МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Кафедра Прикладная математика и информатика
(наименование)

09.04.03 Прикладная информатика
(код и наименование направления подготовки)

Технология бизнес-анализа
(направленность (профиль))

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)

на тему Исследование технологии полностью гомоморфного шифрования для защиты данных в облачном машинном обучении

Обучающийся И.В. Хаустова
(Инициалы Фамилия) (личная подпись)

Научный руководитель канд. техн. наук, доцент, О.В. Аникина
(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2025

Оглавление

Введение

Многие современные компании активно внедряют технологии машинного обучения для решения различных бизнес-задач, например, оптимизации обработки заявок или прогнозирования возникновения задолженности у клиента. Однако эффективное построение моделей машинного обучения требует обработки большого объёма данных, что предъявляет высокие требования к вычислительным мощностям.

При этом если компания сталкивается с нехваткой собственных вычислительных ресурсов для эффективной работы с большими данными, то она может арендовать облачный сервер и получить доступ к необходимым мощным вычислительным ресурсам. Но этот подход сопровождается дополнительными рисками, связанными с безопасностью данных. Когда данные передаются на сторонний сервер, возникает угроза несанкционированного доступа, что особенно критично для конфиденциальной информации, такой как платежные истории и личные данные клиентов.

В настоящее время только технология полностью гомоморфного шифрования позволяет производить вычисления над зашифрованными данными, поэтому является лучшим выбором для обеспечения защиты данных в облачном машинном обучении. Основная идея заключается в том, чтобы позволить компаниям безопасно хранить и обрабатывать данные в облаке, не раскрывая их содержание ни провайдеру облачных услуг, ни потенциальным злоумышленникам.

На протяжении последних лет в области информационной безопасности наблюдается рост интереса к технологии полностью гомоморфного шифрования. Хотя концепция полностью гомоморфного шифрования известна давно, её широкое применение долгое время сдерживалось из-за технических трудностей и высокой вычислительной сложности. Однако с развитием вычислительных мощностей и появлением новых алгоритмов полностью гомоморфное

шифрование становится всё более привлекательным инструментом для защиты данных в облачных вычислениях.

Актуальность работы обуславливается тем, что вопрос защиты данных является крайне важным для любой компании, так как компрометация данных может привести к серьезным финансовым и репутационным потерям. При этом облачные технологии, предоставляющие компаниям мощные вычислительные ресурсы, вызывают обеспокоенность по поводу безопасности данных, особенно когда речь идет о конфиденциальной информации. В этих условиях полностью гомоморфное шифрование представляет собой уникальный механизм, позволяющий защищать данные на всех этапах их жизненного цикла, включая хранение, передачу и обработку.

Целью работы является исследование использования полностью гомоморфного шифрования для защиты данных в облачном машинном обучении.

Для достижения поставленной цели были поставлены следующие задачи:

- исследовать понятие и теоретические основы полностью гомоморфного шифрования, а также его применение в машинном обучении;
- проанализировать существующие библиотеки полностью гомоморфного шифрования;
- проанализировать бизнес-процессы, включающих в себя машинное обучение, в компании «Квартплата 24», а также собрать требования и выработать рекомендации по внедрению в него технологии полностью гомоморфного шифрования;
- разработать модель логистической регрессии, обученной на зашифрованных данных;
- оценить эффективность использования полностью гомоморфного шифрования в облачном машинном обучении;
- проанализировать выгоды от использования полностью гомоморфного шифрования.

Объектом исследования является процесс облачного машинного обучения в компании «Квартплата 24».

Предметом исследования является применение полностью гомоморфного шифрования для защиты данных в облачном машинном обучении.

Гипотеза исследования заключается в том, что применение технологии полностью гомоморфного шифрования обеспечивает высокий уровень информационной безопасности данных, используемых в облачном машинном обучении.

Теоретическая основа исследования включает изучение и использование научных трудов отечественных и зарубежных автором по теме использования полностью гомоморфного шифрования для защиты данных в машинном обучении.

Методами исследования являются анализ и синтез, сравнение, эксперимент, методы визуализации данных.

По теме исследования опубликовано 2 статьи:

- Хаустова И. В., Аникина О. В Использование полностью гомоморфного шифрования для защиты данных в машинном обучении в облаке // Вестник науки. 2024. № 4 (75). С. 482–491.
- Хаустова И. В., Аникина О. В влияние размера данных на производительность полностью гомоморфного шифрования для защиты данных в облачном машинном обучении // Вестник науки. 2024. № 11 (80) том 2. С. 1160–1168. 2024 г.

Научная новизна работы состоит в разработке модели обеспечения безопасности данных в облачном машинном обучении с помощью полностью гомоморфного шифрования в контексте адаптации и обучении модели логистической регрессии с использованием библиотеки TenSEAL. Адаптация модели для обучения на зашифрованных данных заключается в аппроксимации сигмоидной функции полиномом, полученным с помощью минимаксной аппроксимации. Также предложен механизм шифрования и упаковки данных на клиенте для их последующей передачи на сервер.

Теоретическая значимость исследования связана с углубленным изучением возможностей и ограничений полностью гомоморфного шифрования, а также с разработкой подходов к интеграции этой технологии в существующие системы обработки данных, что позволяет использовать полученные результаты в дальнейших исследованиях.

Практическая значимость исследования состоит в возможности последующего применения полученных результатов работы как компанией ООО «Квартплата 24» для решения задач машинного обучения, в частности, прогнозирования задолженности по услугам ЖКХ, а также другими компаниями, использующими облачные сервера для машинного обучения, для организации безопасности конфиденциальных данных, обрабатываемых в облаке.

Основные положения, выносимые на защиту:

- модель обеспечения безопасности данных в облачном машинном обучении с помощью полностью гомоморфного шифрования, реализованная в контексте адаптации и обучения модели логистической регрессии на зашифрованных данных с использованием библиотеки TenSEAL для решения задачи прогнозирования задолженности по услугам ЖКХ;
- результаты оценки эффективности использования полностью гомоморфного шифрования, подтверждающие эффективность его использования для обеспечения безопасности данных в облачном машинном обучении.

Объём и структура диссертации: диссертационное исследование состоит из введения, 4 глав, заключения библиографии (54 наименования). Работа изложена на 96 страницах, содержит 35 рисунков и 14 таблиц.

Во введении обосновывается актуальность выбранной темы исследования, определяется объект, предмет и цель исследования, выдвигается гипотеза и формулируются задачи работы, рассматриваются научная новизна исследования.

В первой главе описываются теоретические аспекты полностью гомоморфного шифрования, проанализированы существующие библиотек и особенности его использования в облачном машинном обучении, рассмотрены особенности представления сложных функций в гомоморфных вычислениях.

Во второй главе проведён анализ бизнес-процессов в компании «Квартплата 24», выявляется и производится моделирование процесса машинного обучения, включающего в себя полностью гомоморфное шифрование данных.

В третьей главе производится реализация системы использования полностью гомоморфного шифрования в облачном машинном обучении и разрабатывается модель логистической регрессии, обученная на зашифрованных данных.

В четвёртой главе производится оценка эффективности использования полностью гомоморфного шифрования в облачном машинном обучении с точки зрения удовлетворения требований бизнеса.

Глава 1 Теоретические аспекты защиты данных в облачном машинном обучении

1.1 Способы обеспечения безопасности данных в облачном машинном обучении

Машинное обучение приобретает всё большее значение в современных бизнес-процессах. Компании используют его для анализа больших объемов данных, прогнозирования поведения клиентов, оптимизации процессов и многого другого. Однако обучение алгоритмов машинного обучения требует как большое количество данных, безопасность которых необходимо обеспечивать, так и значительные вычислительные мощности для их обработки.

Закупка собственного оборудования для машинного обучения позволяет полностью настроить аппаратные и программные средства под собственные нужды и может гарантировать значительный контроль над безопасностью данных, так как они не передаются на сторонний сервер.

Тем не менее такой подход может оказаться выгодным только при интенсивном и долгосрочном использовании, так как требует значительных затрат, как на начальном этапе, так и в процессе эксплуатации, а также ограничивает гибкость масштабирования, что может оказаться проблемой при росте объемов данных или изменении требований к производительности.

Но если компания не очень интенсивно использует машинное обучение и, например, реализует подход с обучением модели на сервере и использованием на локальном компьютере, то она арендует для этих целей облачный сервер. Он позволяет использовать мощные вычислительные ресурсы без необходимости инвестировать в дорогостоящее оборудование, а также обеспечивает гибкость и масштабируемость, позволяя быстро увеличивать или уменьшать вычислительные мощности в зависимости от текущих потребностей. Это снижает затраты и ускоряет процессы разработки и использования моделей.

Однако аренда облачных серверов сопряжена с риском утраты контроля над конфиденциальными данными. Чтобы минимизировать эти риски, существуют программные способы обеспечения безопасности данных.

Таким образом, можно выделить два основных решения проблемы обеспечения безопасности данных в облачном машинном обучении – использование программного обеспечения безопасности данных и полный отказ от использования облачного сервера и закупка собственного оборудования, как показано на рисунке 1.

Рисунок 1 – Способы обеспечения безопасности данных в облачном машинном обучении

В настоящее время в защите данных выделяют две основные области: симметричное и ассиметричное шифрование.

Симметричное шифрование, схема которого приведена на рисунке 2, заключается в том, что обе стороны при обмене данными имеют одинаковые ключи для шифрования и расшифровки данных.

Асимметричное шифрование, схема которого приведена на рисунке 3, в свою очередь, предполагает использование для этих целей двух ключей: открытого и закрытого [1, 4].

Рисунок 2 – Симметричное шифрование

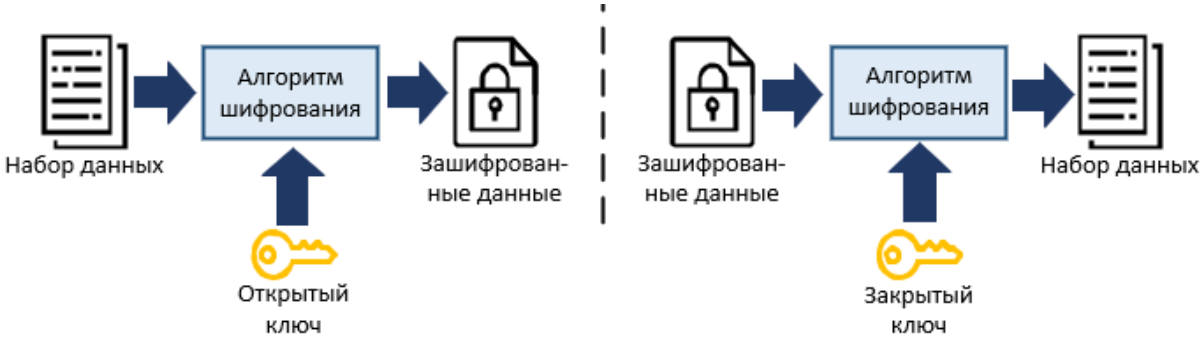

Рисунок 3 – Ассиметричное шифрование

Оба вида шифрования позволяют обеспечить безопасность хранения и передачи данные, однако машинное обучение требует проведения вычислений над данными, загруженными на сервер, что невозможно обеспечить, используя симметричное или ассиметричное шифрование, так как для работы с данными их требуется расшифровать, то есть раскрыть их содержимое.

Современные тенденции в области обеспечения безопасности данных в машинном обучении включают в себя использование различных методов для обеспечения безопасности и конфиденциальности данных. Например, многосторонние вычисления позволяют нескольким участникам совместно выполнять вычисления над общими данными без раскрытия этих данных друг другу

[14], а концепция дифференциальной приватности позволяет уменьшить связь данных с пользователями за счёт создания ограничений [15].

Такие методы достаточно практичны и могут быть легко интегрированы в существующие системы машинного обучения, но при их использовании содержание данных в том или ином виде всё равно раскрывается перед сервером.

Предотвратить раскрытие данных становится возможным только при выполнении вычислений именно над зашифрованными данными, что возможно только с использованием полностью гомоморфного шифрование. Оно позволяет выполнять вычисления, в частности, сложение и умножение, над зашифрованными данными без их расшифровки, что обеспечивает наивысший уровень безопасности [16], так как содержание данных никогда не раскрывается, как показано на рисунке 4.

Рисунок 4 – Полностью гомоморфное шифрование

Такое свойство гомоморфного шифрования, из-за которого его также называют «святым Граалем криптографии» [37], делает его применение лучшим выбором для защиты данных в машинном обучении, так как делает возможным обучение моделей на зашифрованных данных.

«Основная роль использования гомоморфного шифрования в машинном обучении заключается в шифровании конфиденциальных данных. Также оно может быть полезно при совместном обучении модели, когда ни одна из сторон не раскрывает свои данные перед другими» [40]. Кроме того, одна сторона

может отправить зашифрованные данные другой, которая получит результат, который в зашифрованном же виде отправит обратно. В этом случае гарантируется получение точных прогнозов при исключении риска раскрытия содержания данных.

Однако «гомоморфное шифрование может быть вычислительно затратным и требовать особых методов обработки данных и обучения моделей. Также существует ограничение на типы операций, которые можно выполнять над зашифрованными данными» [36], что может повлиять на производительность в сравнении с обработкой незашифрованных данных.

В целом, использование гомоморфного шифрования в машинном обучении обычно рассматривается как компромисс между обеспечением конфиденциальности данных и эффективностью вычислений, а также имеет огромный потенциал.

1.2 Определение полностью гомоморфного шифрования

Полностью гомоморфное шифрование позволяет производить арифметические операции с зашифрованными текстами так, как они бы производились над открытыми текстами [2], при этом свойство гомоморфизма должно выполняться относительно как сложения, так и умножения.

«Если свойство гомоморфизма выполняется только относительно какой-либо одной операции, то такие схемы называются частично гомоморфными, а если схема разрешает только какое-то ограниченное количество алгебраических операций» [51] – несколько гомоморфными.

В общем виде схема полностью гомоморфного шифрования представляет собой набор следующих функций.

Функция, генерирующая ключи для шифрования и вычислений, а также секретный ключ по параметру уровня криптографической стойкости (1):

$$KeyGen(1^{\lambda}) \rightarrow (pk, evk, sk) \quad (1)$$

где pk – ключ шифрования;

evk – ключ для вычислений;

sk – секретный ключ;

λ – параметр уровня криптографической стойкости.

Функция шифрования открытого текста (2):

$$Encrypt(pk, m) \rightarrow c \tag{2}$$

где m – открытый текст;

c – зашифрованный текст.

Функция расшифровки зашифрованного текста (3):

$$Decrypt(sk, c) \rightarrow m \tag{3}$$

где m – открытый текст;

c – зашифрованный текст.

Функция, обеспечивающая гомоморфные вычисления над набором зашифрованных текстов (4):

$$Evaluate(evk, F, c_1 \dots, c_n) \rightarrow c_f \tag{4}$$

где $c_1 \dots, c_n$ – набор зашифрованных текстов;

F – функция, применяемая к набору зашифрованных текстов.

Тогда соблюдение свойства гомоморфизма относительно сложения и умножения для схемы полностью гомоморфного шифрования можно представить в виде следующей системы (5):

$$\begin{cases} Decrypt\big(Encrypt(m_1)\big) \oplus Decrypt\big(Encrypt(m_2)\big) = m_1 + m_2 \\ Decrypt\big(Encrypt(m_1)\big) \otimes Decrypt\big(Encrypt(m_2)\big) = m_1 \cdot m_2 \end{cases} \tag{5}$$

То есть результат вычислений над зашифрованными данными равен результату вычислений над расшифрованными данными.

Причём схема гомоморфного шифрования будет корректной только в том случае, если после произведения гомоморфных вычислений над зашифрованными данными, расшифрование их результата совпадает с результатов этих же вычислений над открытыми текстами, т.е. выполняется условие (6):

$$Decrypt\big(sk, Evaluate(evk, F, c_1 \dots, c_n)\big) = F(m_1, \dots, m_n) \qquad (6)$$

«Таким образом, можно сформулировать определение схемы полностью гомоморфного шифрования как набора криптографических алгоритмов, позволяющих выполнять произвольные вычисления над зашифрованными данными без необходимости их расшифровки, при этом сохраняя конфиденциальность исходных данных. Это свойство делает полностью гомоморфное шифрование мощным инструментом» [23] для обеспечения безопасности данных в облачном машинном обучении.

1.3 Вычислительные задачи, лежащие в основе схем полностью гомоморфного шифрования

1.3.1 Формулировка задач LWE и RLWE

В основе схем полностью гомоморфного шифрования могут лежать задачи обучения с ошибками (англ. «learning with errors», LWE) и обучения с ошибками в кольце (англ. «ring learning with errors», RLWE), так как их вычислительная сложность позволяет обеспечить достаточный уровень защиты даже от квантовых угроз.

Для применения в области защиты информации в основе задачи обучения с ошибками лежит идея представления секретной информации (ключа) в виде системы линейных уравнений с ошибками над кольцом вычетов. Внесе-

ние ошибок делает невозможным её решение известными методами за приемлемое время, т.е. задача удовлетворяет требованию криптографической безопасности [44].

Параметрами задачи LWE являются:

- размерность пространства, $n \geq 1$;
- модуль $q \geq 2$, принимаемый с полиномиальной зависимостью от n;
- кольцо вычетов $\mathbb{Z}_q$ по модулю q;
- функция распределения вероятности ошибки $\chi_e \in \mathbb{Z}_q$.

