , 6.10030723, 10.42687603, 18.06814151,
11.28615251, 7.01473726, 24.09338693, 15.39724183, 11.35194021,
16.04241274, 28.38048539, 24.66603931, 25.33437782, 14.50856207,
15.4042653 , 12.51774012, 14.26213458, 18.85097243, 1.5791226 ,
6.54525059, 5.81032108, 26.4386528 , 27.94423711, 12.63734604,
19.41617401, 7.23409298, 2.10391029, 5.33206464, 6.27430455,
6.796538 , 17.72156636, 26.08381558, 26.24965184, 22.4053002 ,
21.27271241, 5.71639799, 28.12162694, 29.64777351, 17.36799922,
7.63811632, 10.22066039, 7.86959872, 22.77555468, 12.04435367,
29.26811965, 19.09375091, 7.51706832, 2.20558554, 24.41420513,
5.49481431, 8.75549746, 12.53265704, 4.76100794, 2.87493351,
5.63586864, 1.75520347, 13.4852952 , 18.45437348, 15.81331503,
17.1943482 , 3.25638837, 9.99489773, 25.47400224, 1.83600782,
16.36718841, 13.16617764, 21.19996635, 1.20477277, 27.08679054,
1.6851673 , 22.20360007, 7.31210793, 19.62356094, 11.65317257,
0.28755454, 13.02502407, 8.81444836, 10.15966991, 21.26081294,
7.41197731, 7.6727683 , 3.95481837, 22.86123326, 3.75843934,
14.48994482, 7.30041254, 26.45911346, 21.93412543, 11.78348429,
4.75072715, 16.00674446, 25.81945984, 3.81563013, 5.73007333,
21.88747634, 19.15972833, 8.71334314, 6.82665493, 26.05915262,
10.23947107, 23.51900396, 29.52143021, 21.16029915, 11.30060467,
7.81806355, 11.04160501, 4.25095408, 29.9881201 , 23.84359668,
3.17697883, 7.07378216, 12.39502137, 24.87770696, 7.95483728,
3.66234822, 12.62038564, 4.40051431, 22.91490957, 20.94131525,
7.6785041 , 1.83255136, 2.35762395, 12.30180062, 15.67958125,
13.10617309, 6.76910578, 26.61137473, 6.43582529, 23.0727524 ,
23.25685625, 15.79752459, 17.3822609 , 26.94378218, 2.39805825,
7.92749538, 5.70888477, 17.67892132, 0.20754045, 6.77887792,
27.29303857, 11.40893802, 14.68228945, 18.1874197 , 23.74085074,
2.24324605, 26.42481147, 10.08617372, 17.93741157, 22.2393777 ,
7.40308662, 3.227899 , 26.20798667, 17.66964856, 16.48044279,
2.76947184, 26.7685282 , 9.29788868, 18.37609165, 28.62649111,
15.25653591, 7.07312084, 14.30264246, 13.32252896, 26.58863017,
25.5903568 , 4.23610929, 25.7128749 , 27.43276234, 2.08902511,
25.06095232, 10.27870292, 10.38566788, 15.75957488, 16.55353655,
10.18948134, 29.67304563, 28.91097192, 4.9768542 , 25.64263614,
4.63562362, 18.54512595, 5.84966708, 18.12476108, 23.94016862,
3.37854904, 15.80341124, 4.62390744, 23.43336839, 15.7647908 ,
4.0193538 , 25.77598446, 19.42953083, 23.78264913, 27.73470785,
6.98135723, 21.88859071, 13.08896922, 11.67161913, 7.94740625,
18.93022378, 29.32723273, 7.65182938, 16.74317814, 1.7302094 ,
4.35873082, 16.75429381, 23.24900235, 28.76859495, 16.27441272,
16.20279206, 23.08981965, 27.85106129, 20.83883202, 2.614527 ,
19.32769528, 13.22061172, 27.43945873, 22.23451775, 10.97113916,
28.84966807, 26.36489343, 28.96550841, 28.77108224, 24.8992386 ,
18.70142242, 2.49630902, 21.04196891, 27.36758049, 8.18928001,
21.76941246, 1.93521338, 28.43814021, 13.60675041, 3.86005904,
0.7421407 , 4.47988573, 29.39826699, 23.13737379, 7.20133787])

