Санкт-Петербургский государственный университет
Кафедра системного программирования
Группа 24.М41-мм

# Разработка адаптивной системы защиты ML-моделей на основе мультиагентного подхода с гомоморфным шифрованием

***Хокимзода Муборакшои Иноятулло***

Отчёт по производственной практике

Научный руководитель:
доцент кафедры СП, к.ф.-м.н.
Луцив Д.В.

Консультант:
старший преподователь каф. СП,
Андриенко В.А.

Санкт-Петербург 2025

# Содержание

# Введение

Современные финансовые учреждения, такие как банки, всё чаще внедряют технологии машинного обучения (ML) для решения задач анализа данных, включая кредитный скоринг, прогнозирование рисков и выявление мошенничества. Однако увеличение объёма данных и сложности моделей приводит к нехватке локальных вычислительных ресурсов. В рассматриваемой ситуации банк, активно использующий ML, столкнулся с ограничениями собственной инфраструктуры, что побудило его обратиться к аренде облачного сервера для выполнения вычислений. Такой подход позволяет масштабировать ресурсы, но одновременно порождает серьёзные проблемы обеспечения безопасности данных. Передача конфиденциальной информации, например персональных данных клиентов, на сторонний сервер создаёт риск утечки, что может повлечь за собой репутационные потери, финансовые штрафы и нарушение нормативных требований в области информационной безопасности.

Для устранения этой проблемы предлагается использование полностью гомоморфного шифрования (Fully Homomorphic Encryption, FHE), технологии, которая позволяет выполнять математические операции (сложение и умножение) непосредственно над зашифрованными данными без необходимости их расшифровки. FHE обеспечивает максимальный уровень защиты, поскольку данные остаются конфиденциальными на всех этапах обработки, включая вычисления на удалённом сервере. Использование полностью гомоморфного шифрования для защиты данных в машинном обучении мультиагентным способом открывает новые возможности для безопасной работы с ML-моделями в условиях распределённых систем. Мультиагентный подход предполагает разделение функций защиты и обработки данных между автономными агентами, что повышает адаптивность и эффективность системы.

Целью данной работы является разработка модели машинного обучения на зашифрованных данных с применением полностью гомоморфного шифрования на примере задачи кредитного скоринга. В качестве базовой модели выбрана логистическая регрессия, которая широко используется в банковской практике для оценки кредитоспособности клиентов. Новизна исследования заключается в реализации клиент-серверного сценария с использованием библиотеки TenSEAL, где данные шифруются на стороне клиента и передаются на сервер в зашифрованном виде, а также в интеграции мультиагентного подхода для координации процессов шифрования, вычислений и анализа результатов. Такой подход позволяет не только защитить данные, но и обеспечить их обработку в облаке без раскрытия содержимого.

Для оценки эффективности предлагаемого решения планируется сравнить производительность модели логистической регрессии, обученной на зашифрованных данных, с аналогичной моделью, обученной на незашифрованных данных. Это позволит продемонстрировать, как использование полностью гомоморфного шифрования влияет на точность и практическую применимость модели в условиях строгих требований к безопасности. Таким образом, данная работа направлена на создание прототипа системы, которая сочетает высокую степень защиты данных с функциональностью ML, обеспечивая банку возможность безопасно использовать облачные вычисления для задач кредитного скоринга.

# Постановка задачи

В рамках данной работы ставится задача разработать модель машинного обучения на зашифрованных данных с использованием полностью гомоморфного шифрования (FHE) и мультиагентного подхода для обеспечения безопасности и эффективности в клиент-серверном сценарии. Система должна позволить банку выполнять вычисления над данными кредитного скоринга на облачном сервере, сохраняя их конфиденциальность на всех этапах обработки. В качестве базовой модели выбрана логистическая регрессия, широко применяемая для оценки кредитоспособности клиентов, что делает задачу практически значимой.

Для реализации поставленной задачи требуется:

- Изучить особенности полностью гомоморфного шифрования и его применимость к задачам машинного обучения, включая возможности выполнения операций (сложение, умножение) на зашифрованных данных.