Решение задачи LWE заключается в нахождении секретного ключа – вектора $\vec{s}$ (секретного ключа), произвольно выбранного на кольце вычетов $\mathbb{Z}_q$, на котором определено множество случайных векторов A, множество ошибок E, а также множество B, каждый элемент которого находится как скалярное произведение векторов по формуле (7):

$$b_i = \overrightarrow{a_i} \cdot \overrightarrow{s_i} + e_i, \qquad i = 1, 2, \dots, m \tag{7}$$

где $\overrightarrow{s_i}$ – секретный ключ, вектор, произвольно выбранный на кольце вычетов $\mathbb{Z}_q$;

e_i – ошибка, вектор из множества случайных неизвестных векторов из множества ошибок $E = \{e_1, e_2, \dots, e_m\}$;

$\overrightarrow{a_i}$ – вектор из множества случайных известных векторов $A = \{\overrightarrow{a_1}, \overrightarrow{a_2}, \dots, \overrightarrow{a_m}\}$.

Тогда распознавательная задача LWE сводится к поиску секретного ключа – вектора $\vec{s}$ по набору пар $(\overrightarrow{a_i},\ b_i)$.

На основе задачи «LWE была сформулирована более общая задача RLWE, в которой вместо кольца вычетов используется кольцо многочленов, а вместо скалярного произведения – умножение в кольце» [3], что увеличивает возможности шифрования и значительно сокращает длину ключа.

Параметрами задачи RLWE являются:

– размерность кольца n;
– модуль кольца q;
– факторкольцо многочленов R;
– конечное поле для модуля q, R_q;
– функция распределения вероятности ошибки χ_e;
– функция распределения вероятности секретного ключа χ_s.

Решение задачи RLWE заключается в нахождении полинома $s(x)$ (секретного ключа) относительно границы b, в кольце R, на котором определено множество случайных известных полиномов A, множество случайных неизвестных полиномов E, а также множество B, каждый элемент которого находится как умножение в кольце по формуле (8):

$$b_i = (a_i(x) \cdot s(x) + e_i(x)), \qquad i = 1, 2, \dots, m \tag{8}$$

где $s(x)$ – секретный ключ, произвольный полином в кольце R;

$e_i(x)$ – ошибка, полином из множества случайных известных полиномов $E = \{e_1(x), e_2(x), \dots, e_m(x)\}$;

$a_i(x)$ – полином из множества случайных известных полиномов $A = \{a_1(x), a_2(x), \dots, a_m(x)\}$.

Тогда распознавательная задача RLWE сводится к поиску полинома $s(x)$ по набору полиноминальных пар $(a_i(x), b_i(x))$ [44].

1.3.2 Параметры безопасности задач LWE и RLWE

Для большинства схем полностью гомоморфного шифрования, использующих задачи LWE и RLWE, распределением вероятности ошибки χ_e является нормальное распределение с математическим ожиданием $\mu = 0$ и среднеквадратичным отклонением $\sigma = \frac{8}{\sqrt{2\pi}} \approx 3.2$ [9].

Распределение вероятности секретного ключа χ_s для задачи RLWE может быть равномерным или разреженным, в зависимости от требований к уровню безопасности [10].

Целое число n, представляющее собой размерность пространства или кольца, предполагается рассматривать как степень двойки, а модуль q принимается с полиномиальной зависимостью от n [10].

Однако значения параметров безопасности может меняться не только в зависимости от схемы полностью гомоморфного шифрования, но и от библиотеки.

Также следует отметить, что практически любую задачу можно решить, если имеется достаточная вычислительная мощность или достаточно времени, например, перебирая все возможные решения, пока не будет найдено правильное (этот подход известен как атака грубой силы). Учитывая это, целью доказуемой безопасности является затем оценить количество усилий, необходимых для решения определенной проблемы (т.е. найти секретный ключ).

Схема полностью гомоморфного шифрования будет считаться безопасной, если её решение потребует невозможных в обычных условиях усилий, например, недостижимого уровня вычислительной мощности, что формализовано оценивается с помощью параметра криптографической стойкости λ, который используется при гомоморфном шифровании в алгоритме генерации ключей. Если для решения задачи, лежащей в основе схемы, требуется не менее 2^{λ} операций, то можно сказать, что λ – это уровень безопасности схемы.

1.3.3 Методы решения задач LWE и RLWE

Существует большое множество методов решения задач LWE, так как решение задачи RLWE обычно сводится к решению задачи LWE из-за того, что структура кольца многочленов не оказывает влияние на решение.

Методы решения задач LWE могут быть основаны на алгоритмах BKW и Arora-Ge, а также теории решеток.

Алгоритм BKW или алгоритм Блам-Калай-Вассермана [7] предполагает добавление к методу Гаусса промежуточного шага, на котором «подбор значений осуществляется таким образом, чтобы распределение ошибки было наиболее близко к ожидаемому.

Алгоритм Арора-Ге [8, 12], предлагает исключить шум из системы уравнений, основываясь на том, что для, если для какого-то натурального числа t ошибка лежит в интервале $[-t, t]$, то системы уравнений (9) и (10), равносильны» [3]:

$$\overrightarrow{a_i} \cdot \overrightarrow{s_i} - b_i = e_i \quad (9)$$

$$P(\overrightarrow{a_i} \cdot \overrightarrow{s_i} - b_i) = 0 \quad (10)$$

где $P(z) = \prod_{i=-t}^{t}(z - i)$ – многочлен с корнями $[-t, t]$.

«С учётом этого, система уравнений решается относительно всех одночленов, принятых за переменные.

Но наиболее эффективными методами решения задачи LWE являются методы, основанные на сведении системы линейных уравнений» [3] к задаче теории решеток, для чего система приводится к базису, например, с помощью блочного алгоритма Коркина-Золотарева [48], после чего решение задачи LWE заключается в решении одной из следующих задач:

- задача CVP/BDD (англ. «closes vector problem/Bounded Distance Decoding») – задача поиска ближайшего вектора решётки [36];
- задача SIS (англ. «short integer solution problem») – задача поиска короткого вектора на решётке в окрестности точки [43];
- задача uSVP (англ. «unique shortest vector problem») – задача поиска единственного кратчайшего вектора решётки [9].

1.4 Способы борьбы с накоплением ошибки в схемах полностью гомоморфного шифрования

Безопасность схемы полностью гомоморфного шифрования заключается в использовании определенного уровня шума, который можно удалить только путем расшифровки, и который может ограничить количество опера-

ций над зашифрованным текстом, так как ошибка растет с каждой гомоморфной операцией, и когда она достигнет порога, расшифрование станет невозможным.

Поэтому для поддержания ошибки ниже порогового уровня некоторые схемы, например, CKKS предлагают возможность использования Bootstrapping или начальной загрузки, что делает схему самонастраиваемой, т. е. способной гомоморфно вычислять собственную функцию дешифрования [26].

Основной частью процесса начальной загрузки является функция перешифрования $Decrypt_{\varepsilon}$, позволяющая уменьшить шум в зашифрованных текстах. Её можно применять неограниченное количество раз для получения новых зашифрованных текстов, что гарантирует правильную расшифровку зашифрованного текста после неограниченного числа операций.

С алгоритмической точки зрения начальную загрузку можно определить как (11):

$$Encrypt\big(pk, Decrypt_{\varepsilon}(sk, c)\big) \rightarrow c' \tag{11}$$

где c' – новый зашифрованный текст, который содержит меньше шума, чем исходный текст c.

Основным недостатком метода начальной загрузки являются вычислительные затраты, так как большинство процессов начальной загрузки сложны и медленны.

Однако, несмотря на эти ограничения, их преодоление возможно с помощью методов высокопроизводительных, распределенных и параллельных вычислений.

1.5 Схемы полностью гомоморфного шифрования

1.5.1 Схема первого поколения: схема Джентри

В 1978 году Ривест, Адлеман и Дертузос впервые предложили идею вычисления зашифрованных данных без их расшифровки [46], что привело к построению систем частично гомоморфного шифрования:

- гомоморфные относительно умножения (RSA и криптосистема Эль-Гамаля [23, 46, 47]);
- гомоморфные относительно сложения (криптосистема Пэйе [42]).

«Позже криптосистема Пэйе была модифицирована в криптосистему Боне-Го-Ниссима [13], обладающую аддитивным гомоморфизмом, т.е. допускающую произвольное количество сложение и только одно умножение, за которым снова могут следовать произвольные сложения.

Однако первая система полностью гомоморфного шифрования, то есть гомоморфная относительно как сложения, так и умножения, была предложена только спустя почти 30 лет. В 2009 году. Джентри Крейг разработал первую систему полностью гомоморфного шифрования, основанную на криптографии на идеальных решетках» [3], и поддерживающую операции сложения и умножения над зашифрованными данными [26].

Ранее при гомоморфном шифровании операции над данными всегда сопровождались возрастанием шума, появляющегося в следствие вероятностного характера метода шифрования, из-за чего условие гомоморфности соблюдалось только относительно определённого числа операций. Однако Джентри Крейг решил данную проблему с помощью перешифрования в процессе построения схемы для вычислений по зашифрованным данным, что помогло снизить шум до допустимого уровня, т.е. для «шумной» схемы Джентри почти гомоморфная схема модифицируется в гибкую путём обновления зашифрованных данных с помощью гомоморфной процедуры [26].

Также Джентри Крейг в своей работе смог доказать стойкость предложенной им системы, и несмотря на то, что разработанная им схема не совсем

подходила для решения практических задач, после неё были разработаны схемы следующих поколений, как показано на рисунке 5.

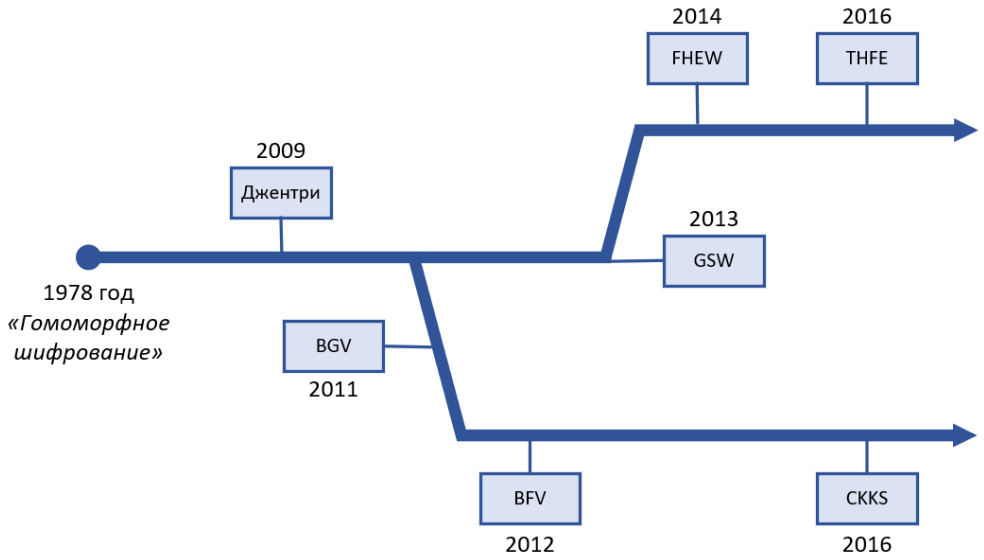

Рисунок 5 – Схема развития полностью гомоморфного шифрования

Работа Джентри Крейга стала фундаментальной основой для последующих исследований в области полностью гомоморфного шифрования и сыграла ключевую роль в развитии современных схем полностью гомоморфного шифрования, открыв путь для создания более совершенных решений, которые используются, в том числе, для защиты данных в облачном машинном обучении.

1.5.2 Схемы второго поколения: BGV и BFV

В 2011-2012 годах на основе задачи RLWE были разработаны схемы Бракерски-Гентри-Вайкунтанатан (BGV) [53] и Бракерски-Фан-Веркаутерен (BFV) [24, 54], которые также соответствуют базовой схеме, предложенной Джентри Крейгом, однако отличаются более медленным ростом шума во время вычислений. Это было достигнуто с помощью методов, позволяющих упаковывать множество значений открытого текста в один зашифрованный текст и работать со всеми этими значениями открытого текста в режиме одиночного потока команд и множественного потока данных [49].

Схемы BFV и BGV создаются в двух кольцах:

– в кольце открытого текста, которое включает в себя кодировки незашифрованных сообщений;
– кольцо зашифрованного текста, которое включает зашифрованные сообщения.

Результат шифрования обычно представляет собой два элемента пространства зашифрованного текста; первый из которых содержит замаскированные данные открытого текста, тогда как второй содержит вспомогательную информацию, которая может использоваться в процедуре расшифровки.

Одним из важнейших понятий в схемах полностью гомоморфного шифрования на основе RLWE является ошибка, также известная как шум, которая добавляется к открытому текстовому сообщению во время шифрования. Ошибка имеет решающее значение для обеспечения сложности задачи RLWE, иначе она превратится в простую вычислительную задачу, что сделает схемы на её основе легко взламываемыми. В схеме BFV величина ошибки растет по мере выполнения гомоморфных вычислений (в основном гомоморфных сложение и гомоморфное умножение), а схема BGV демонстрирует аналогичное шумовое поведение, но ошибка, возникающая вследствие гомоморфного сложения, растет медленнее, чем ошибка, обусловленная гомоморфное умножение.

Если говорить о различиях рассматриваемых схем, то схема BFV является масштабно-независимой, т.е. модуль зашифрованного текста не меняется во время гомоморфных вычислений. С другой стороны, схема BGV – это масштабно-зависимая схема, которая определяет несколько модулей зашифрованного текста по одному модулю на уровень.

Еще одно различие между схемами BFV и BGV заключается в структуре зашифрованного текста и способе шифрования открытого текстового сообщения.

На рисунке 6 показана структура зашифрованного текста в схемах BFV и BGV. В первом случае открытое текстовое сообщение помещается в направ-

лении наиболее значимого бита (англ. «most significant bit», MSB), что делается путем масштабирования сообщения. С другой стороны, схема BGV помещает сообщение в сторону наименее значащих битов (англ. «leat significant bit», LSB) зашифрованного текста.

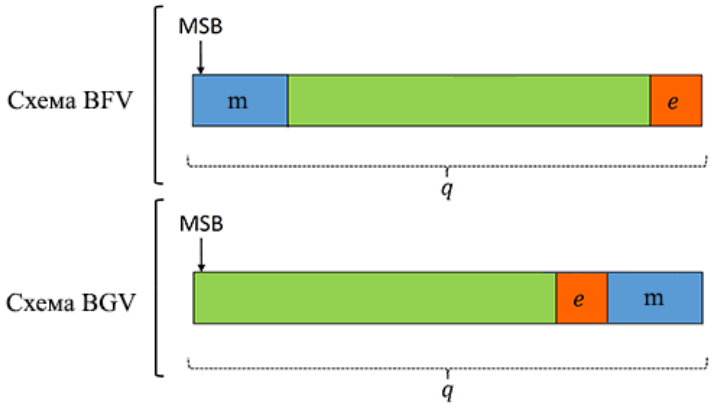

Рисунок 6 – Расположение шума e и открытого текста m в схемах BFV и BGV

«Важно, чтобы при гомоморфном вычислении исходный текст и шум никогда не перекрывались, чтобы расшифровка могла восстановить ожидаемый результат вычислений» [3].

Также схемы BGV и BFV имеют незначительные различия в производительности с точки зрения вычислительных возможностей [19, 20].

Схемы BGV и BFV по-прежнему считаются безопасными схемами и без каких-либо ограничений применяются на практике.

1.5.3 Схемы третьего поколения: GSW, FHEW и TFHE

В 2013 году Джентри Крейг, Амит Сахаи и Брент Уотерс разработали схему GSW, основанную на LWE, но без использования релинеаризации при гомоморфном умножении [27], а Джейкоб Альперин-Шериф и Крис Пайкерт предложили механизм её самонастройки [11], на основе которого были разработаны схемы FHEW (Fastest Homomorphic Encryption in the West) [22] и TFHE (Fast Fully Homomorphic Encryption over the Torus) [18].

«Схема FHEW была первой, которая показала, что, обновляя зашифрованные тексты после каждой отдельной операции, можно сократить время начальной загрузки до доли секунды. FHEW представил новый метод вычисления логических вентилей для зашифрованных данных, который значительно упрощает начальную загрузку, и реализовал вариант процедуры начальной загрузки» [11].

«Схема THFE улучшила схему FHEW с помощью кольцевого варианта процедуры самонастройки» [25].

1.5.4 Схема четвёртого поколения: CKKS

В 2016 году Чон Хи Чхон, Андрей Ким, Миран Ким и Ёнсу Сон на основе задачи RLWE разработали схему CKKS [17] для проведения арифметических операций с числами с плавающей запятой.

Основная идея заключается в том, чтобы учитывать шум или ошибку как часть сообщения открытого текста, т.е. схема в качестве входных данных принимает $(e + m)$, как показано на рисунке 7, и генерирует зашифрованный текст как приблизительное значение открытого текста, так как если шум намного меньше по величине по сравнению с сообщением, это может не оказать на него заметного влияния, что справедливо для большинства сценариев использования схемы.