(The data printed above is synthetic - it's an imitation of the real data.)

```

Рисунок 13 – Вывод тензора radius_mean

Для того, чтобы аналитик данных смог получить данные с узла, ему нужно вызвать метод `publish` определенного объекта данных, чтобы узел наделил исходные данные этого объекта дифференциальной приватностью. Процесс преобразования данных происходит на узле, к которому подключается аналитик данных. В параметрах этого метода указывается значение сигмы. Значение сигмы определяет насколько точны будут полученные данные. Сигма определяет максимальный разброс между значениями исходных и предоставляемых данных. Преобразование данных происходит так, что к исходным значениям данных добавляется случайная величина в диапазоне от отрицательного до положительного значения сигмы. Чем меньше значение сигмы, тем более точны будут полученные

данные, и тем больше бюджета конфиденциальности будет потрачено у аналитика данных. Для каждой из характеристик ядра клетки экспериментальным путем были подобраны оптимальные значения сигм.

После того, как узел преобразовал данные, аналитик данных может получить преобразованные данные вызвав метод `get`.

Процесс получения данных о характеристиках ядер клеток аналитиком показан в листинге 7.

Листинг 7 – Получение данных

```
published_data = []
sigmas = [3, 4, 20, 100, 0.03, 0.01, 0.05, 0.03, 0.04, 0.01, 0.05, 0.15, 0.4, 5, 0.01, 0.04, 0.08, 0.01, 0.01, 0.01,
4, 5, 15, 80, 0.015, 0.01, 0.1, 0.1, 0.08, 0.02]
for asset_n, sigma_n in zip(assets, sigmas):
    feature = cancer_dataset[asset_n['name']]
    feature_public = feature.publish(sigma=sigma_n)
    feature_public.block_with_timeout(50)
    privacy_feature = feature_public.get(delete_obj=False)
    published_data.append(privacy_feature)
```

Полученные данные были обработаны и преобразованы в `DataFrame` (листинг 8).

Листинг 8 – Преобразование данных

```
published_data1 = np.asarray(published_data)
deboxed = np.array([item for item in published_data1.flatten()]).reshape(published_data1.shape)
data_df = pd.DataFrame(deboxed)
data_df = data_df.transpose()
```

Преобразованные данные продемонстрированы на рисунке 14.

	0	1	2	3	4	5	6	7	8	9	...	20	21
0	16.132178	8.434578	134.104971	1100.135552	0.087600	0.280242	0.298157	0.153714	0.226333	0.071672	...	34.266756	15.488959
1	17.078989	17.094781	127.665273	1272.185442	0.105969	0.089305	0.032212	0.048682	0.154750	0.063071	...	36.662754	23.940783
2	15.622685	15.774698	162.606136	1219.693821	0.103326	0.171460	0.152722	0.105217	0.220147	0.063774	...	18.326853	18.217385
3	10.051632	22.897328	55.129015	359.715497	0.160802	0.288066	0.231524	0.052108	0.190165	0.093511	...	12.677428	18.982321
4	20.652202	17.420749	126.831422	1330.350017	0.100585	0.117940	0.205592	0.097690	0.126316	0.048576	...	20.672949	19.742730
...
245	10.478143	17.744799	80.199168	281.126391	0.119582	0.058763	0.050669	-0.000533	0.211496	0.069394	...	7.003120	34.228511
246	13.277229	11.052537	120.697787	579.991444	0.045019	0.052506	0.023185	0.038215	0.102785	0.059122	...	20.335097	33.831021
247	13.609391	12.406311	89.819592	332.250218	0.133811	0.123512	0.107841	0.053611	0.173542	0.065869	...	12.548284	17.728222
248	9.921877	26.998829	88.430236	475.303414	0.091708	0.073301	-0.026388	0.005190	0.159686	0.054961	...	7.946641	27.365150
249	6.742087	19.200117	80.417527	363.016130	0.099238	0.067331	0.005977	0.022594	0.138886	0.057071	...	14.359081	23.946826

250 rows × 30 columns

Активация Wind

Рисунок 14 – Преобразованные данные

Также аналитиком были получены данные о типах клеток, так как эти данные не были аннотированы дифференциальной приватностью, для их получения достаточно вызвать метод `get`. Процесс получения и преобразования данных в `DataFrame` показан в листинге 9.

Листинг 9 – Получение данных