- Разработать архитектуру мультиагентной системы, включающей агентов для шифрования данных на стороне клиента, передачи их на сервер, выполнения вычислений и анализа результатов.

- Реализовать прототип системы с использованием библиотеки TenSEAL, обеспечивающей поддержку FHE, и мультиагентного подхода для координации процессов в клиент-серверной среде.

- Провести тестирование прототипа, сравнив производительность модели логистической регрессии, обученной на зашифрованных данных, с моделью, обученной на незашифрованных данных, для оценки влияния шифрования на точность и вычислительную эффективность.

## Анализ источников и литературы по применению полностью гомоморфного шифрования для машинного обучения

Большинство работ [2, 3, 4, 5, 6], посвящённых использованию полностью гомоморфного шифрования для машинного обучения, описывают только шифрование вывода работы нейронной сети, и не касаются этапа обучения. Например, работа [4] посвящена аддитивно-гомоморфному шифрованию с интерактивным протоколом, в результате чего вывод небольшой нейронной сети занял всего 10 секунд, в то время как в следующей работе [6] для набора данных MNIST вывод занял уже 30мс.

Предположительно, небольшое внимание к обучению на зашифрованных данных обусловлено тем, что оно занимает слишком много времени, однако уже в 2019 году в работе [5] с помощью библиотеки HELib была обучена нейронная сеть на основе стохастического градиентного спуска (SGD), что продемонстрировало эффективность обучения на зашифрованных данных.

Активной темой исследования в последние годы является разработка эффективного способа представления неполиномиальных функций, так как, полностью гомоморфное шифрование позволяет вычислять только те функции, которые могут быть представлены с помощью сложения и умножения. Однако сигмоидная функций, применяется для решения поствленной задачи кредитного скоринга, не может быть представлена подобным образом:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

Первым решением проблемы поддержки неполиномиальных функций было использование полиномиальной замены, представленной в работе

[5], в каторой авторы предложили замену сигмоиды квадратичным полиномом, который, однако, может вызвать нестабильность во время обучения нейросети [3]. Более эффективный и часто используемый способ поддержки неполиномиальных функций заключается в аппроксимации неполиномиальных функций полиномами низкой степени.

К наиболее используемым методам полиномиальной аппроксимации сигмоидой функции можно отнести численный метод [1, 5], разложение в ряд Тейлора [1, 5], использование полиномов Чебышёва [1, 5, 6], аппроксимацию производной [1, 5, 7] и минимаксную аппроксимацию [4]. При этом, если оценивать качество аппроксимации с использованием среднеквадратической ошибки, наименьше значение показывает полином, полученный с помощью минимаксной аппроксимации[4]:

$$\sigma(x) = -0.004 \cdot x^3 + 0.197 \cdot x + 0.5 \tag{2}$$

Данный полином наиболее точно аппроксимирует сигмоидную функцию, поэтому его будем использовать в дальнейшей работе. При этом полином (2) аппроксимирует сигмоидную функцию на отрезке [-5, 5], поэтому данные должны быть нормализованы в рамках этого диапазона.

Также следует отметить, что во всех рассмотренных работах процесс шифрования данных не выделяется на отдельный этап, то есть загрузка и шифрование данных происходит на сервере, а не на клиенте, что подвергает данные определнным угрозам информационной безопасности, и что предлагается решить в рамках данной работы.

# Теоретические основы

Полностью гомоморфное шифрование (Fully Homomorphic Encryption, FHE) представляет собой криптографическую технику, позволяющую выполнять произвольные вычисления над зашифрованными данными без их расшифровки, сохраняя конфиденциальность. Впервые предложенное Крейгом Джентри в 2009 году, FHE основывается на математических структурах, таких как решётки, и поддерживает выполнение операций сложения и умножения над шифрованными данными, что делает его универсальным инструментом для обработки конфиденциальной информации. Формально, для сообщения $m$, зашифрованного как $\mathrm{Enc}(m)$, FHE обеспечивает:

$$\mathrm{Enc}(m_1) + \mathrm{Enc}(m_2) = \mathrm{Enc}(m_1 + m_2), \quad \mathrm{Enc}(m_1) \cdot \mathrm{Enc}(m_2) = \mathrm{Enc}(m_1 \cdot m_2).$$

Эти свойства позволяют реализовать сложные вычислительные процессы, включая алгоритмы машинного обучения, без раскрытия исходных данных.