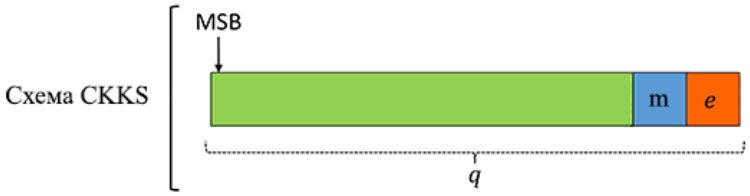

Рисунок 7 – Расположение шума e и открытого текста m в схеме CKKS

В основе схемы CKKS лежит механизм, называемый RESCALE, который используется для уменьшения размера открытого текста с учётом шума, а также механизм реализации функции масштабирования, т.к. схема CKKS, как

и схема BGV, включает в себя эффективную операцию изменения размера, которая уменьшает размер зашифрованного сообщения после умножения. Например, в схемах BGV и BFV, такое масштабирование требует начальной загрузки. Т.е. операция изменения масштаба делает схему CKKS наиболее эффективным методом вычислений полиномиальных приближений и предпочтительным подходом для защиты данных в машинном обучении [34].

1.6 Библиотеки полностью гомоморфного шифрования

Рассмотренные схемы полностью гомоморфного шифрования получили свои программные реализации, каждая из которых поддерживает определённые схемы.

Рассмотрим наиболее популярные из библиотек, выложенные в открытом доступе, и обладающие, в том числе, открытым исходным кодом, для чего сначала сведём основную информацию о них в таблицу 1.

Большинство библиотек написано на языке C++, что обусловлено его способностью обеспечивать высокую производительность, контроль над управлением памятью и поддержку параллельных вычислений.

Таблица 1 – Библиотеки полного гомоморфного шифрования

Библиотека	Язык	Разработчик	Лицензия	Поддерживаемые схемы
HELib	C++	IBM	Apache-2.0	BGV, CKKS
Microsoft SEAL	C++	Microsoft	MIT	BGV, BFV, CKKS
PALISADE	C++	New Jersey Institute of Technology	BSD 2-Clause	BGV, BFV, FHEW, TFHE, CKKS
HEAAN	C++	CryptoLab, Inc	CC BY-NC 3.0	CKKS, CKKS Bootstrapping
FHEW	C++	Leo Ducas and Daniele Micciancio	Apache-2.0	FHEW
TFHE	C++	Gama et al.	Apache-2.0	TFHE
OpenFHE	C++	Duality Technologies, Samsung Advanced Institute of Technology и др.	BSD 2-Clause	BGV, BFV, FHEW, TFHE, CKKS, CKKS Bootstrapping
Lattigo	Go	EPFL-LDS, Tune Insight	Apache-2.0	BGV, BFV, CKKS

Рассмотрим основные особенности каждой из библиотек.

HELib (Homomorphic-Encryption Library), разработанная Шаем Халеви и Виктором Шоуп [30], предоставляет возможность тонкой настройки режимов работы схем гомоморфного шифрования [3]. Библиотека поддерживает схемы BGV и CKKS и предоставляет возможность низкоуровневого манипулирования данными (сложение, умножение, сдвиг и т.д.), сложное автоматическое управление шумом, улучшенную загрузку BGV и многопоточность.

Microsoft SEAL (Simple Encrypted Arithmetic Library), разработанная в Microsoft Research, позволяет выполнять сложение и умножение зашифрованных целых или действительных чисел. Библиотека включает в себя схемы BFV и BGV, которые «позволяют выполнять модульные вычисления над зашифрованными целыми числами, и схему CKKS, которая позволяет складывать и умножать зашифрованные действительные или комплексные числа, но дает лишь приблизительные результаты» [12]. В таких приложениях, как суммирование зашифрованных действительных чисел, оценка моделей машинного обучения на основе зашифрованных данных или вычисление расстояний до зашифрованных мест, CKKS будет, безусловно, лучшим выбором. Для приложений, где необходимы точные значения, больше подходят схемы BFV и BGV [3].

PALISADE является общей библиотекой решетчатой криптографии, которая, в том числе, включает в себя схемы полностью гомоморфного шифрования BFV и BGV для целочисленных вычислений, CKKS для вычислений действительных чисел и FHEW/THFE для вычислений булевых схем. Библиотека основана на модульной архитектуре с уровнями математических операций, решётчатых операций, уровень решётчатой криптографии и уровень кодирования. Основное внимание в PALISADE уделяется удобству использования схем, например, все схемы полностью гомоморфного шифрования используют один и тот же общий API и реализуются с использованием полиморфизма во время выполнения.

HEAAN [17] реализует схему CKKS и её вариант с начальной загрузкой CKKS Bootstrapping и поддерживает вычисления комплексных и действительных чисел. Библиотека поддерживает приближенные вычисления рациональных чисел, причём ошибка зависит от подбора параметров и почти совпадает с ошибкой вычислений чисел с плавающей запятой.

FHEW является реализацией одноименной схемы и в своей работе использует библиотеку FFTW (Fastest Fourier Transform in the West). Библиотека предоставляет симметричную схему шифрования однобитовых открытых текстов и поддерживает гомоморфную оценку произвольных логических схем.

THFE также является реализацией одноименной схемы и появился как модернизация схемы и библиотеки FHEW. Библиотека может гомоморфно вычислять список соединений логических вентилей со скоростью около 50 вентилей в секунду на ядро, без расшифровки входных данных. В отличие от других библиотек, TFHE не имеет ограничений ни на количество вентилей, ни на их состав. Это делает библиотеку пригодной для использования как с схемы, созданные вручную, или с помощью инструментов автоматического создания схем.

OpenFHE является преемником PALISADE и включает в себя часть особенностей библиотек HElib, HEAAN и FHEW и эффективные реализации всех распространенных схем полностью гомоморфного шифрования. Библиотека также поддерживает переключение между CKKS и FHEW/TFHE для оценки негладких функций, например сравнения, с использованием функциональной начальной загрузки FHEW/TFHE, а также поддерживает несколько схем начальной загрузки.

Lattigo – модуль, поддерживающий схемы BGV, BFV и CKKS, и предназначенный для поддержки полностью гомоморфного шифрования в распределенных системах и микросервисной архитектуре, для которых Go является языком разработки.

Каждая из рассмотренных библиотека имеет свои особенности в плане функциональности и производительности. Например, некоторые из них ориентированы на высокую скорость вычислений, другие – на поддержку сложных операций над зашифрованными данными. Выбор конкретной библиотеки зависит от того, какие операции необходимо выполнять и насколько критична скорость выполнения этих операций.

При этом все рассматриваемые библиотеки требуют от пользователя высокого уровня понимания многих конкретных концепций гомоморфного шифрования, так как разница между эффективным и неэффективным использованием данной библиотеки может составлять несколько порядков, и неопытным пользователям может быть сложно понять, как повысить производительность своих вычислений.

Также рассмотренные библиотеки не предназначены для непосредственного использования в машинном обучении, так как их изначальная цель состояла в исследовании технологии полностью гомоморфного шифрования или обеспечении безопасности данных в условиях распределённой обработки.

Тем не менее, в 2020 году библиотека Microsoft SEAL была адаптирована для использования в машинном обучении в виде библиотеки TenSEAL, которая обеспечивает простоту использования языка Python, но сохраняет эффективность за счет реализации большинства операций с использованием C++, предоставляя полный API для Microsoft SEAL.

TenSEAL легко интегрируется с популярными фреймворками машинного обучения, такими как TensorFlow и PyTorch, позволяет производить поэлементное сложение, вычитание и умножение зашифрованных векторов, а шифрование и дешифрование данных осуществляет по схеме BFV для целых чисел или CKKS для вещественных чисел. Следует отметить, что в машинном обучении чаще всего используется именно схема CKKS, так как использование вещественных чисел позволяет добиться большей точности.

Также данная библиотека уже нашла применение в реальных проектах и исследованиях [39, 45, 50].

Таким образом, использование TenSEAL позволяет разработчикам эффективно обеспечить безопасность данных, обрабатываемых в облаке, при этом минимизируя сложность внедрения и эксплуатации.

1.7 Полностью гомоморфное шифрование в облачном машинном обучении

1.7.1 Примеры использования полностью гомоморфного шифрования в облачном машинном обучении

«Большинство работ [39, 40, 45, 50, 51], посвященных использованию полностью гомоморфного шифрования для защиты данных в машинном обучении, описывают только шифрование на фазе вывода работы нейронной сети, и не касаются этапа обучения. Например, работа [39] 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 [45] для набора данных MNIST вывод занял уже 30 мс.

Предположительно, небольшое внимание к обучению на зашифрованных данных обусловлено тем, что оно занимает слишком много времени, однако уже в 2019 году в работе [33] с помощью библиотеки HELib была обучена нейронная сеть на основе стохастического градиентного спуска (SGD), что продемонстрировало эффективность обучения на зашифрованных данных, а также авторы предложили несколько методов для улучшения обучения, например, анализ сети» [5], выбор подходящего представления данных и оптимизация упаковки данных в зашифрованных текстах.

В работе [37] была реализована логистическая регрессия и аппроксимированная функция логистической регрессии с помощью схемы CKKS и с использованием Microsoft SEAL. Процент правильных прогнозов совпал для обучения на открытых и зашифрованных данных, однако после шифрования размер входного набора данных увеличился в 272.5. раз, а время обработки данных увеличилось примерно в 4865 раз.

Также не все работы используют существующие схемы полностью гомоморфного шифрования. В работе [35] был представлен вариант бесшумной схемы гомоморфного шифрования на основе матрицы (MORE), которая требует достаточно небольших вычислительных затрат и, что важно, позволяет выполнять операции непосредственно с числами с плавающей запятой, что собой критическое свойство для нейронных сетей. Однако схема шифрования MORE обеспечивает меньшую безопасность по сравнению со стандартными схемами, так как основывается на линейных функциях и существует возможность определения секретного ключа, имея доступ к достаточно большому количеству пар зашифрованных и незашифрованных значений.

1.7.2 Гомоморфные вычисления в моделях машинного обучения

При использовании полностью гомоморфного шифрования над «зашифрованными данными возможны только операции сложения и умножения, то с помощью них должны быть представлены все вычисляемые функции» [5]. В то время как в машинном обучении требуются неполиномиальные функции, в частности, логистической регрессии, применяемой для решения задачи прогнозирования задолженности, необходима сигмоидная функция (12):

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{12}$$

Как видно из формулы (1.12), сигмоида не может быть эффективно представлена с помощью только сложения и умножения, и поэтому не может быть вычислена гомоморфно, поэтому разработка эффективного способа поддержки неполиномиальных функций является активной темой исследований в последние годы.

«Для поддержки неполиномиальных функций, в частности, логистической сигмоиды» [41], в гомоморфных вычислениях выделяются следующие подходы:

– использование полиномиальной замены;

– «предварительное вычисление функции для дискретных значений и сохранение этих значений в справочных таблицах» [41];
– использование полиномов низкой степени для аппроксимации неполиномиальных функций.

Подход, «заключающийся в использовании полиномиальной замены, был самым первым решением проблемы поддержки неполиномиальных функций, и был представлен в 2016 году в работе [15], посвящённой CryptoNets, в которой авторы предложили замену сигмоиды квадратичным полиномом» (13) [5]:

$$f(x) = x^2 \tag{13}$$

Но квадратичный полином растет экспоненциально и вызывает нестабильность во время обучения нейросети [16]. Как следствие, CryptoNets не может поддерживать более одного уровня активации, что оказывает значительное влияние на точность.

Другой подход был представлен в 2018 году в работе [21], заключающийся в предварительном вычислении дискретных выборок неполиномиальной функции активации и сохранении этих «дискретных значений в справочной таблице. Во время гомоморфных вычислений программа выполняет поиск по таблице, однако низкие частоты дискретизации снизят точность вычислений с плавающей точкой и окажут прямое влияние на производительность нейронных сетей» [41]. И наоборот, увеличение частоты выборки может привести к созданию слишком больших справочных таблиц.

«Более эффективный и часто используемый подход заключается в аппроксимации неполиномиальных функций полиномами низкой степени» [5]. Использование низкой степени обусловлено тем, что схемы гомоморфного шифрования ограничивают количество операций умножения.

Рассмотрим наиболее используемые методы полиномиальной аппроксимации.

Численный метод

Данный метод заключается в том, что «коэффициенты полинома $p(x)$ степени N подбираются при помощи логистической регрессии таким образом, чтобы минимизировать среднюю квадратичную ошибку (mean squared error – MSE) в узлах интерполяции» [31].

В работе [28] был исследован данный метод, для чего авторы сгенерировали набор вычисленных значений точек функции ReLU, который передали функции аппроксимации вместе со степенью полинома. Эксперименты проводились с полиномами степени от 3 до 13, в результате чего было выяснено, что чем ниже степень полинома, тем ниже точность, то есть для достижения достаточной точности требуется увеличивать степень полинома, что не является эффективным для гомоморфных вычислений. Кроме этого, использование численного метода продемонстрировало низкую производительность.

Ряд Тейлора

«Аппроксимация функции строится численным методом, и заключается в построении полинома, приближающего исходную функцию на заданном отрезке, методом Крога [31], и разложении функции $f(x)$ в ряд Тейлора в окрестностях заданной точки c» (14) [5]:

$$f(x) = f(c) + \sum_{k=1}^{\infty} \frac{f^{(k)} \cdot (c)}{k!} \cdot (x - c)^k \tag{14}$$

Например, функция ReLU была аппроксимирована с помощью разложения в ряд Тейлора в работе [29], и в сочетании с другими методами, такими как пакетная нормализация [32], устанавливающей границу входных данных для каждого слоя активации, авторы смогли достичь точность 99.3 %

В работе [28] также исследовалась аппроксимация функции ReLU с помощью разложения в ряд Тейлора, для чего были использованы полиномы разных степеней, однако метод столкнулся с двумя основными проблемами, де-

лающими его неэффективным. Во-первых, проблемой является высокая степень полиномиальной аппроксимации, хотя она ниже, чем у численного метода, но всё еще выше для использования в схемах полностью гомоморфного шифрования. Во-вторых, в данном методе существует проблема интервала аппроксимации, так как разложение в ряд Тейлора заключается в аппроксимации функции в окрестностях какой-то точки, и для точек, не из рассматриваемой окрестности, ошибка аппроксимации значительно выше, чем для точки, принадлежащей ей.

Полиномы Чебышёва

В отличии от разложения в ряд Тейлора использование полиномов Чебышёва позволяет аппроксимировать функцию на интервале, а не в окрестности точки.

Аппроксимация функции $f(x)$ на отрезке $[a, b]$ строится в форме полинома следующего вида (15) [31]:

$$f(x) = \sum_{i=0}^{N} a_i \cdot T_i \cdot \left(2 \cdot \frac{x-a}{b-a} - 1\right) \tag{15}$$

Сами полиномы Чебышёва определяются как (16):

$$T_i(x) = \cos \cdot \left(i \cdot arccos(x)\right), x \in [-1, 1] \tag{16}$$

где a_i – соответствующие коэффициенты.

В работе [28] было рассмотрено использование как стандартных полиномов Чебышёва, которые ортогональны на отрезке $[-1, 1]$ с весом (17):

$$h = \frac{1}{\sqrt{1-x^2}} \tag{17}$$

Так и их модифицированная версия, в которой было предложено рассчитывать вес следующим образом (18):

$$h = e^{-\frac{1}{1e-5+x^2}} \tag{18}$$

В результате чего было выяснено, что использование стандартных полиномов Чебышёва позволила достичь точности только в 69.98 %, в то время как их модицифированная версия достигла точность 88.53 %.

Большая точность связана с тем, что вес, рассчитанный специально для аппроксимируемой функции, приводит к меньшим колебаниям в результирующем приближении и, следовательно, к большему сходству производных с сигмоидной функцией.

Производная аппроксимируемой функции

Данный метод предполагает использование аппроксимации полиномами низкой степени одним из методов, описанных ранее, но вместо самой функции $f(x)$ аппроксимировать её производную $f'(x)$. Тогда аппроксимацией функции $f(x)$ на отрезке $[a, b]$ является полином $p(x)$ (19) [28]:

$$p(x) = \int p_1(x)dx \tag{19}$$

где $p_1(x)$ – полиномиальное приближение для производной $f'(x)$ на отрезке $[a, b]$.

Сигмоидная функция является непрерывной и бесконечно дифференцируемой, и в работе [28] было доказано, что это позволяет аппроксимировать ее более точно, чем, например, функцию ReLU, которая не дифференцируема в точке 0.

Минимаксная аппроксимация

При использовании данного метода максимальное отклонение аппроксимирующей функции от исходной на заданном интервале минимизируется.

Такая задача обычно решается с помощью численных методов оптимизации, например, полиномов Чебышёва.

В работе [52] была рассмотрена минимаксная аппроксимация сигмоидной функции с использованием алгоритма Ремезы, а также сочетание начальной загрузки с операцией масштабирования для вычислений чисел с фиксированной точкой, в результате чего была достигнута точность 97.8 % при затратах в 0,4-3,2 часа на одну итерацию градиентного спуска.

Как видно из существующих работ, подход полиномиальной аппроксимации более надежен, чем подход поиска по таблице, и дает более высокую точность, чем простое использование функции замены низкой степени. Но существующие работы также показывают, что определение наиболее эффективного полинома является наиболее сложной и важной задачей.

Поэтому, рассмотрев пять подходов к аппроксимации сигмоидной функции, сведём примеры полиномов, аппроксимирующих сигмоидную функцию, в таблицу 2, используя полиномы для первых четырёх подходов из работы [31], а для минимаксной аппроксимации – из работы [52].

При этом ограничимся только полиномами третьей степени, так как полиномы второй степени не обеспечивают достаточную точность [37], а использование полиномов высоких степеней ограничивается возможным количеством умножений в гомоморфном шифровании [16].