```
diagnosis_data = diagnosis_dataset["diagnosis"].get(delete_obj=False)
diagnosis_data = diagnosis_data.child

deboxed_diagnosis = np.array([item for item in diagnosis_data.flatten()]).reshape(diagnosis_data.shape)
data_df_diagnosis = pd.DataFrame(deboxed_diagnosis)
```

На операции получения данных аналитик данных потратил 311996 единиц конфиденциальности бюджета.

Аналогичным образом аналитиком данных были получены данные с узла, размещенном на втором устройстве.

3.5. Реализация прикладной задачи

Для построение модели дерева решений был использован класс `DecisionTreeClassifier` библиотеки `sklearn`. В конструкторе класса был указан критерий энтропии, который вычисляет энтропию Шеннона возможных классов [19].

После разделения данных, полученных с первого узла, на обучающие и тестовые выборки, модель была обучена на обучающей выборке с помощью метода `fit`, где обучающими входными образцами являлись данные о характеристиках ядер клеток, а целевым значением данные о типах клеток. Также на тестовых данных было подсчитана средняя точность модели с помощью метода `score`.

Перед обучением модели, она с помощью модуля `pickle` стандартной библиотеки `python` загружается из файла «`model.pkl`», а после обучения

модели на данных, она сохраняется в этот же файл. Тем самым, модель совершенствуется после каждого обучения на новом наборе данных.

Модель была повторно обучена на данных, которые были получены со второго узла.

Процесс загрузки, обучения и сохранения модели продемонстрирован в листинге 10.

Листинг 10 – Загрузка, обучение и сохранение модели

```
from sklearn import tree
from sklearn.model_selection import train_test_split
import pickle
import os.path

X = data_df
y = data_df_diagnosis

#загрузка или создание модели
if os.path.isfile('model.pkl'):
    with open("model.pkl", 'rb') as file:
        clf = pickle.load(file)
else:
    clf = tree.DecisionTreeClassifier(criterion='entropy')

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

clf.fit(X_train, y_train)

accuracy = clf.score(X_test, y_test)
print("Accuracy:", accuracy)