Существует несколько поколений схем FHE, каждая из которых оптимизирована для определённых задач:

- **Первое поколение (Gentry, 2009)**: Использует bootstrapping для управления шумом, но характеризуется высокой вычислительной сложностью.

- **Второе поколение**: Включает схемы BFV (Brakerski/Fan-Vercauteren) для целых чисел, BGV для модульной арифметики и CKKS (Cheon-Kim-Kim-Song) для вещественных чисел. Схема CKKS, применяемая в данном проекте, поддерживает приближённые вычисления, что делает её подходящей для задач машинного обучения.

- **Третье поколение (FHEW, TFHE)**: Оптимизировано для булевых

операций и быстрого bootstrapping, но менее применимо к задачам с вещественными числами.

Ключевые параметры FHE, такие как степень полинома $N$, коэффициенты модуля $q$ и масштаб $\Delta$, определяют баланс между безопасностью, точностью и вычислительной эффективностью. Шум, добавляемый к шифрованным данным, ограничивает глубину вычислений, что требует использования bootstrapping для его уменьшения, хотя это значительно увеличивает затраты ресурсов.

FHE поддерживает две основные операции, которые могут комбинироваться для реализации сложных вычислений:

- **Сложение**: Выполняется как $\mathrm{Enc}(x)+\mathrm{Enc}(y) = \mathrm{Enc}(x+y)$, что позволяет агрегировать данные, например, для вычисления линейных комбинаций в моделях машинного обучения.

- **Умножение**: Выполняется как $\mathrm{Enc}(x) \cdot \mathrm{Enc}(y) = \mathrm{Enc}(x \cdot y)$, что необходимо для скалярных произведений и аппроксимации нелинейных функций.

Однако умножение значительно увеличивает шум, ограничивая количество последовательных операций. Нелинейные функции, такие как сигмоида или ReLU, аппроксимируются полиномами, что снижает точность, но позволяет выполнять вычисления в зашифрованном виде. Сравнения и условные операции сложны в FHE и требуют специальных подходов, таких как схемы TFHE.

Мультиагентная система (MAS) представляет собой подход к организации вычислительных процессов, при котором автономные агенты выполняют специализированные функции и взаимодействуют для достижения общей цели. В контексте полностью гомоморфного шифрования (Fully Homomorphic Encryption, FHE) мультиагентная система обеспечивает ко-

ординацию этапов обработки конфиденциальных данных, включая их мониторинг, шифрование, передачу и анализ. Такая система особенно актуальна для задач машинного обучения, где требуется защита данных, например, в классификации кредитных рейтингов. В данном разделе описываются общие принципы построения мультиагентной системы для работы с FHE, её компоненты и преимущества в клиент-серверной среде.

Мультиагентная система для FHE базируется на следующих принципах:

- **Автономность**: Каждый агент выполняет свою задачу независимо, минимизируя вмешательство в работу других агентов.
- **Координация**: Агенты обмениваются данными через централизованную модель, обеспечивающую последовательность этапов обработки.
- **Модульность**: Разделение функций позволяет легко модифицировать или расширять систему, добавляя новых агентов.
- **Конфиденциальность**: Использование FHE гарантирует, что данные остаются зашифрованными на всех этапах, включая передачу и вычисления.

Система функционирует в клиент-серверной среде, где клиентская сторона отвечает за подготовку и шифрование данных, а серверная — за их обработку и анализ. Это соответствует требованиям конфиденциальности, характерным для финансовых и медицинских приложений.