Для оценки качества аппроксимации используем среднюю квадратичную ошибку. В работе [31] значения MSE уже рассчитаны, в то время как для минимаксной аппроксимации её требуется рассчитать по формуле (20):

$$MSE(f,p) = \frac{1}{m} \cdot \sum_{i=1}^{m} (f(x_i) - p(x_i))^2 \qquad (20)$$

Таблица 2 – Полиномиальные аппроксимации для сигмоидной функции

Метод	Полином	MSE
Численный метод [10] $x_i \sim U(-200, 200)$	$\sigma(x) = -1.378 \cdot 10^{-7} \cdot x^3 - 3.097 \cdot 10^{-7} \cdot x^2 + 0.007042 \cdot x + 0.5127$	0.0328
Ряд Тейлора [10]	$\sigma(x) = -2.437 \cdot 10^{-8} \cdot x^3 - 2.054 \cdot 10^{-19} \cdot x^2 + 0.003363 \cdot x + 0.5$	0.0634
Полиномы Чебышёва [10]	$\sigma(x) = -1.061 \cdot 10^{-7} \cdot x^3 - 2.141 \cdot 10^{-20} \cdot x^2 + 0.006365 \cdot x + 0.5$	0.0340
Аппроксимация производной [10]	$\sigma(x) = -5.305 \cdot 10^{-8} \cdot x^3 + 0.004774 \cdot x + 0.5305$	0.0445
Минимаксная аппроксимация [47] $x \in [-5; 5]$	$\sigma(x) = -0.004 \cdot x^3 + 0.197 \cdot x + 0.5$	0.0114

Из таблицы 2 видно, что минимаксная аппроксимация сигмоидной функции показывает наименьшую среднюю квадратичную ошибку, но, чтобы убедиться в этом, построим графики аппроксимирующей и сигмоидной функций, как показано на рисунке 8.

Таким образом, в результате проведённого анализа, для решения задачи прогнозирования задолженности предлагается использовать аппроксимирующую функцию, «полученную с помощью минимаксной аппроксимации, так как она наиболее точно аппроксимирует сигмоидную функцию» [5].

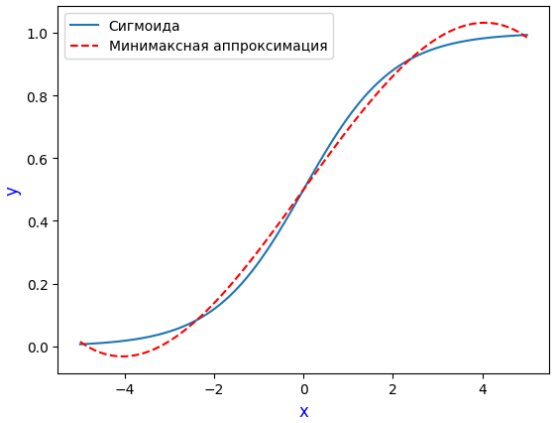

Рисунок 8 – Графики аппроксимирующей (минимаксная аппроксимация) и сигмоидной функций

«Также следует отметить, что во всех рассмотренных работах процесс шифрования данных не выделяется на отдельный этап, то есть в рамках клиент-серверной архитектуры загрузка и шифрование данных происходит на сервере, а не на клиенте, что подвергает данные определённым угрозам информационной безопасности, что предлагается решить в рамках данной работы» [5].

Выводы по первой главе

В первой главе рассмотрены способы обеспечения безопасности данных в облачном машинном обучении, состоящие либо в полном отказе от использования облачного сервера и закупке собственного оборудования, либо в программном обеспечении безопасности данных. Было обосновано использование программного способа, заключающегося в использовании полностью гомоморфного шифрования. Рассмотрено понятие полностью гомоморфного шифрования и вычислительные задачи, лежащие в его основе, в частности, задачи обучения с ошибками (в кольце). Также проанализированы параметры безопасности задач и методы их решения. Рассмотрены способы борьбы с накоплением ошибки в схемах полностью гомоморфного шифрования.

Рассмотрены существующие схемы полностью гомоморфного шифрования, и схема CKKS определена как наиболее эффективная для использования в машинном обучении.Произведён анализ наиболее популярных библиотек полностью гомоморфного шифрования и выбрана библиотека TenSEAL, так как она разработана специально для использования в машинном обучении.

Рассмотрено использование полностью гомоморфного шифрования для защиты данных в облачном машинном обучении. Было установлено, что из-за того, что гомоморфные вычисления поддерживают только операции сложения и умножения, требуется аппроксимация неполиномиальных функций, используемых в моделях машинного обучения. Были рассмотрены различные способы аппроксимации сигмоидной функции, и был выбран минимаксный способ аппроксимации, как наиболее точный.

Глава 2 Анализ бизнес-процессов компании «КВАРТПЛАТА 24»

2.1 Характеристика деятельности компании «Квартплата 24» и сервиса по работе с должниками

Компания ООО «Квартплата 24» – это ИТ-компания, специализирующаяся на решении задач по расчету и учету платежей за жилищно-коммунальные услуги, а также приеме, распределении и взыскании долгов с соблюдением всех законодательных норм. Клиентами компании являются товарищества собственников жилья, управляющие и ресурсоснабжающие организации и информационно-расчетные центры по всей России.

Отраслевая цифровая платформа «Квартплата 24» является агрегатором биллинговых и платежных систем, обеспечивая сетевое взаимодействие и взаимную интеграцию участников рынка ЖКХ с применением технологии распределенных реестров (DLT).

Таким образом, «Квартплата 24» предоставляет различные услуг с помощью более 10 тесно интегрированных между собой облачных сервисов, как показано на рисунке 9.

Компания предоставляет большое количество услуг, в том числе, консалтинг, экспертную поддержку и процессинг платежей. Но в данной работе более подробно рассмотрим сервис по работе с должниками, являющийся инструментом долгосрочного поддержания высокого уровня платежной дисциплины за счет установления эффективного бизнес-процесса востребования задолженности. Он позволяет дополнительно повысить собираемость платежей на 3-8 %, а также повысить эффективность работы сотрудников, позволяя каждому формировать до 800 заявлений в суд в месяц. Кроме того, вся информация о должниках и работе с ними хранится в одном месте.

Рисунок 9 – Экосистема облачных сервисов «Квартплата 24»

Однако важно не только работать с существующими должниками, но и прогнозировать потенциальных. Определения лиц с высоким риском возникновения задолженности по услугам ЖКХ позволит своевременно уведомить об этом соответствующие организации. Вследствие чего они могут заранее

принять меры для решения проблемы, что тем самым снизит количество просроченных платежей и повысит уровень удовлетворенности клиентов от использования услуг компании «Квартплата 24».

Использование машинного обучения для анализа данных о клиентах открывает новые возможности для выявления лиц с повышенным риском возникновения задолженности по услугам ЖКХ. Такая аналитика позволяет оперативно идентифицировать группы потребителей, которым могут потребоваться дополнительные напоминания или индивидуальные предложения по оплате счетов.

Тем не менее, данный процесс ещё находится в процессе разработки, и так как компания не обладает достаточной собственной вычислительной мощностью и не готова инвестировать в дорогостоящее оборудование, то ей необходимо арендовать облачный сервер. Однако этот подход несет риски утечки конфиденциальных данных клиентов, поскольку облачная инфраструктура находится под контролем стороннего провайдера.

2.2 Анализ существующего процесса облачного машинного обучения с использованием открытых данных

Машинное обучение является сложным и многоэтапным процессом, в течении которого модель обучают, разрабатывают и обслуживают. Так как обучение происходит на облачном сервере, то на него передаётся большое количество конфиденциальных данных. Для решения задачи прогнозирования задолженности такими данными являются история платежей, сведения о месте жительства, о предыдущих задолженностях, данные о потреблении коммунальных услуг и т. д.

Представим BPMN-диаграмму существующего процесса разработки модели прогнозирования задолженности по услугам ЖКХ на рисунке 10.

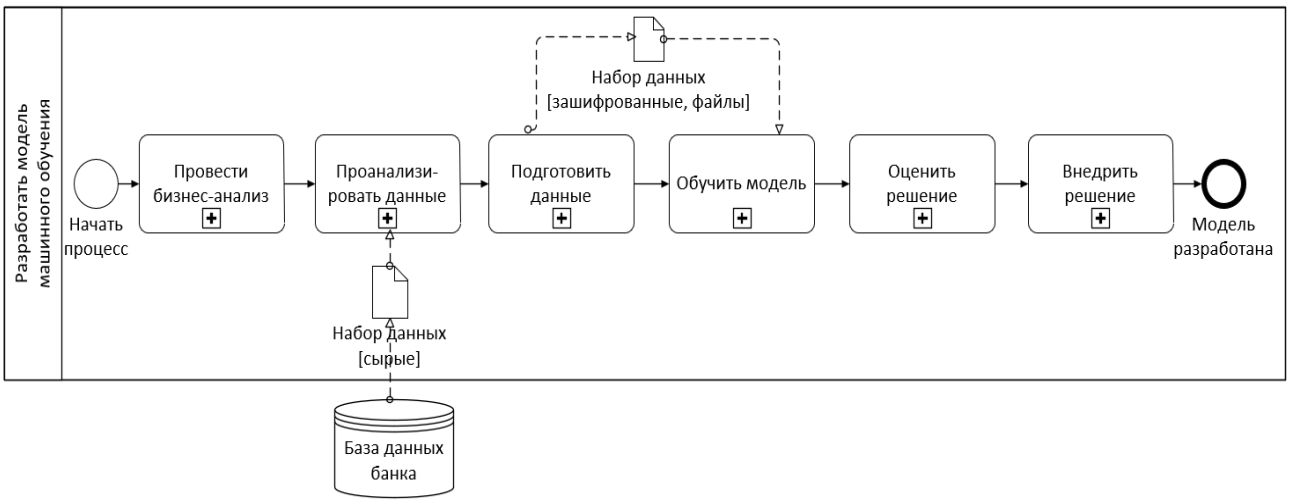

Рисунок 10 – BPMN-диаграмма модели разработки модели машинного обучения КАК ЕСТЬ

Процесс разработки модели начинается с проведения бизнес-анализа, целью которого является определение бизнес-требований и формулирование проблемы бизнеса, которые будет решать модель.

Данный свернутый подпроцесс включает в себя такие задачи как сбор требований, определение бизнес-переменных, которые будет прогнозировать модель, оценку ресурсов, которые потребуются в течении проекта, описание вероятных рисков и действий по их уменьшению, постановку задачи в терминах машинного обучения, а также формирование команды проекта и расширенного поэтапного плана проекта.

После бизнес-анализа следует этап анализа данных, который включает в себя такие задачи как понимание слабых и сильных сторон в имеющихся данных, определение их достаточности, предложение идей их использования, полный анализ и описание текущих данных, а также их оценка.

Результаты проведённого анализа данных становится основой для подготовки данных. Контекстная диаграмма этого процесса представлена на рисунке 11.

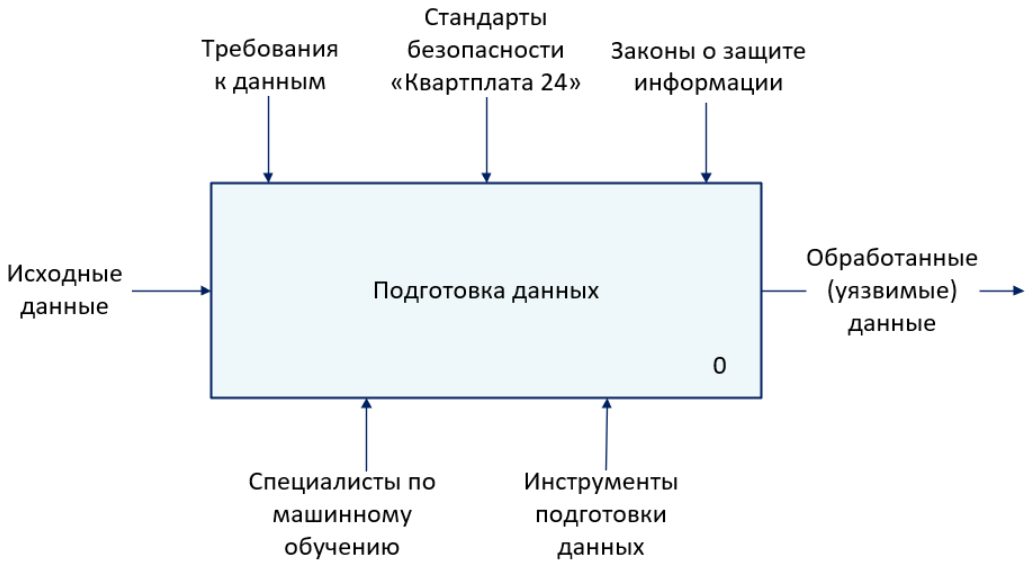

Рисунок 11 – Контекстная диаграмма процесса подготовки данных КАК ЕСТЬ

На этом этапе подготовки данных требования к данным преобразуются в данные. Рассмотрим процесс более подробно, представив декомпозицию контекстной диаграммы процесса подготовки данных на рисунке 12.

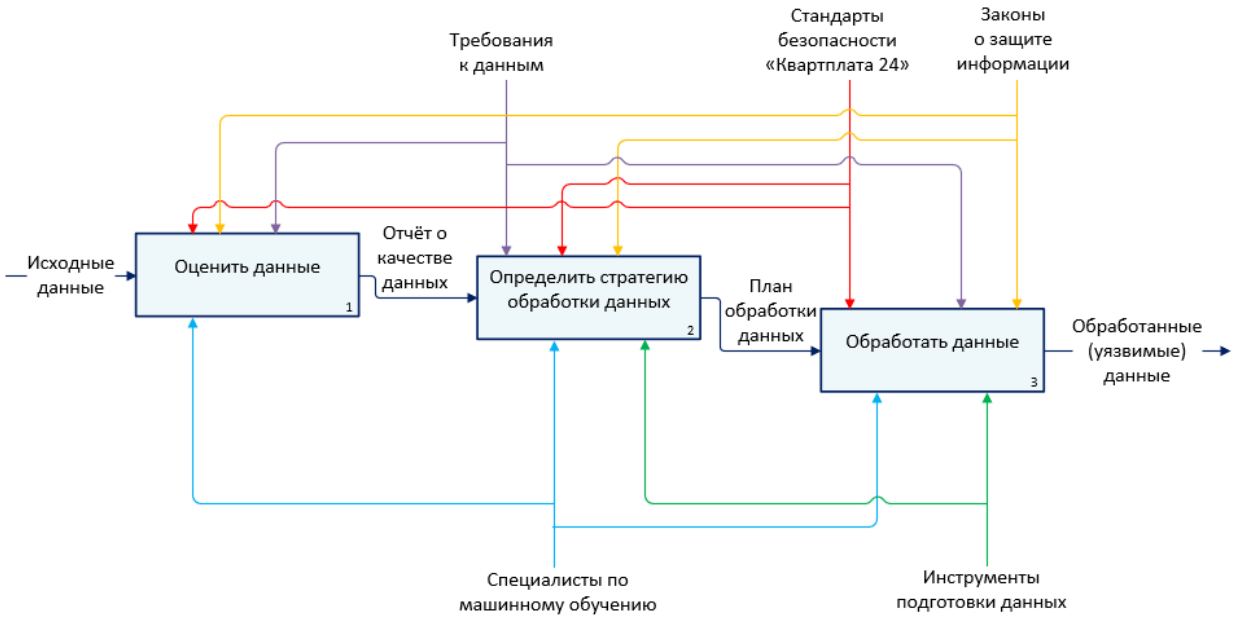

Рисунок 12 – Декомпозиция контекстной диаграммы процесса подготовки данных КАК ЕСТЬ

На рисунке 12 видно, что подготовка данных включает в себя три основных подпроцесса, заключающихся в сборе, оценке и обработке данных, что также включает в себя задачи очистки данных от пропусков и аномалий, нормализацию данных, кодирование категориальных данных, и разбиение данных на тренировочные и тестовые наборы.

Подготовленные данные передаются на сервер в открытом виде для дальнейшего обучения. Данный процесс представлен на рисунке 13.

Процесс обучения модели кроме задачи загрузки подготовленных данных на сервер, включает в себя подпроцессы выбора алгоритма, разработки и обучения модели, а также оценку результата.