#сохранение модели
pkl_filename = "model.pkl"
with open(pkl_filename, 'wb') as file:
    pickle.dump(clf, file)
```

Выводы по третьей главе

В данной главе была реализована система для анализа географически-распределенных данных на платформе PySyft, было запущено два узла, на которые были загружены данные о ядрах клеток опухоли молочной железы, с другого устройства были получены эти данные, обладающие дифференциальной приватностью, также была реализована прикладная задача по построению модели дерева решений на основе полученных данных.

4. ТЕСТИРОВАНИЕ

Для тестирования системы для анализа географически-распределенных данных на платформе PySyft и функции системы наделять данные на узлах дифференциальной приватностью выведем данные, которые были загружены на узел (рисунок 15), а также данные, которые были получены аналитиком данных (рисунок 16).

radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean
17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710
20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017
19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790
11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520
20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430
...
10.48	19.86	66.72	337.7	0.10700	0.05971	0.04831	0.03070
13.20	17.43	84.13	541.6	0.07215	0.04524	0.04336	0.01105
12.89	14.11	84.95	512.2	0.08760	0.13460	0.13740	0.03980
10.65	25.22	68.01	347.0	0.09657	0.07234	0.02379	0.01615
11.52	14.93	73.87	406.3	0.10130	0.07808	0.04328	0.02929

Рисунок 15 – Загруженные на узел данные

0	1	2	3	4	5	6	7
16.132178	8.434578	134.104971	1100.135552	0.087600	0.280242	0.298157	0.153714
17.078989	17.094781	127.665273	1272.185442	0.105969	0.089305	0.032212	0.048682
15.622685	15.774698	162.606136	1219.693821	0.103326	0.171460	0.152722	0.105217
10.051632	22.897328	55.129015	359.715497	0.160802	0.288066	0.231524	0.052108
20.652202	17.420749	126.831422	1330.350017	0.100585	0.117940	0.205592	0.097690
...
10.478143	17.744799	80.199168	281.126391	0.119582	0.058763	0.050669	-0.000533
13.277229	11.052537	120.697787	579.991444	0.045019	0.052506	0.023185	0.038215
13.609391	12.406311	89.819592	332.250218	0.133811	0.123512	0.107841	0.053611
9.921877	26.998829	88.430236	475.303414	0.091708	0.073301	-0.026388	0.005190
6.742087	19.200117	80.417527	363.016130	0.099238	0.067331	0.005977	0.022594

Рисунок 16 – Полученные данные

После обучения модели на наборе данных, размещенном на первом узле, среднее значение точности модели было равно 0,88. После последующего обучения этой же модели на наборе данных, размещенном на втором узле, среднее значение точности модели было равно 0,94.

Выводы по четвертой главе

В данной главе была протестирована система для анализа географически-распределенных данных на платформе PySyft на возможность загрузки данные на узел, получения этих данных, преобразованных с дифференциальной приватностью. Также были получены результаты точности построенной модели дерева решений после обучения на данных, размещенных на двух узлах.

ЗАКЛЮЧЕНИЕ

В рамках данной работы была спроектирована и реализована система для анализа географически-распределенных данных на платформе PySyft. При этом были решены следующие задачи.

1. Выполнен обзор литературы.
2. Выполнен анализ аналогичных проектов.
3. Определены функциональные и нефункциональные требования к системе.
4. Спроектирована система для анализа географически-распределенных данных на платформе PySyft.
5. Реализована система для анализа географически-распределенных данных на платформе PySyft.
6. Проведено тестирование системы для анализа географически-распределенных данных на платформе PySyft.

ПРИЛОЖЕНИЕ. Аннотация набора данных

Листинг 1 – Аннотация набора данных

```
final_dataset = dict()

data_subjects = sy.DataSubjectArray.from_objs(dataset1["id"])

final_dataset["radius_mean"] = sy.Tensor(dataset1["radius_mean"]).annotate_with_dp_metadata(
    lower_bound=0, upper_bound=30, data_subject=data_subjects
)
final_dataset["texture_mean"] = sy.Tensor(dataset1["texture_mean"]).annotate_with_dp_metadata(
    lower_bound=8, upper_bound=42, data_subject=data_subjects
)
final_dataset["perimeter_mean"] = sy.Tensor(dataset1["perimeter_mean"]).annotate_with_dp_metadata(
    lower_bound=40, upper_bound=200, data_subject=data_subjects
)
final_dataset["area_mean"] = sy.Tensor(dataset1["area_mean"]).annotate_with_dp_metadata(
    lower_bound=100, upper_bound=3000, data_subject=data_subjects
)
final_dataset["smoothness_mean"] =
sy.Tensor(dataset1["smoothness_mean"]).annotate_with_dp_metadata(
    lower_bound=0.03, upper_bound=0.20, data_subject=data_subjects
)
final_dataset["compactness_mean"] =
sy.Tensor(dataset1["compactness_mean"]).annotate_with_dp_metadata(
    lower_bound=0.01, upper_bound=0.45, data_subject=data_subjects
)
final_dataset["concavity_mean"] = sy.Tensor(dataset1["concavity_mean"]).annotate_with_dp_metadata(
    lower_bound=0, upper_bound=0.5, data_subject=data_subjects
)
final_dataset["concave points_mean"] = sy.Tensor(dataset1["concave
points_mean"]).annotate_with_dp_metadata(
    lower_bound=0, upper_bound=0.3, data_subject=data_subjects
)
final_dataset["symmetry_mean"] = sy.Tensor(dataset1["symmetry_mean"]).annotate_with_dp_metadata(
    lower_bound=0.08, upper_bound=0.4, data_subject=data_subjects
)
final_dataset["fractal_dimension_mean"] =
sy.Tensor(dataset1["fractal_dimension_mean"]).annotate_with_dp_metadata(
    lower_bound=0.02, upper_bound=0.15, data_subject=data_subjects
)
final_dataset["radius_se"] = sy.Tensor(dataset1["radius_se"]).annotate_with_dp_metadata(
    lower_bound=0.08, upper_bound=3, data_subject=data_subjects
)
final_dataset["texture_se"] = sy.Tensor(dataset1["texture_se"]).annotate_with_dp_metadata(
    lower_bound=0.2, upper_bound=6, data_subject=data_subjects
)
final_dataset["perimeter_se"] = sy.Tensor(dataset1["perimeter_se"]).annotate_with_dp_metadata(
    lower_bound=0.5, upper_bound=30, data_subject=data_subjects
)
```

Продолжение листинга 1 приложения