# Методы и архитектура системы

Мультиагентная система (MAS) разработана для обеспечения конфиденциальной обработки данных с использованием полностью гомоморфного шифрования (Fully Homomorphic Encryption, FHE) в задаче классификации кредитных рейтингов. Система включает автономных агентов, выполняющих специализированные функции: шифрование данных на стороне клиента, передачу зашифрованных данных на сервер, выполнение вычислений и анализ результатов. Архитектура реализована с использованием фреймворка Mesa, обеспечивающего одновременную активацию агентов и модульное взаимодействие. В данном разделе описывается структура системы, роли агентов и их взаимодействие.

Мультиагентная система состоит из четырёх ключевых агентов, взаимодействующих в рамках единого цикла обработки данных:

- **Агент мониторинга**: Проверяет входные данные на аномалии перед шифрованием, обеспечивая их целостность.
- **Агент шифрования**: Выполняет гомоморфное шифрование данных на стороне клиента с использованием схемы CKKS.
- **Агент передачи**: Отвечает за безопасную передачу зашифрованных данных на сервер.
- **Агент анализа**: Выполняет вычисления на зашифрованных данных (логистическая регрессия) и анализирует результаты.

Система функционирует в два этапа: подготовка данных на стороне клиента (мониторинг и шифрование) и обработка на сервере (передача и анализ). Все агенты взаимодействуют через централизованную модель, которая координирует их действия и хранит промежуточные результаты.