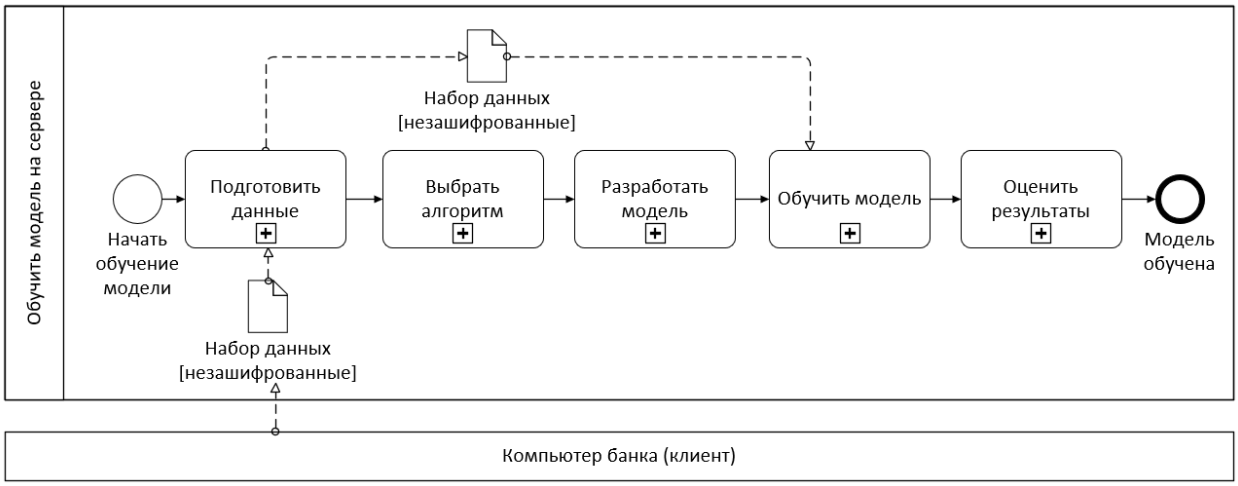

Рисунок 13 – BPMN-диаграмма процесса обучения модели КАК ЕСТЬ

Обучение модели является сложным процессом и может включать в себя множество подпроцессов и задач. Например, выбор алгоритма начинается с понимания того, какие модели будут использоваться, что зависит от решаемой задачи, используемых признаков и требований по сложности. Но в целом наиболее важными задачами в данном подпроцессе являются разработка самой модели, в том числе, настройка её гиперпараметров, непосредственное обучение модели на тренировочных данных и её тестирование на тестовом

наборе, а также итеративное улучшение модели и возможный выбор других алгоритмов.

Обученную модель необходимо оценить с точки зрения её способности решить поставленную проблему прогнозирования задолженности по услугам ЖКХ, и если результат оценки удовлетворителен, то её необходимо внедрить. Это означает сделать её доступной для других бизнес-систем. Внедряя модель, другие системы могут отправлять ей данные и получать от модели прогнозы, которые, в свою очередь, используются в системах компании. Данный подпроцесс включает в себя такие задачи как подготовку инфраструктуры для внедрения модели, разработку программного обеспечения для взаимодействия с моделью, тестовую эксплуатацию модели и сбор обратной связи от пользователей, а также мониторинг и поддержку модели, включая обновление данных для повторного обучения при необходимости.

Таким образом, существующий процесс машинного обучения можно более наглядно представить на рисунке 14.

Рисунок 14 – Опасный процесс облачного машинного обучения

Из-за того, что данные передаются на сервер в открытом виде, существует риск информационной безопасности. То есть в процессе разработки модели машинного существует такое узкое место как недостаточная защищённость данных, обрабатываемых в облаке. Его улучшение возможно с помощью программного шифрования данных с использованием полностью гомоморфного шифрования, которое позволяет не только шифровать данные, но и производить вычисления с зашифрованными данными.

2.3 Требования к системе обеспечения безопасности данных в облачном машинном обучении

Для реализации системы, обеспечивающей безопасность данных в облачном машинном обучении, необходимо учесть требования различных заинтересованных сторон, к которым относятся:

- руководство компании;
- специалисты отдела информационной безопасности;
- специалисты отдела машинного обучения;
- клиенты.

Для фиксирования требований заинтересованных сторон используем интервью. Отметим, что во время интервью аналитик формально или неформально направляет вопросы заинтересованной стороне, чтобы получить ответы, которые будут использованы для создания формальных требований.

Представим результаты определения требований заинтересованных сторон в виде таблицы 3.

Во время определения требований важно задавать конкретные вопросы, например, специалистам машинного обучения обязательно следует задать вопрос: «Какие алгоритмы машинного обучения наиболее часто используются в вашем отделе, и возможно ли их представить с помощью операций сложения и умножения?».

Таблица 3 – Содержание для всех требований

Заинтересованная сторона / Требование	Руководство компании	Специалисты отдела информационной безопасности	Специалисты отдела машинного обучения	Клиенты
Обеспечение безопасности данных	✓	✓	✓	✓
Соблюдение законодательства по обеспечению информационной безопасности	✓	✗	✗	✗
Затраты на внедрение и использование новой технологии оправданы	✓	✗	✗	✗
Внедрение методов оценивания обеспечения безопасности данных	✗	✓	✗	✗
Надёжная система управления ключами шифрования	✗	✓	✓	✗
Логирование всех операций с данными	✗	✓	✗	✗
Шифрование данных	✗	✗	✓	✗
Дешифрование данных	✗	✗	✓	✗
Вычисления с зашифрованными данными	✗	✗	✓	✗
Возможность оптимизации используемых алгоритмов машинного обучения	✗	✗	✓	✗
Обработка больших наборов данных с приемлемой производительностью	✗	✗	✓	✗
Обучение работе с библиотекой	✗	✗	✓	✗

Для лучшего понимания нужд и ожиданий специалистов машинного обучения, как основных пользователей библиотеки полностью гомоморфного шифрования, разработаем карту эмпатии, показанную на рисунке 15.

Связь карты эмпатии с процессом выявления требований в проекте заключается в том, что карта эмпатии предоставляет ценную информацию о чувствах, потребностях, целях и болевых точках заинтересованных сторон. Эта информация может быть использована в качестве отправной точки для формулирования требований, которые соответствуют ожиданиям и потребностям клиентов проекта.

Рисунок 15 – Карта эмпатии

На основе данных карты эмпатии формулируются требования, которые отражают ожидания, потребности и цели клиентов. Эти требования становятся основой для создания документа с требованиями.

2.4 Рекомендации по реализации требований, предъявленных к обеспечению безопасности данных в облачном машинном обучении

Для реализации требований, предъявляемых к системе обеспечения безопасности данных в облачном машинном обучении необходима комплексная стратегия, включающая такие направления как обеспечение безопасности данных, соблюдение законодательства в области информационной безопасности,

внедрение методов оценки безопасности, управление ключами шифрования, логирование операций с данными, а также поддержку необходимых вычислительных процессов и обработку больших объемов данных с приемлемой производительностью.

Определим методы и пути решения основных требований.

Для достижения высокого уровня защищённости данных должно использоваться полностью гомоморфное шифрование, которая позволяет выполнять операции над зашифрованными данными без их предварительного расшифрования. Это может быть достигнуто за счет использования библиотеки TenSEAL, обеспечивающей реализацию клиент-серверного сценария работы, как показано на рисунке 16. Данные шифруются на стороне клиента перед передачей на сервер, где производится обучение модели на основе этих зашифрованных данных. Такой подход гарантирует защиту конфиденциальной информации от несанкционированного доступа даже в случае компрометации сервера.

Рисунок 16 – Безопасный процесс облачного машинного обучения

Система управления ключами должна быть основана на ротации ключей и использовании защищённых хранилищ. Ключи должны регулярно обновляться, чтобы снизить риск их компрометации. Применение политик управления доступом и строгих правил хранения ключей обеспечит высокую степень защиты криптографических материалов.

Библиотека TenSEAL предоставляет все необходимые инструменты для выполнения операций шифрования, дешифрования и вычислений с зашифрованными данными. Эти функции интегрированы в систему таким образом, чтобы обеспечить максимальную производительность и удобство использования.

На рисунке 17 представлена диаграмма плавательных дорожек для реализации требования «Шифрование данных», требующего от специалиста по машинному обучению взаимодействие как с системой управления ключами, так и с библиотекой TenSEAL.

Кроме этого, специалист по машинном обучению должен адаптировать алгоритмы машинного обучения под особенности полностью гомоморфного шифрования. Например, для логистической регрессии сигмоидная функция может быть аппроксимирована полиномами низких степеней, что позволяет сохранить точность моделей при минимизации вычислительной сложности.

Также должны быть разработаны и внедрены процедуры и стандарты обеспечения безопасности, проводиться регулярный мониторинг и оценка эффективности применяемых мер безопасности, включая тестирование на проникновение и анализ уязвимостей. Все операции с данными должны фиксироваться в журнале событий, а также должна быть разработана система автоматического реагирования на инциденты безопасности, позволяющая оперативно обнаруживать и устранять нарушения.

При этом стоимость внедрения, поддержки и эксплуатации технологии полностью гомоморфного шифрования будет компенсирована значительными преимуществами в плане конфиденциальности и безопасности данных. Кроме

того, затраты снижаются благодаря использованию облачных решений, которые позволяют гибко управлять ресурсами и исключить вложения в дорогостоящее оборудование.

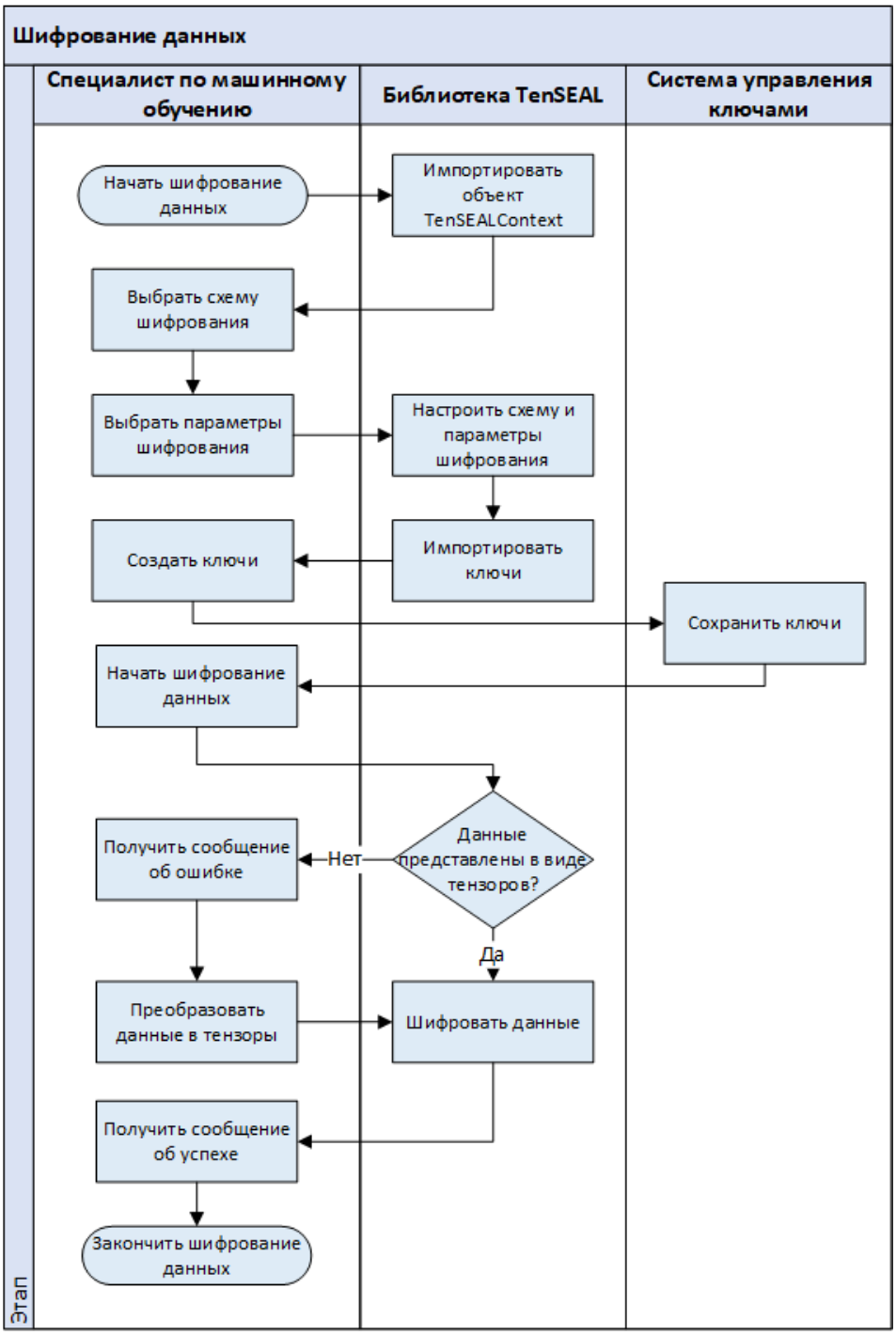

Рисунок 17 – Диаграмма плавательных дорожек для реализации требования «Шифрование данных»

Полностью гомоморфное шифрование так же обеспечивает соблюдение требований российского законодательства в области информационной безопасности. В частности, компания должна соблюдать положения Федерального закона №152-ФЗ «О персональных данных».

Таким образом, предложенные рекомендации позволят удовлетворить большинство предъявленных требований к системе обеспечения безопасности данных в обучении модели прогнозирования задолженности по услугам ЖКХ, обеспечивая высокий уровень защиты информации, соответствие законодательству и эффективную работу с данными.

Схема обеспечения выполнения всех требований, использующей рассмотренные методы и пути решения, представлена на рисунке 18.

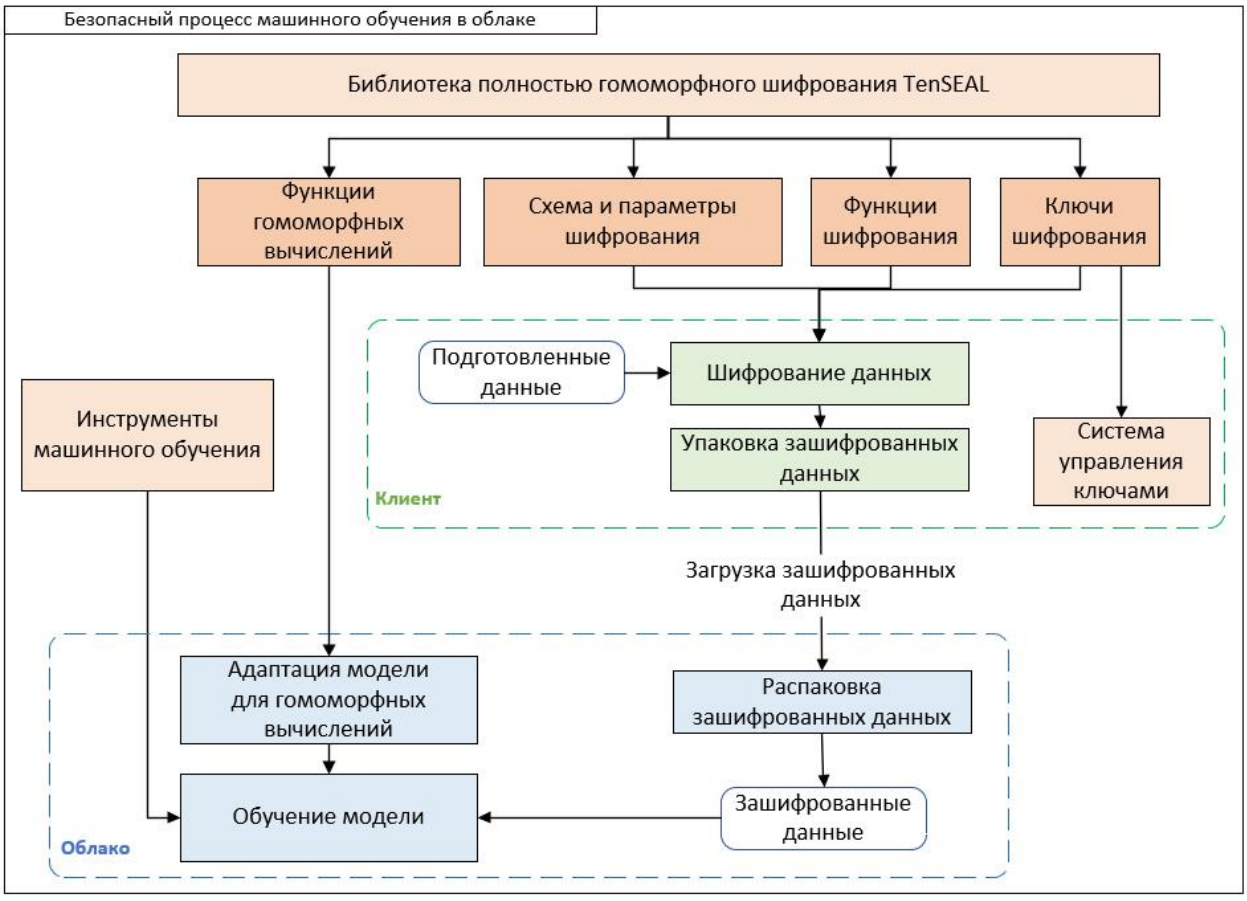

Рисунок 18 – Схема безопасного процесса облачного машинного обучения

Таким образом, предложенная схема обеспечивает полный цикл защиты данных, начиная от их шифрования и заканчивая выполнением вычислений на зашифрованных данных. Также она позволит компании «Квартплата 24» не только соответствовать строгим требованиям безопасности, но и укрепить свою позицию на рынке, предлагая клиентам и партнерам надежные и безопасные сервисы.

Отметим, что такая система, удовлетворяющая всем требованиям безопасности, сопровождается определёнными ограничениями, такими как производительность. Это связано с тем, что полностью гомоморфное шифрование может значительно увеличивать время вычислений по сравнению с обработкой незашифрованных данных, что требует оптимизации алгоритмов машинного обучения.

Поэтому требование к производительности обрабатываемых данных не может выполняться в полной мере.

2.5. Анализ усовершенствованного процесса облачного машинного обучения с обеспечением безопасности данных, передаваемых на сервер

Усовершенствованный процесс облачного машинного обучения, использующий зашифрованные данные, направлен на повышение уровня безопасности и соответствия требованиям законодательства в области защиты информации. Он предусматривает предварительное шифрование данных на стороне клиента и дальнейшую передачу зашифрованной информации на сервер для проведения обучения модели.