```
final_dataset["area_se"] = sy.Tensor(dataset1["area_se"]).annotate_with_dp_metadata(
    lower_bound=5, upper_bound=650, data_subject=data_subjects
```

```

)
final_dataset["smoothness_se"] = sy.Tensor(dataset1["smoothness_se"]).annotate_with_dp_metadata(
    lower_bound=0, upper_bound=0.05, data_subject=data_subjects
)
final_dataset["compactness_se"] = sy.Tensor(dataset1["compactness_se"]).annotate_with_dp_metadata(
    lower_bound=0, upper_bound=0.2, data_subject=data_subjects
)
final_dataset["concavity_se"] = sy.Tensor(dataset1["concavity_se"]).annotate_with_dp_metadata(
    lower_bound=0, upper_bound=0.6, data_subject=data_subjects
)
final_dataset["concave points_se"] = sy.Tensor(dataset1["concave points_se"]).annotate_with_dp_metadata(
    lower_bound=0, upper_bound=0.1, data_subject=data_subjects
)
final_dataset["symmetry_se"] = sy.Tensor(dataset1["symmetry_se"]).annotate_with_dp_metadata(
    lower_bound=0, upper_bound=0.1, data_subject=data_subjects
)
final_dataset["fractal_dimension_se"] =
sy.Tensor(dataset1["fractal_dimension_se"]).annotate_with_dp_metadata(
    lower_bound=0, upper_bound=0.05, data_subject=data_subjects
)
final_dataset["radius_worst"] = sy.Tensor(dataset1["radius_worst"]).annotate_with_dp_metadata(
    lower_bound=5, upper_bound=50, data_subject=data_subjects
)
final_dataset["texture_worst"] = sy.Tensor(dataset1["texture_worst"]).annotate_with_dp_metadata(
    lower_bound=6, upper_bound=80, data_subject=data_subjects
)
final_dataset["perimeter_worst"] = sy.Tensor(dataset1["perimeter_worst"]).annotate_with_dp_metadata(
    lower_bound=25, upper_bound=400, data_subject=data_subjects
)
final_dataset["area_worst"] = sy.Tensor(dataset1["area_worst"]).annotate_with_dp_metadata(
    lower_bound=100, upper_bound=8000, data_subject=data_subjects
)
final_dataset["smoothness_worst"] =
sy.Tensor(dataset1["smoothness_worst"]).annotate_with_dp_metadata(
    lower_bound=0.02, upper_bound=0.50, data_subject=data_subjects
)
final_dataset["compactness_worst"] =
sy.Tensor(dataset1["compactness_worst"]).annotate_with_dp_metadata(
    lower_bound=0.01, upper_bound=2, data_subject=data_subjects
)
final_dataset["concavity_worst"] = sy.Tensor(dataset1["concavity_worst"]).annotate_with_dp_metadata(
    lower_bound=0, upper_bound=3, data_subject=data_subjects
)
final_dataset["concave points_worst"] = sy.Tensor(dataset1["concave
points_worst"]).annotate_with_dp_metadata(
    lower_bound=0, upper_bound=0.5, data_subject=data_subjects
)

```

Окончание листинга 1 приложения

```

)
final_dataset["symmetry_worst"] = sy.Tensor(dataset1["symmetry_worst"]).annotate_with_dp_metadata(
    lower_bound=0.1, upper_bound=1, data_subject=data_subjects
)
final_dataset["fractal_dimension_worst"] =
sy.Tensor(dataset1["fractal_dimension_worst"]).annotate_with_dp_metadata(
    lower_bound=0.02, upper_bound=0.5, data_subject=data_subjects
)

```