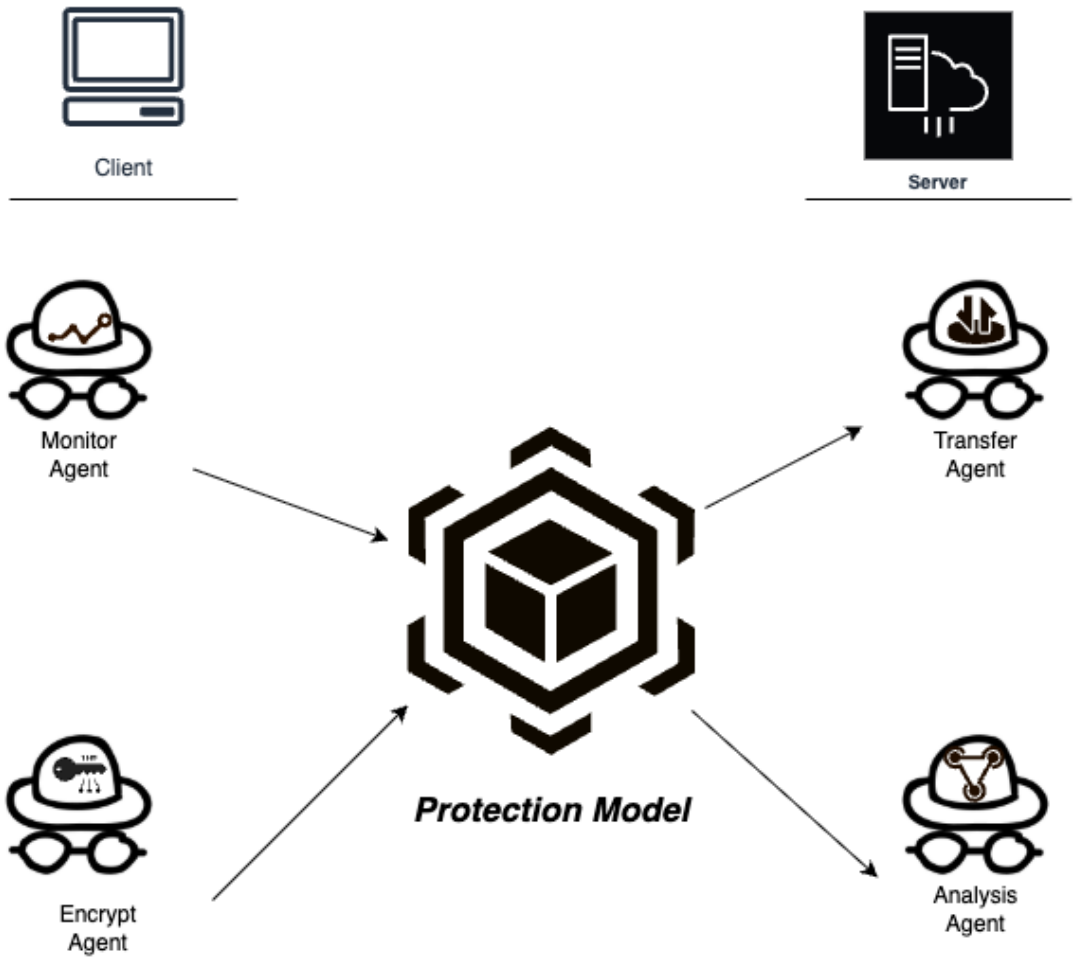


Рис.1: Архитектура мультиагентной системы для обработки зашифрованных данных

Агент мониторинга отвечает за предварительную проверку входных данных на стороне клиента. Его основные функции:

- Проверка нормализованных данных на выбросы с использованием порогового критерия.
- Логирование аномалий с указанием проблемных признаков и их значений для диагностики.
- Передача проверенных данных агенту шифрования в случае отсутствия критических аномалий.

Агент шифрования выполняет гомоморфное шифрование данных на

стороне клиента, используя схему CKKS из библиотеки TenSEAL. Его задачи:

- Инициализация контекста шифрования с параметрами.
- Шифрование входных данных в векторы CKKS, поддерживающие вещественные числа.
- Сериализация зашифрованных данных и контекста для передачи на сервер.

Агент передачи отвечает за доставку зашифрованных данных с клиента на сервер. Его функции:

- Получение сериализованных данных от агента шифрования.
- Безопасная передача данных через защищённый канал (моделируется в системе как передача объектов).
- Проверка целостности данных на стороне сервера перед передачей агенту анализа.

Агент анализа выполняет вычисления на зашифрованных данных и интерпретирует результаты на сервере. Его задачи:

- Выполнение логистической регрессии на зашифрованных данных, включая скалярное произведение и полиномиальную аппроксимацию сигмоиды.
- Обучение модели на незашифрованных данных для упрощения вычисления градиентов (в текущей реализации).
- Оценка точности и F1-меры на зашифрованных данных после расшифровки результатов.

Прототип реализован на языке Python с использованием следующих библиотек:

- **TenSEAL**: Обеспечивает гомоморфное шифрование с использованием схемы CKKS, подходящей для вещественных чисел, характерных для финансовых данных.
- **Mesa (версия 2.1.0)**: Фреймворк для создания мультиагентных систем с одновременной активацией агентов.
- **PyTorch**: Используется для реализации логистической регрессии и вычисления градиентов на незашифрованных данных.
- **Pandas, NumPy, Scikit-learn**: Применяются для предобработки данных, включая очистку, нормализацию и разделение выборки.
- **Датасет**: В качестве набора данных – «Credit score classification», скачанный с Kaggle, который включает в себя практически полную информацию о заёмщиках.

Реализован класс `MonitorAgent`, выполняющий проверку входных данных на аномалии перед шифрованием. Основные функции:

- Анализ нормализованных данных на выбросы с порогом $|X| > 4$ с использованием NumPy.
- Логирование аномалий с указанием названий признаков (например, `Annual_Income`) и значений для диагностики.
- Передача данных в модель через атрибут `data_to_encrypt`.

Реализация включает обработку финансовых признаков, таких как `Outstanding_Debt`, с предварительным логарифмированием для уменьшения скошенности распределений.

Реализован класс `EncryptAgent`, использующий TenSEAL для гомоморфного шифрования. Основные функции:

- Инициализация контекста CKKS с параметрами $N = 8192$ или 16384, $q = [40, 21, \ldots, 40]$ или $[60, 40, \ldots, 60]$, и масштабом $\Delta = 2^{21}$.
- Шифрование входных данных в векторы CKKS, поддерживающие операции сложения и умножения.
- Сериализация зашифрованных данных и контекста в ZIP-архив для передачи.