С учётом рекомендаций к реализации требований к обеспечению безопасности данных покажем, как изменится процесс разработки модели с использованием технологии полностью гомоморфного шифрования на рисунке 19.

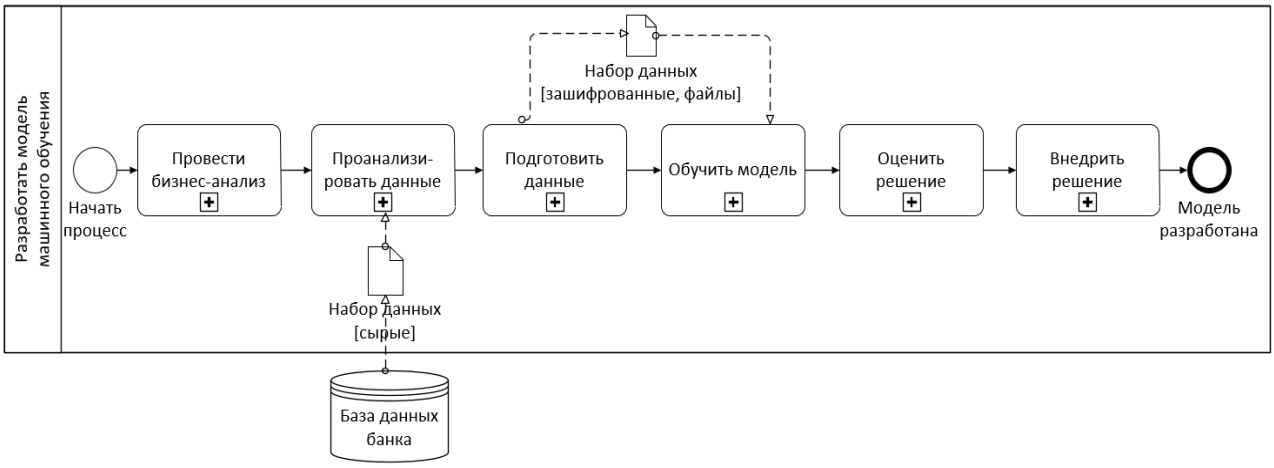

Рисунок 19 – BPMN-диаграмма модели разработки модели машинного обучения КАК ДОЛЖНО БЫТЬ

На этапе подготовки данных, контекстная диаграмма которого представлена на рисунке 20, добавляется использование такого инструмента как библиотека TenSEAL.

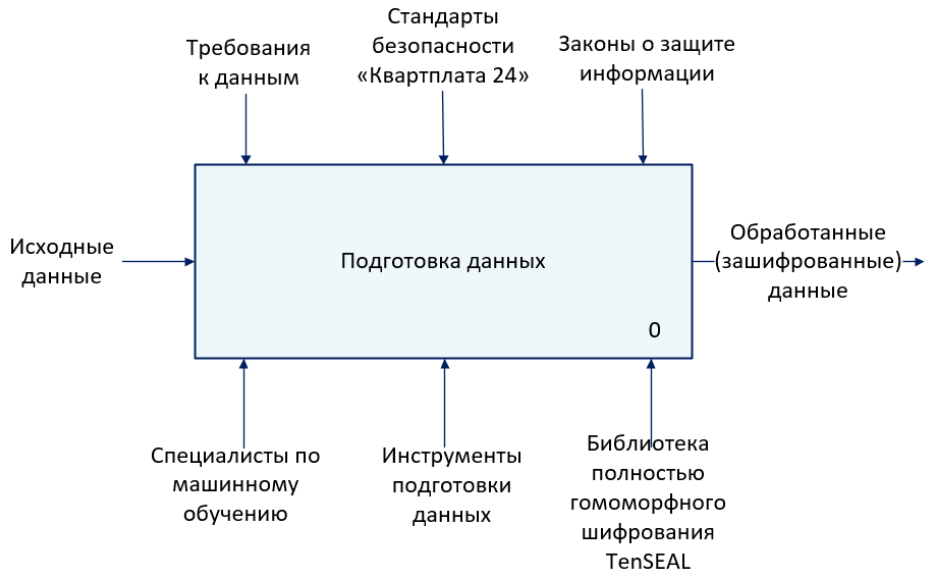

Рисунок 20 – Контекстная диаграмма процесса подготовки данных КАК ДОЛЖНО БЫТЬ

Декомпозируем данную диаграмму, как показано на рисунке 21.

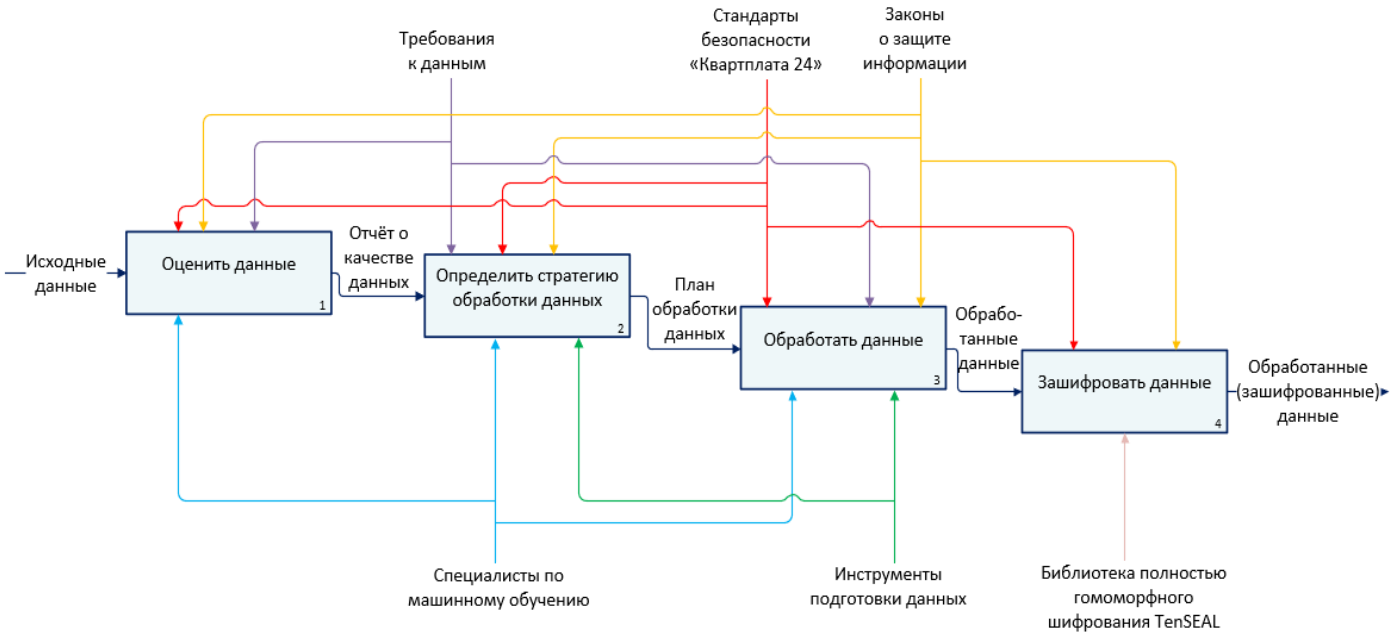

Рисунок 21– Декомпозиция контекстной диаграммы процесса подготовки данных КАК ДОЛЖНО БЫТЬ

На этапе подготовки данных перед их шифрованием настраивают «параметры шифрования, такие как схема шифрования, степень полиномиального модуля, размеры модуля коэффициентов» [5] и другие параметры, влияющие на безопасность и эффективность процесса. После настройки параметров данные зашифровывают и сохраняют в отдельные файлы для их дальнейшей загрузки на сервер.

На сервере происходит обучение модели машинного обучения на зашифрованных данных. BPMN-диаграмма данного процесса показана на рисунке 22. Учитывая ограничения полностью гомоморфного шифрования, поддерживающего только операции сложения и умножения, требуется оптимизация алгоритмов машинного обучения. Например, аппроксимация неполиномиальных функций полиномами низких степеней, что позволит сохранить точность модели при минимальном увеличении вычислительной сложности.

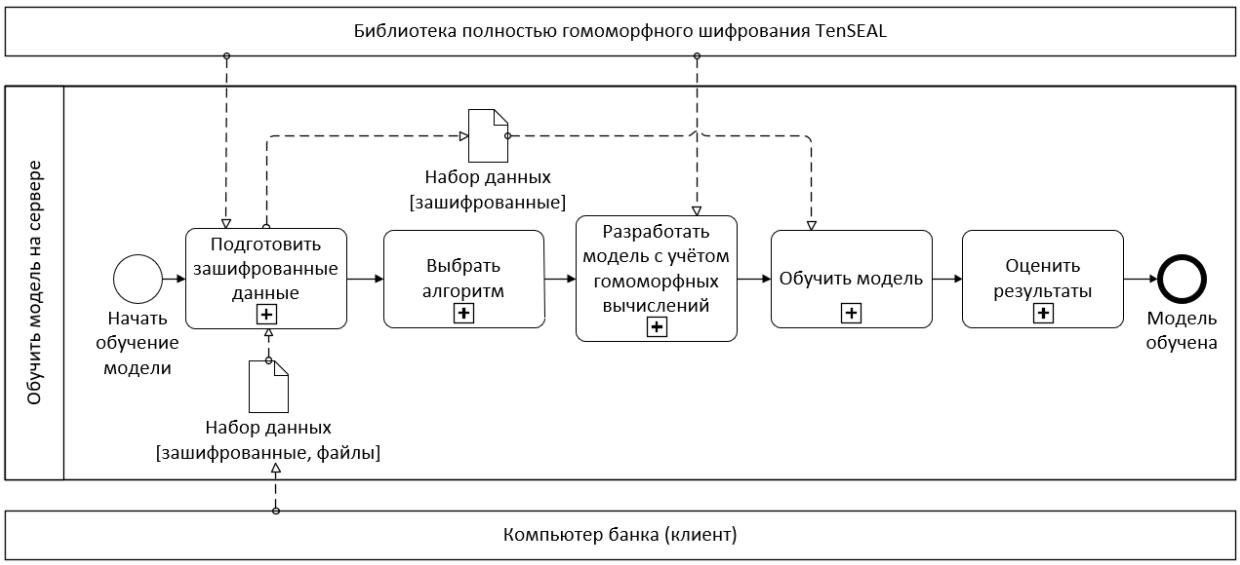

Рисунок 22 – BPMN-диаграмма процесса обучения модели на сервере КАК ДОЛЖНО БЫТЬ

То есть процесс обучения модели на зашифрованных данных включает в себя не только сам процесс обучения, но и адаптацию модели для гомоморфных вычислений. Так модель логистической регрессии требует сигмоидную функцию, которую для гомоморфных вычислений следует представить в виде полинома, полученного с помощью минимаксной аппроксимации, как показано в таблице 2.

2.6 Анализ реализации внедрения полностью гомоморфного шифрования в процесс облачного машинного обучения

Проект обеспечения безопасности данных в облачном машинном обучении с использованием полностью гомоморфного шифрования, требует комплексного подхода, охватывающего как технические, так и управленческие стороны.

Дерево целей проекта структурирует задачи и ожидаемые результаты на разных уровнях, как показано на рисунке 23.

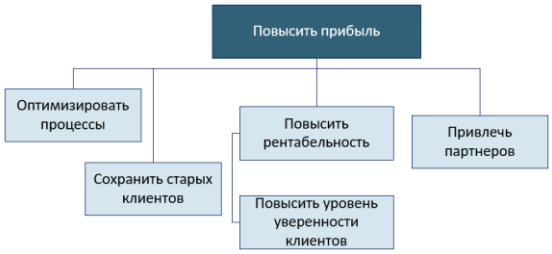

Рисунок 23 – Дерево целей проекта

Для успешного запуска проекта необходим бизнес-план, включающий анализ рынка и конкурентов, материальные ресурсы, организацию, оценку эффективности проекта и анализ ресурсов, в виде таблицы 4.

Таблица 4 – Бизнес-план проекта

Раздел	Краткое содержание
Резюме	• Сущность проекта: внедрение технологии полностью гомоморфного шифрования в процесс машинного обучения. • Оценка потребительской ценности внедряемой библиотеке и потенциала рынка этой продукции связан с необходимостью обеспечения безопасности данных, обрабатываемых в облаке.
Анализ действующего рынка	• Основная идея и цель проекта: обеспечение безопасности облачных вычислений. • Изучение текущих тенденций в использовании полностью гомоморфного шифрования в облачном машинном обучении. • Определение других компаний, предлагающих альтернативные решения для обеспечения безопасности данных в облачных вычислениях. • Разработка маркетинговой стратегии, выбор целевой аудитории, планирование стратегий по расширению клиентской базы. • Изучение текущих технологий, используемых в облачных вычислениях.
Материальные ресурсы	• Организация рабочего пространства: помещение, офисная мебель, компьютеры. • Оборудование: серверы, специализированное оборудование для разработки и тестирования. • Программное обеспечение: средства разработки, инструменты тестирования, системы управления версиями, средства автоматизации развертывания.
Организация	• Составление календарного плана реализации проекта. • Планирование штатной организационной структуры.
Уплата налогов	• Расчет налоговых выплат: НДС, налог на прибыль, на имущество, выплаты в пенсионный и страховой фонды.
Финансовый план	• Предоставление прогнозного отчета о движении средств, о прибылях и убытках.

Продолжение таблицы 4

Раздел	Краткое содержание
Оценка эффективности проекта	• Анализ безубыточности. • Расчет финансовых показателей (рентабельность). • Изучение показателей финансовой эффективности.
Анализ рисков	• Организационные и управленческие риски (риск срыва сроков разработки и релиза продукта, ошибок в подборе персонала, высоких цен на продукцию). • Отработка мер по снижению или полного исключения риска. • Учёт возможных финансовых (недостачи финансирования проекта) и экономических рисков (изменений в системе налогообложения, увеличения ставок налогов).

Чтобы обеспечить эффективное взаимодействие внутри команды проекта для его реализации, необходимо создать матрицу ролей, определяющую обязанности и зоны ответственности каждого участника проекта, как показано в таблице 5.

Таблица 5 – Матрица ролей

Задача	Заказчик	Бизнес-аналитик	Руководитель проекта	Архитектор	Отдел информационной безопасности	Специалисты по машинном обучению
Анализ требований	C	A	R	CI	C	C
Планирование проекта	C	R	A	R	C	C
Проектирование	-	I	R	A	C	R
Разработка моделей с использованием полностью гомоморфного шифрования	-	I	R	A	C	R
Тестирование	-	I	R	CI	A	R
Внедрение решения на облачном сервере	-	I	A	CI	I	R
Обучение пользователей	-	I	A	I	I	R
Мониторинг и поддержка решения	-	C	A	I	C	R

Последовательность шагов, которую проходит клиент при взаимодействии с проектом, отражается в карте пути клиента, приведённую в таблице 6.

Карта пути клиента помогает лучше понять, как клиенты взаимодействуют с продуктом или услугой на каждом этапе своего пути. В контексте внедрения полностью гомоморфного шифрования в процесс облачного машинного обучения, карта пути клиента позволяет выявить точки соприкосновения с пользователями, оптимизировать их опыт и минимизировать возможные препятствия на пути к успешному внедрению.

Таблица 6 – Карта пути клиента

Этап	Ознакомление	Первое использование	Работа в реальных условиях	Обратная связь
Идентификационные данные	Специалист по машинному обучению	Специалист по машинному обучению	Специалист по машинному обучению	Специалист по машинному обучению, разработчики
Контекст	Пользователь только что прошёл обучение (прочитал документацию) по использованию библиотеки.	Пользователь импортирует библиотеку в программу.	Ежедневное использование библиотеки при вычислениях на облачном сервере.	Возможность высказать мнение об использовании библиотеки.
Возможности	Новый инструмент для безопасной обработки данных в облаке.	Защита данных при их обработке.	Безопасность вычислений с зашифрованными данными.	Предложение улучшений.
Ожидания	Простота использования, интуитивный интерфейс.	Безопасность данных.	Быстрая обработка данных, минимальное влияние на производительность.	Внимательное восприятие отзывов.
Эмоциональное состояние	Легкое беспокойство перед первым опытом использования. 😟	Ожидание положительного опыта 😊	Уверенность в безопасности. 😎	Желание внести свой вклад в проект 🤗

Таким образом, каждый из рассмотренных элементов играет важную роль в успешной реализации проекта. Дерево целей устанавливает приоритеты

и направления работы, бизнес-план формирует финансовую и организационную базу, матрица ролей распределяет ответственность, а карта пути клиента помогает выстроить стратегию взаимодействия с клиентами. Все эти составляющие способствуют эффективной реализации проекта по обеспечению безопасности данных в облачном машинном обучении.

Выводы по второй главе

Дана характеристика деятельности компании «Квартплата 24», включающая в себя более 10 облачных сервисов. Рассмотрен сервис по работе с должниками и использование машинного обучения для решения задачи прогнозирования задолженности по услугам ЖКХ. Построены BPMN-диаграмма и IDEF-0 диаграмма существующего процесса обучения модели. Определено наличие такого узкого места, как недостаточная защищённость данных, обрабатываемых в облаке.

Выделены заинтересованные стороны и сформулированы их требования к системе обеспечения данных в облачном машинном обучении. Определены способы реализации требований и построена модель системы обеспечения защиты данных.

Предложены усовершенствование процесса облачного машинного обучения с обеспечением безопасности данных, передаваемых на сервер с использованием полностью гомоморфного шифрования, а также бизнес-план и матрица ролей для реализации проекта.

Глава 3 Обучение модели логистической регрессии на зашифрованных данных

3.1 Модель логистической регрессии для решения прогнозирования задолженности по услугам ЖКХ

Задача прогнозирования задолженности по услугам ЖКХ заключается в оценке способности потребителя коммунальных услуг выполнить свои финансовые обязательства. Это помогает компании «Квартплата 24» и организациям, пользующимися её услугами, оценивать риски неплатежеспособности клиентов, минимизировать убытки от невыплат и повысить эффективность управления долговыми обязательствами.

Решение подобной задачи возможно с использованием методов машинного обучения, так как они «позволяют обрабатывать большие объёмы данных и строить точные прогнозы. Поскольку задача сводится к определению двух взаимоисключающих классов («возникнет задолженность» или «задолженность не возникнет»), она относится к задачам бинарной классификации.

Одной из наиболее распространённых моделей для решения задач» [5] такого типа является логистическая регрессия. Эта модель оценивает вероятность принадлежности потребителя к одному из двух классов, которые можно определить как «Задолженность возникнет» и «Задолженность не возникнет». Логистическая регрессия предсказывает вероятность того, что потребитель окажется добросовестным, на основе набора входных данных, например, возраст потребителя, тип жилья, район, история предыдущих платежей и т. д.

В отличие от многих других алгоритмов классификации, логистическая регрессия предоставляет не только прогноз класса, но и вероятность принадлежности к каждому из классов, что полезно при оценке рисков. Также она основана на относительно простых математических операциях, что делает её вычислительно эффективной даже для больших объёмов данных, в том числе, зашифрованных.

Несмотря на существование более сложных и мощных алгоритмов машинного обучения, логистическая регрессия остаётся популярным выбором для решения задачи прогнозирования задолженности благодаря своей простоте и интерпретируемости и способности, что делает её идеальным первым шагом в построении более сложных моделей.

«Логистическую регрессию можно рассматривать как простую однослойную нейронную сеть, использующую сигмоидальную функцию активации (12)» [5].

Модель логистической регрессии вычисляет взвешенную сумму входных переменных (плюс смещение), и выводит вероятность результата (21):

$$\hat{p} = h_\theta(x) = \sigma(\theta^T x) \tag{21}$$

где h_θ – функция гипотезы с использованием параметров модели;

σ – логистическая функция (12);

θ^T – транспонированный вектор параметров модели.

Если предполагаемая вероятность больше 50 %, то модель предсказывает, что экземпляр принадлежит положительному классу «1» (задолженность не возникнет), а если меньше 50 %, то экземпляр принадлежит отрицательному классу «0» (задолженность возникнет) (22):

$$\hat{y} = \begin{cases} 0, \hat{p} < 0.5 \\ 1, \hat{p} \geq 0.5 \end{cases} \tag{22}$$

«Так как решаемая задача прогнозирования задолженности относится к задачам бинарной классификации, то в качестве функции потери используем бинарную потерю кросс-энтропии, которая измеряет несоответствие между предсказанными вероятностями и истинными бинарными метками, что позволяет оценить, насколько хорошо модель предсказывает вероятность принадлежности к положительному классу» (23) [5]:

$$L = \frac{1}{m} \cdot \sum_{i=1}^{m} [y^{(i)} \cdot \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \cdot \log(1 - \hat{y}^{(i)})] \quad (23)$$

где m – количество экземпляров в наборе данных;

$y^{(i)}$ – фактическая метка для каждого экземпляра (0 или 1);

$\hat{y}^{(i)}$ – предсказанная вероятность принадлежности экземпляра положительному классу;

«В качестве оптимизатора логистической регрессии используем стохастический градиентный спуск» [5], так как он требует меньше вычислений на итерацию, используя только один обучающий пример за раз при обновлении веса модели. Это делает его особенно подходящим для обучения на больших данных, где полный проход может быть очень затратным по времени.

3.2 Полностью гомоморфное шифрование данных с использованием библиотеки TenSEAL

В соответствии с проведенным ранее анализом, для реализации задачи обучения модели логистической регрессии на зашифрованных данных была выбрана библиотека TenSEAL. Это высокоуровневая оболочка для библиотеки Microsoft SEAL на языке Python, которая предоставляет удобные инструменты для шифрования вектором чисел без необходимости иметь высокий уровень понимания технологии полностью гомоморфного шифрования.

TenSEAL поддерживает такие схемы шифрования, как CKKS для вещественных и BFV для целых чисел, что позволяет выбрать оптимальный алгоритм в зависимости от требований к точности и скорости вычислений.

Представим диаграмму вариантов использования данной библиотеки в контексте соответствия требованиям обеспечения безопасности данных в облачном машинном обучении на рисунке 24.

Специалист по машинному обучению может импортировать из библиотеки TenSEAL инструменты для настройки параметров шифрования, осуществлять само шифрование, включая подготовку данных для загрузки на сервер путём их сериализации, а также расшифровывать данные и управлять ключами.

Рисунок 24 – Диаграмма вариантов использования библиотеки TenSEAL

В данной работе роль полностью гомоморфного шифрования в защите данных в облачном машинном обучении исследуется на примере решения задачи прогнозирования задолженности по услугам ЖКХ. Решение этой задачи требует большое количество данных, такие как тип жилья, сведения о потреблении коммунальных услуг и другие, которые невозможно представить только целыми числами без потери точности, что обосновывает необходимость использования схемы CKKS.

Для указания схемы и параметров шифрования в TenSEAL используется специальный объект – TenSEALContext.

Параметры шифрования θ схемы CKKS включают в себя:

– степень полиномиального модуля N;
– размеры модуля коэффициента q.

«Значения обоих параметров зависят от мультипликативной глубины, которая равна сумме необходимых операций умножения. Для обучения модели логистической регрессии требуется следующее количество умножений» [5]:

– 1 умножение для операции скалярного произведения;
– 2 умножения для аппроксимации сигмоидной функции;
– 3 умножения для осуществления обратного распространения ошибки.

Таким образом, мультипликативная глубина равна 1 + 2 + 3 = 6, что определяет количество чисел, составляющих модуль коэффициента.

Степень полиномиального модуля, которую ещё называют коэффициентом масштабирования, должна быть степенью двойки, так как она определяет точность кодирования для двоичного представления числа. Также эта степень напрямую влияет на количество коэффициентов в полиноме открытого текста, размер элементов зашифрованного текста, вычислительную производительность схемы и уровень безопасности. При этом, чем выше степень полиномиального модуля, тем ниже производительность, но выше уровень безопасности.

«Приняв желаемый уровень защищенности, эквивалентный AES, равный 128 битам, степень полиномиального модуля должна быть равна минимум 8192, что позволит группировать до 4096 значений в одном зашифрованном тексте» [5].

Размеры модуля коэффициента представляют собой список двоичных размеров, используя который TenSEAL генерирует список простых чисел этих двоичных размеров. Модуль коэффициента влияет на размер элементов зашифрованного текста, а длина списка указывает уровень схемы, то есть количество поддерживаемых умножений. При этом, чем длиннее данный список, тем выше уровень безопасности.

Так как «желаемый уровень защищённости составляет 128 бит, количество умножений равно 6, а 128 / 6 = 21.3, то в качестве двоичного размера используем 21 бит.

При этом двоичные размеры в количестве, равном количеству умножений, определяются в списке размеров со второй по предпоследнюю позицию, в то время как первый и последний размер в данном списке должны быть больше их по размеру.

Поэтому размеры модуля коэффициента [40, 21, 21, 21, 21, 21, 21, 40] обозначают то, что модуль коэффициента содержит 8 простых чисел: первое и последнее по 40 бит и остальные по 21 бит» [5].

Сведём определённые параметры шифрования схемы CKKS в таблицу 7.

Таблица 7 – Выбранные параметры шифрования схемы CKKS

Степень полиномиального модуля	Размеры модуля коэффициента
8192	[40, 21, 21, 21, 21, 21, 21, 40]

При создании TenSEALContext по умолчанию создаются ключи релинеаризации, а также включена автоматическая релинеаризация и масштабирование, однако для поддержки выполнения операции скалярного произведения векторов требуется отдельное создание ключей Галуа.

Для обеспечения безопасности данных на сервер должны быть загружены зашифрованные данные.

На рисунке 25 представлен код функции encrypt_data, которая с помощью средств библиотеки TenSEAL сначала зашифровывает данные, затем сериализует и архивирует.

```
def encrypt_data(data, filename, ctx_training):
    t_start = time()
    rows = x_train.shape[0]
    zip_name = f'encrypted data//{filename}.zip'

    with ZipFile(zip_name, 'w') as filezip:
        for i in range(rows):
            vector = ts.ckks_vector(ctx_training, data[i].tolist()).serialize()
            filename_with_num = f'{filename}_{i}.hex'
            with open(filename_with_num, 'wb') as file:
                file.write(vector)
            filezip.write(filename_with_num)
            os.remove(filename_with_num)
    t_end = time()
    print(f'Шифрование заняло {int(t_end - t_start)} секунд')
```

Рисунок 25 – Функция шифрования данных

«Таким же образом осуществим сериализацию объекта TenSEALContext для того, чтобы развернуть его на сервере и не настраивать параметры шифрования ещё раз» [5].

Отметим, что обычно для отправления данных на сервер используется формат JSON, однако для записи сериализованных векторов в такой файл их придётся преобразовать в строки, что кратно увеличит размер JSON файла, поэтому такой формат хранения зашифрованных данных подходит на этапе использования, а не обучения модели.

3.3 Разработка модели логистической регрессии, обучаемой на зашифрованных данных

«Перед началом обучением модели на сервер следует загрузить и восстановить на нём сериализованный TenSEALContext, содержащий схему и параметры шифрования, настроенные на клиенте, а также сериализованные зашифрованные векторы» [5] с помощью функции get_ckks_vector, представленной на рисунке 26.

```
def get_ckks_vector(filename, ctx_training):
    zip_name = f'encrypted data//{filename}.zip'
    with ZipFile(zip_name, "r") as myzip:
        myzip.extractall(path='encrypted data')

    ckks_vectors_list = []
    i = 1
    while True:
        filename_with_num = f'encrypted data//{filename}_{i}.hex'
        try:
            with open(filename_with_num, 'rb') as file:
                bytes_vector = file.read()
            ckks_vector = ts.ckks_vector_from(ctx_training, bytes_vector)
            ckks_vectors_list.append(ckks_vector)
            os.remove(filename_with_num)
            i += 1
        except:
            break

    return ckks_vectors_list
```

Рисунок 26 – Функция восстановления зашифрованных файлов в тензоры

Для восстановления TenSEALContext используется функция библиотеки TenSEAL context_from(), но для зашифрованных векторов следует сначала распаковать архив, а затем последовательно открыть каждый из двоичных файлов, из которого уже восстанавливается зашифрованный вектор в виде объекта CKKSVector.

Как уже отмечалось ранее, модель логистической регрессии в качестве функции потери использует бинарную потерю кросс-энтропии, однако для борьбы с переобучением модели следует использовать регуляризацию, которая действует путем добавления штрафа к функции потерь.

Величина штрафа меняется в зависимости от типа регуляризации. L1 регуляризация основана на добавлении штрафа, равного абсолютному значению коэффициентов модели, а L2 регуляризация добавляет штрафную функцию, являющуюся суммой квадратов весов модели, умноженных на гиперпараметр регуляризации.

L1 регуляризация может быть полезна в ситуациях, когда необходимо уменьшить количество признаков для упрощения модели, так как она может обнулять коэффициенты для менее значимых признаков.

Однако для принятия решения о возможности возникновения задолженности следует учитывать все признаки, поэтому следует использовать L2 регуляризацию, которая не обнуляет коэффициенты, а лишь уменьшает их величину.

Тогда функция потерь (23) примет вид (24):

$$L = \frac{1}{m} \cdot \sum_{i=1}^{m} \left[y^{(i)} \cdot \log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) \cdot \log\left(1 - \hat{y}^{(i)}\right) \right] + \frac{\lambda}{2 \cdot m} \cdot \sum_{j=1}^{n} \theta_j^2 \quad (24)$$

где λ – параметр регуляризации, который контролирует степень влияния штрафа на функцию потерь;

θ – вектор параметров модели.

Для обновления параметров применяется следующее правило (25):

$$\theta_j = \theta_j - \alpha \cdot [\frac{1}{m} \cdot \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)}\right) \cdot x^{(i)} + \frac{\lambda}{m} \cdot \theta_j] \quad (25)$$

«Однако, учитывая ограничения полностью гомоморфного шифрования, примем $\alpha = 1$ и $\frac{\lambda}{m} = 0.05$, и тогда правило обновление параметров примет следующий вид» (26) [5]:

$$\theta_j = \theta_j - [\frac{1}{m} \cdot \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)}\right) \cdot x^{(i)} + 0.05 \cdot \theta_j] \quad (26)$$

Поскольку полностью гомоморфное шифрование не позволяет напрямую вычислить сигмоидную функцию, так как её невозможно представить с

помощью сложения и умножения, то её необходимо аппроксимировать полиномами низких степеней.

В результате проведённого анализа источников и литературы по теме исследования, в качестве аппроксимирующей функции была выбрана следующая функция (27):

$$\sigma(x) = -0.004 \cdot x^3 + 0.197 \cdot x + 0.5 \quad (27)$$

Данная функция была получена с помощью минимаксной аппроксимации, и она аппроксимирует сигмоидную функцию на отрезке [-5; 5], поэтому на этапе подготовки данные были нормализованы в соответствии с данным диапазоном. Используя PyTorch, создадим модель логистической регрессии, использующей зашифрованные данные, как показано на рисунке 27.

```
class EncryptedLogisticRegression:
    def __init__(self, torch_lr):
        self.weight = torch_lr.lr.weight.data.tolist()[0]
        self.bias = torch_lr.lr.bias.data.tolist()
        self._delta_w = 0
        self._delta_b = 0
        self._count = 0
def forward(self, enc_x):
        enc_out = enc_x.dot(self.weight) + self.bias
        enc_out = EncryptedLogisticRegression.sigmoid(enc_out)
        return enc_out
    def backward(self, enc_x, enc_out, enc_y):
        out_minus_y = (enc_out - enc_y)
        self._delta_w += enc_x * out_minus_y
        self._delta_b += out_minus_y
        self._count += 1
    def update_parameters(self):
        self.weight -= self._delta_w * (1 / self._count) + self.weight * 0.05
        self.bias -= self._delta_b * (1 / self._count)
        self._delta_w = 0
        self._delta_b = 0
        self._count = 0
    @staticmethod
    def sigmoid(enc_x):
        return enc_x.polyval([0.5, 0.197, 0, -0.004])
    def encrypt(self, context):
        self.weight = ts.ckks_vector(context, self.weight)
        self.bias = ts.ckks_vector(context, self.bias)
    def decrypt(self):
        self.weight = self.weight.decrypt()
        self.bias = self.bias.decrypt()
    def __call__(self, *args, **kwargs):
        return self.forward(*args, **kwargs)
```

Рисунок 27 – Модель обучения логистической регрессии на зашифрованных данных

Данная модель учитывает обновление параметров в соответствии с L2 регуляризацией, вычисляя вес с учётом принятого коэффициента, равного 0.05, а также переопределяет сигмоидальную функцию как полином с коэффициентами 0.5, 0.197, 0 и -0.004.

«Основные результаты, полученные в ходе эксперимента по обеспечению безопасности данных при обучении модели логистической регрессии на сервере, приведены в таблице 8» [5].

Отметим, что данные, на которые производилось обучение, содержат 4400 объектов и 23 признака.

Таблица 8 – Результаты обучения модели логистической регрессии на зашифрованных данных

Параметр	Значение
Вес исходного файла с набором данных (csv)	30.4 Мбайт
Вес одного зашифрованного вектора (hex)	0.388 Мбайт
Время шифрования одного тензора	54 секунды
Точность обучения модели на зашифрованных данных	87.07 %
Точность обучения модели на незашифрованных данных	87.07 %
Среднее время обучения модели на зашифрованных данных	388 секунд

«Вес одного зашифрованного вектора составляет 388 Кбайт, в то время как файл, содержащий набор из 4400 строк, весит 30.4 Мбайт, что свидетельствует о значительном увеличении объема данных после шифрования, что может оказывать влияние на требования к хранению и передаче данных.

Также время шифрования одного тензора, составляющее 54 секунды, демонстрирует то, что процесс шифрования может требовать достаточно много времени, что необходимо учитывать при планировании операций обработки больших данных.

При этом точность модели логистической регрессии, обученной на зашифрованных данных, составила 87 %, что совпадает с точностью модели, обученной на незашифрованных данных. Однако среднее время обучения модели на зашифрованных данных составляет 346 секунд, что указывает на то, что обучение на зашифрованных данных занимает больше времени по сравнению с обучением на незашифрованных данных» [5].

Выводы по третьей главе

Рассмотрено обучение модели логистической регрессии для решения задачи прогнозирования задолженности по услугам ЖКХ. Рассмотрено использование библиотеки TenSEAL для обеспечения безопасности данных в машинном обучении.

Для шифрования данных были определены параметры шифрования в соответствие с желаемым уровнем защищенности, эквивалентным AES, равным 128 битам.

Было установлено, что точность обучения модели на зашифрованных и на незашифрованных совпадает, но при этом значительно увеличивается время обучения и размер файлов с данными.