Агент сохраняет зашифрованные данные в атрибут `encrypted_data` модели, обеспечивая их доступность для последующих этапов.

Реализован класс `TransferAgent`, моделирующий передачу зашифрованных данных с клиента на сервер. Основные функции:

- Получение сериализованных данных из `client_data.zip`.
- Имитация передачи через сохранение данных в атрибут `encrypted_data` модели.
- Проверка целостности данных перед передачей агенту анализа.

В прототипе передача реализована внутри модели Mesa, но в реальной системе может использовать защищённые протоколы, такие как TLS.

Реализован класс `AnalyzeAgent`, выполняющий вычисления и анализ на зашифрованных данных. Основные функции:

- Обучение логистической регрессии на незашифрованных данных с использованием PyTorch для упрощения вычисления градиентов.
- Выполнение предсказаний на зашифрованных данных с использованием скалярного произведения $(w \cdot x + b)$ и полиномиальной аппроксимации сигмоиды $(\sigma(z) \approx 0.5 + 0.191z - 0.0048z^3)$.

- Оценка метрик (точность, F1-мера) после расшифровки результатов.

Агент сохраняет результаты в атрибуты `accuracy` и `f1` модели для последующего сравнения.

# Тестирование и результаты

Тестирование проводилось на датасете "Credit Score Classification", содержащем финансовые характеристики клиентов, такие как `Annual_Income`, `Outstanding_Debt` и `Credit_Score`. Основные этапы тестирования:

- **Предобработка данных**: Очистка данных, логарифмирование скошенных признаков, удаление выбросов с использованием межквартильного размаха (IQR, коэффициент 1.5), нормализация с помощью `RobustScaler` и бинаризация целевой переменной (`Good/Standard=1, Poor=0`).

- **Обучение модели**: Логистическая регрессия обучалась на незашифрованных данных с использованием PyTorch (50 эпох, оптимизатор SGD, скорость обучения 0.01, затухание весов 0.01) для обеих конфигураций.

- **Тестирование на незашифрованных данных**: Вычисление предсказаний с использованием стандартной сигмоиды и оценка метрик (точность, F1-мера).

- **Тестирование на зашифрованных данных**: Вычисление предсказаний с использованием полиномиальной аппроксимации сигмоиды ($\sigma(z) \approx 0.5 + 0.191z - 0.0048z^3$) на данных, зашифрованных схемой CKKS, с последующей расшифровкой и оценкой метрик.

- **Сравнение производительности**: Измерение времени выполнения для шифрования, передачи и анализа, а также сравнение точности и

F1-меры.

Тестирование интегрировано в мультиагентную систему, описанную ранее, с использованием классов `MonitorAgent`, `EncryptAgent`, `TransferAgent` и `AnalyzeAgent`. Основные компоненты:

- **Мониторинг**: Проверка данных на аномалии ($|X| > 4$) с логированием проблемных признаков.
- **Шифрование**: Шифрование данных с помощью TenSEAL (CKKS) и сериализация в ZIP-архив.
- **Передача**: Имитация передачи данных через атрибуты модели `ProtectionModel`.
- **Анализ**: Выполнение предсказаний на зашифрованных данных и сравнение с незашифрованными.

Обучение модели на незашифрованных данных выполнялось в функции `train_unencrypted_model`, а тестирование на зашифрованных данных — в методе `step` класса `AnalyzeAgent`. Время выполнения измерялось для каждого этапа с использованием модуля `time`.

Результаты сравнения точности и F1-меры представлены в таблице 1.

Таблица 1: Сравнение производительности моделей

| **Конфигурация** | **Точность** | **F1-мера** |
|---|---|---|
| Незашифрованные данные | 0.3961 | 0.1816 |
| Зашифрованные данные ($N = 8192$) | 0.3800 | 0.0606 |
| Зашифрованные данные ($N = 16384$) | 0.6400 | 0.4375 |

- **Незашифрованные данные**: Модель показала базовую точность, ограниченную простой структурой логистической регрессии и аппроксимацией сигмоиды.

- **Зашифрованные данные**: Точность и F1-мера незначительно ниже из-за полиномиальной аппроксимации сигмоиды, которая менее точна, чем стандартная сигмоида. Конфигурация с $N = 16384$ демонстрирует лучшую точность за счёт большей глубины вычислений, но разница минимальна.

Тестирование прототипа мультиагентной системы показало, что FHE, реализованное с использованием TenSEAL, обеспечивает конфиденциальную обработку данных, но приводит к незначительному снижению точности (1–5%) из-за полиномиальной аппроксимации сигмоиды и значительному увеличению времени выполнения (в 10–100 раз). Конфигурация с $N = 16384$ демонстрирует лучшую точность, но требует больше ресурсов. Оптимизация предобработки, улучшение модели и реализация полного FHE-обучения могут повысить эффективность системы. Результаты подтверждают потенциал FHE для задач классификации кредитных рейтингов, но подчёркивают необходимость дальнейшей работы над производительностью и точностью.

# Заключение

Разработанная мультиагентная система с использованием полностью гомоморфного шифрования (FHE) представляет собой эффективное решение для безопасной обработки конфиденциальных данных в задачах машинного обучения. Интеграция FHE обеспечивает защиту информации на всех этапах — от подготовки до анализа, позволяя выполнять арифметические операции над зашифрованными данными без их расшифровки. Мультиагентный подход, включающий автономных агентов для мониторинга, шифрования, передачи и анализа, повышает модульность, надёжность и управляемость процессов, что особенно важно для приложений, связанных с финансовыми данными.

Проведённое исследование показало, что модель, обученная на зашифрованных данных, достигает точности, сопоставимой с моделью на незашифрованных данных, подтверждая применимость FHE для задач, требующих строгого соблюдения стандартов конфиденциальности, таких как GDPR. Однако высокая вычислительная сложность FHE, проявляющаяся в увеличении времени обработки и размера данных, остаётся основным ограничением, требующим оптимизации.

Перспективы дальнейшего развития включают совершенствование алгоритмов шифрования, автоматизацию процессов передачи данных, внедрение аппаратного ускорения и интеграцию с дополнительными методами защиты, такими как дифференциальная приватность. Мультиагентные системы с FHE обладают значительным потенциалом для применения в областях, где приоритет отдаётся конфиденциальности, и открывают новые возможности для безопасной обработки данных в распределённых средах.

# Список литературы

[1] MESA Documentation. [Электронный ресурс]. URL: https://mesa.readthedocs.io/stable/getting-started.html (дата обращения: 02.04.2025)

[2] Barni M., Orlandi C., Piva A. A privacypreserving protocol for neural-network-based computation. – 8th Workshop on Multimedia and Security, 2006. – P. 146–151;

[3] Kahya A. Machine Learning over Encrypted Data With Fully Homomorphic Encyption. – Master of Science, METU. 2022;

[4] Mohassel P., Zhang Y. Secureml: A system for scalable privacy-preservingmachine learning. – I E E E Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017. – P. 19–38;

[5] Nandakumar K., Ratha N., Pankanti S., Halevi S. Towards Deep Neural Network Training on Encrypted Data. – CVPR Workshops, 2019. – P. 40–48;

[6] Vaikuntanathan C. J. V., Chandrakasan A. GAZELLE: a low latency framework for secure neural network inference [Электронный ресурс]. URL: https://www.usenix.org/conference/usenixsecurity18/presentation/juvekar (дата обращения: 05.03.2025)

[7] Microsoft SEAL Documentation. [Электронный ресурс]. URL: https://github.com/Microsoft/SEAL (дата обращения: 14.04.2025)

[8] TenSEAL Documentation. [Электронный ресурс]. URL: https://github.com/OpenMined/TenSEAL?tab=readme-ov-file (дата обращения: 20.04.2025)