Глава 4 Оценка эффективности использования полностью гомоморфного шифрования для защиты данных в облачном машинном обучении

4.1 Методика оценки эффективности полностью гомоморфного шифрования для защиты данных в облачном машинном обучении

Методика оценки эффективности полностью гомоморфного шифрования для защиты данных в облачном машинном обучении включает в себя рассмотрение нескольких ключевых критериев, показанных в таблице 9, которые помогут оценить влияние этой технологии как на обеспечение безопасности данных в облаке, так и на решение задачи прогнозирования задолженности по услугам ЖКХ.

Таблица 9 – Критерии оценки эффективности использования полностью гомоморфного шифрования для защиты данных в облачном машинном обучении

Критерий	Содержание
Критерий оценки безопасности данных	оценка безопасности данных, загруженных на облачный сервер.
Критерий оценки качества обучения модели	сравнение точности модели, обученной на зашифрованных и на незашифрованных данных.
Критерий оценки производительности	- сравнение размера файла с зашифрованными и незашифрованными данными; - сравнение времени обучения модели на зашифрованных и на незашифрованных данных; - исследование влияния производительности облачного сервера на производительность полностью гомоморфного шифрования; - исследование влияния размера данных на производительность полностью гомоморфного шифрования.
Критерии оценки бизнес-эффективности	исследование стоимости использования облачного сервера и закупки собственного оборудования.

Методы для проведения оценивания по данной методике включают в себя обучение модели логистической регрессии на данных с разным количеством объектов и признаков.

Эта методика представляет собой комплексный подход к оценке эффективности обеспечения безопасности данных в облачном машинном обучении. Она учитывает как влияние на процессы обеспечения безопасности данных и машинного обучения, так и влияние на бизнес.

4.2 Оценка безопасности данных в облаке

Полностью гомоморфное шифрование позволяет выполнять вычисления над зашифрованными данными, что обеспечивает наивысший уровень безопасности. Это связано с тем, что в облако загружаются предварительно зашифрованные данные и их содержание не будет раскрыто даже в случае несанкционированного доступа к облаку. При этом незашифрованные данные в данном случае будут скомпрометированы с вероятностью 100 %, как показано на рисунке 28.

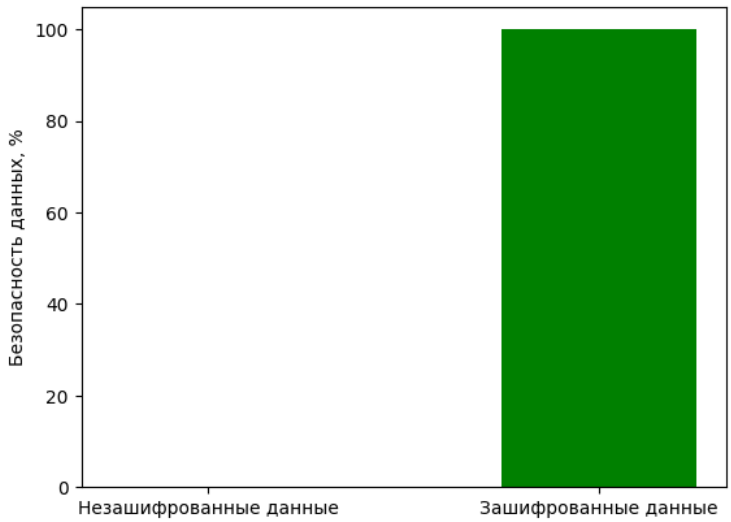

Рисунок 28 – Гистограмма безопасности данных

Качество защищённости данных – основная характеристика системы обеспечения безопасности данных на основе полностью гомоморфного шифрования, но она наиболее трудно поддаётся формальной оценке. В связи с этим её оценка предлагается относительно уже полученного доступа к облачному серверу, когда незашифрованные данные будут скомпрометированы с вероятностью 100 %, а зашифрованные – 0 %.

Отметим, что при настройке параметров шифрования желаемый уровень защищенности, эквивалентный AES, был принят равным 128 битам.

4.3 Оценка качества обучения модели

В соответствии с результатами эксперимента, приведёнными в таблице 9, точность обучения модели логистической регрессии для решения задачи прогнозирования задолженности по услугам ЖКХ совпадает при обучении как на зашифрованных, так и на незашифрованных данных, и составляет 87 %, как показано на рисунке 29.

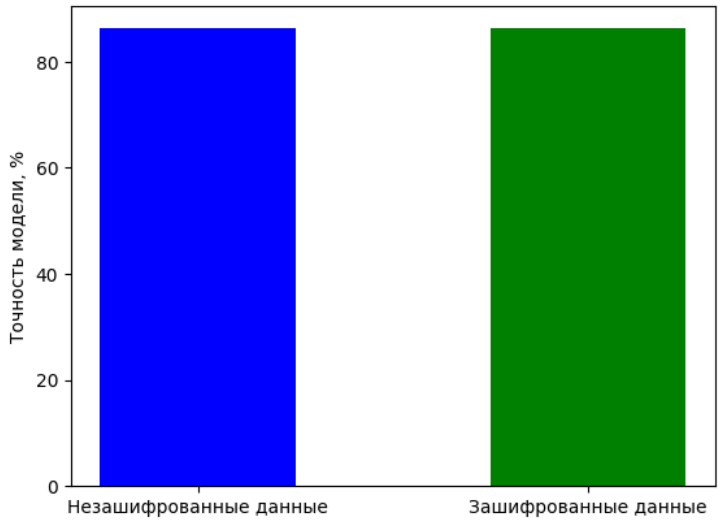

Рисунок 29 – Гистограмма точности модели

«Точность модели логистической регрессии, обученной на зашифрованных данных, составила 87 %, что совпало с точностью модели, обученной на незашифрованных данных. Этот результат подтверждает эффективность применения технологии полностью гомоморфного шифрования для защиты конфиденциальности данных в процессе их обработки, без потери качества моделирования» [5].

4.4 Оценка производительности

4.4.1 Изменение размера файла с данными

В соответствии с результатами эксперимента, приведёнными в таблице 9, размер файла с зашифрованными данными значительно превышает размер файла с незашифрованными данными, как продемонстрировано на рисунке 30.

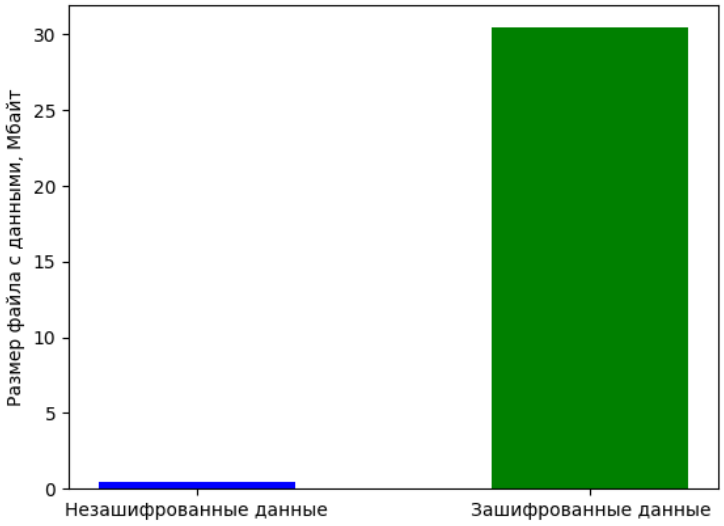

Рисунок 30 – Гистограмма размера файлов с данными

Это свидетельствует о значительном увеличении объема данных после шифрования, что может оказывать влияние на требования к хранению и передаче данных.

4.4.2 Изменение времени обучения модели

«В соответствии с результатами эксперимента, приведёнными в таблице 9, среднее время обучения модели на зашифрованных данных занимает значительно больше времени по сравнению с обучением на незашифрованных данных» [5], как показано на рисунке 31.

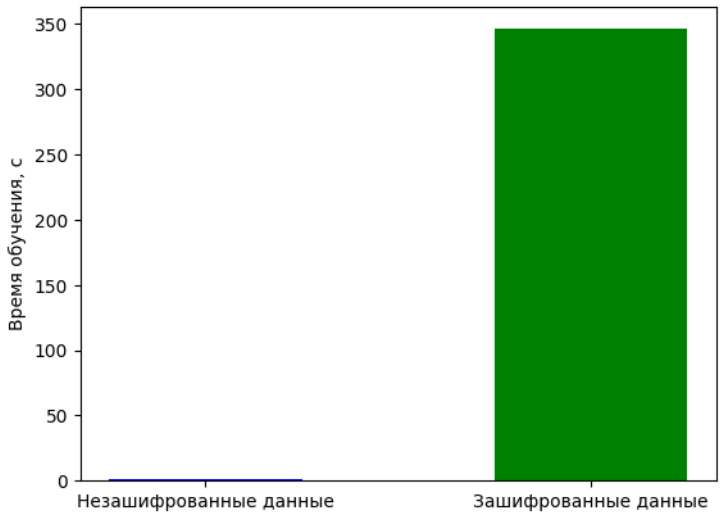

Рисунок 31 – Гистограмма времени обучения модели

При этом обучение проводилось на одинаковых наборах данных, содержащих 4400 объекта и 23 признака.

4.4.3 Зависимость производительности полностью гомоморфного шифрования от конфигурации вычислительных ресурсов

Модель логистической регрессии была обучена на облачном сервере Yandex DataSphere. Выбор этого сервера обоснован как большим количеством возможных конфигураций, так и его удобством, а также возможностью посекундной оплаты.

Для оценки влияния производительности оборудования на время обучения модели логистической регрессии на зашифрованных данных, обучим модель на нескольких доступных конфигурациях, как показано в таблице 10.

Как видно из таблицы 10, то при достижении определённого порога производительности сервера дополнительно увеличение его мощности не целесообразно.

Другими словами, для каждой отдельной задачи следует проводить отдельное тестирование для выбора оптимальной вычислительной конфигурации.

Таблица 10 – Влияние конфигурации сервера на время шифрования и обучения модели логистической регрессии

Конфигурация сервера							Время обучения, с
Имя	vCPU	Количество vCPU	GPU	Количество GPU	RAM, ГБ	VRAM, ГБ	
c1.4	Intel Ice Lake	4	-	-	32	-	389
c1.8		8	-	-	64	-	388
g1.1	Intel Broadwell	8	NVIDIA® Tesla® V100	1	96	32	388
g1.2		16		2	192	64	388

Таким образом, выбор конфигурации зависит, в том числе, от специфики решаемой задачи.

В данной работе проводилось обучение самого базового случая обучения модели логистической регрессии, поэтому использование более мощной конфигурации не привело к существенному ускорению процесса.

4.4.4 Изменение времени обучения и шифрования данных в зависимости от размера данных

Для оценки влияния количества объектов в наборе данных на время обучения и шифрования, из исходного набора данных, содержащего 4400 объекта, создадим наборы данных меньшего размера.

В таблице 11 показана зависимость времени шифрования и обучения от количества объектов в наборе данных.

Таблица 11 – Влияние количества данных на время шифрования и обучения модели логистической регрессии

Размер данных	Время шифрования, с	Время обучения, с
4400 x 23	54	388
3400 x 23	44	297
2400 x 23	29	219
1400 x 23	17	124
400 x 23	4	36

Как видно из таблицы 11, существует линейная зависимость между количеством объектов в данных и временем шифрования и обучения.

Также покажем это на рисунках 32 и 33 соответственно.

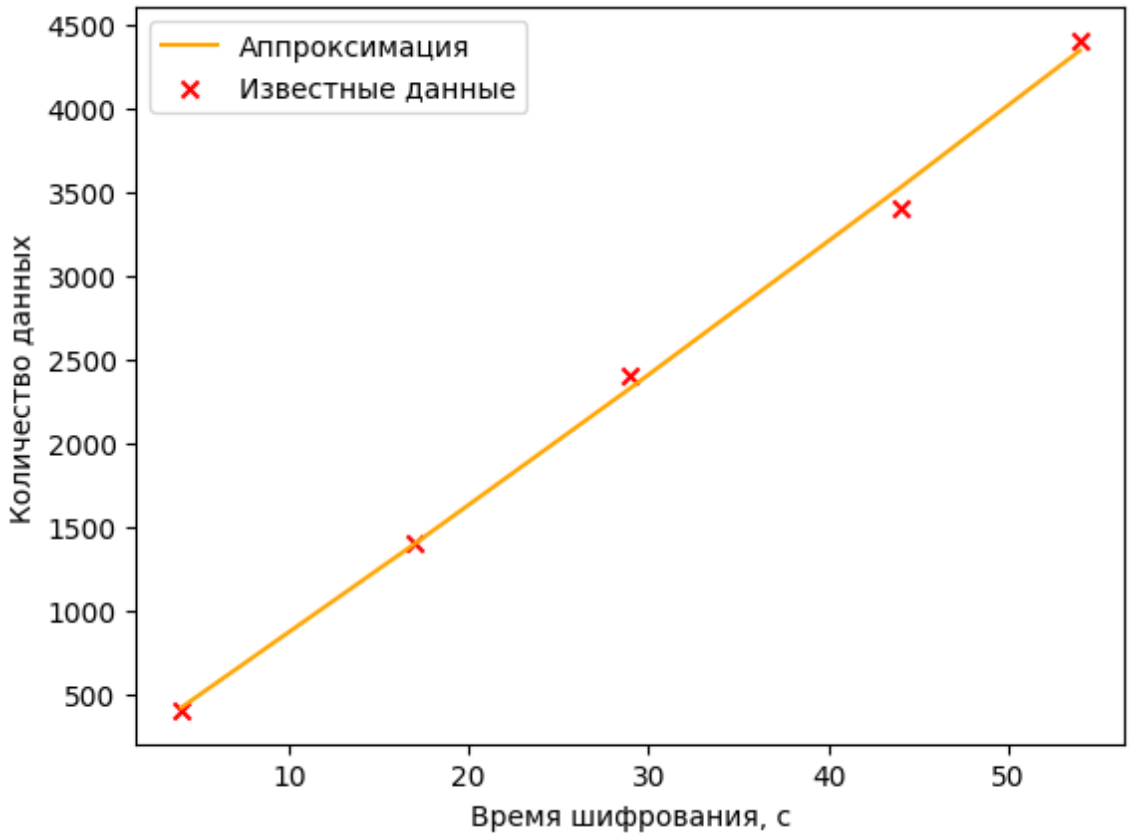

Рисунок 32 – График зависимости времени шифрования от количества данных

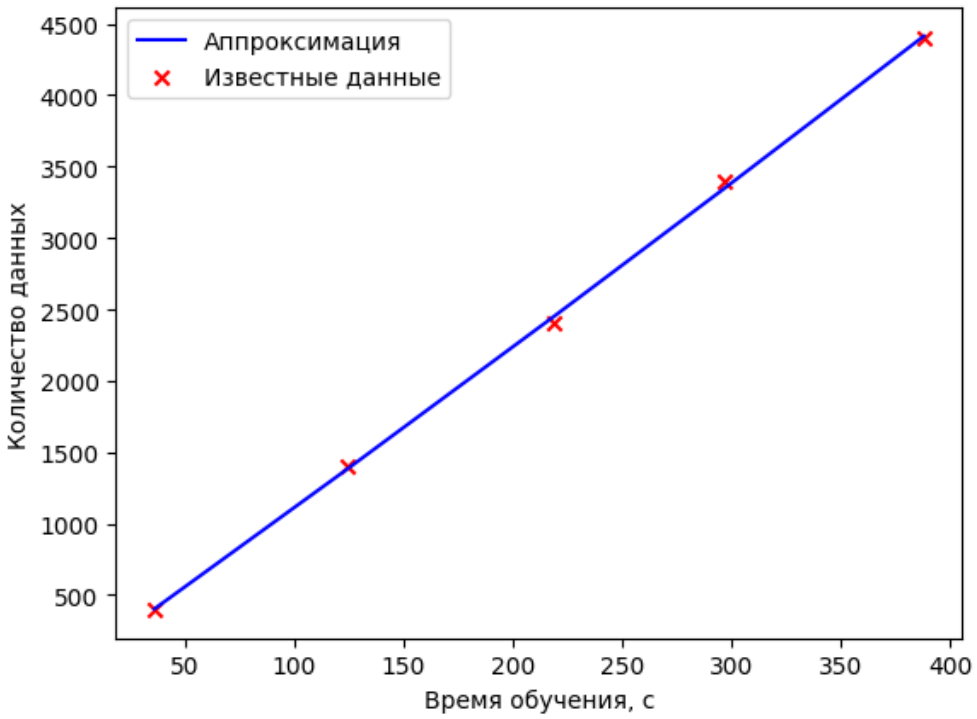

Рисунок 33 – График зависимости времени обучения от количества данных

Для оценки влияния количества признаков в наборе данных на время обучения и шифрования, из исходного набора данных, содержащего 23 признака, создадим наборы данных меньшего размера.

В таблице 12 показана зависимость времени шифрования и обучения от количества признаков в наборе данных при одинаковом количестве объектов.

Таблица 12 – Влияние количества признаков на время шифрования и обучения модели логистической регрессии

Размер данных	Время шифрования, с	Время обучения, с
1400 x 23	17	124
1400 x 21	17	111
1400 x 19	17	104
1400 x 17	17	88
1400 x 15	17	75

Как видно из таблицы 12, количество признаков не влияет на время шифрования. Это обосновано тем, что операция шифрования производится отдельно над каждым объектом, а не признаком. Также покажем это на рисунке 34.

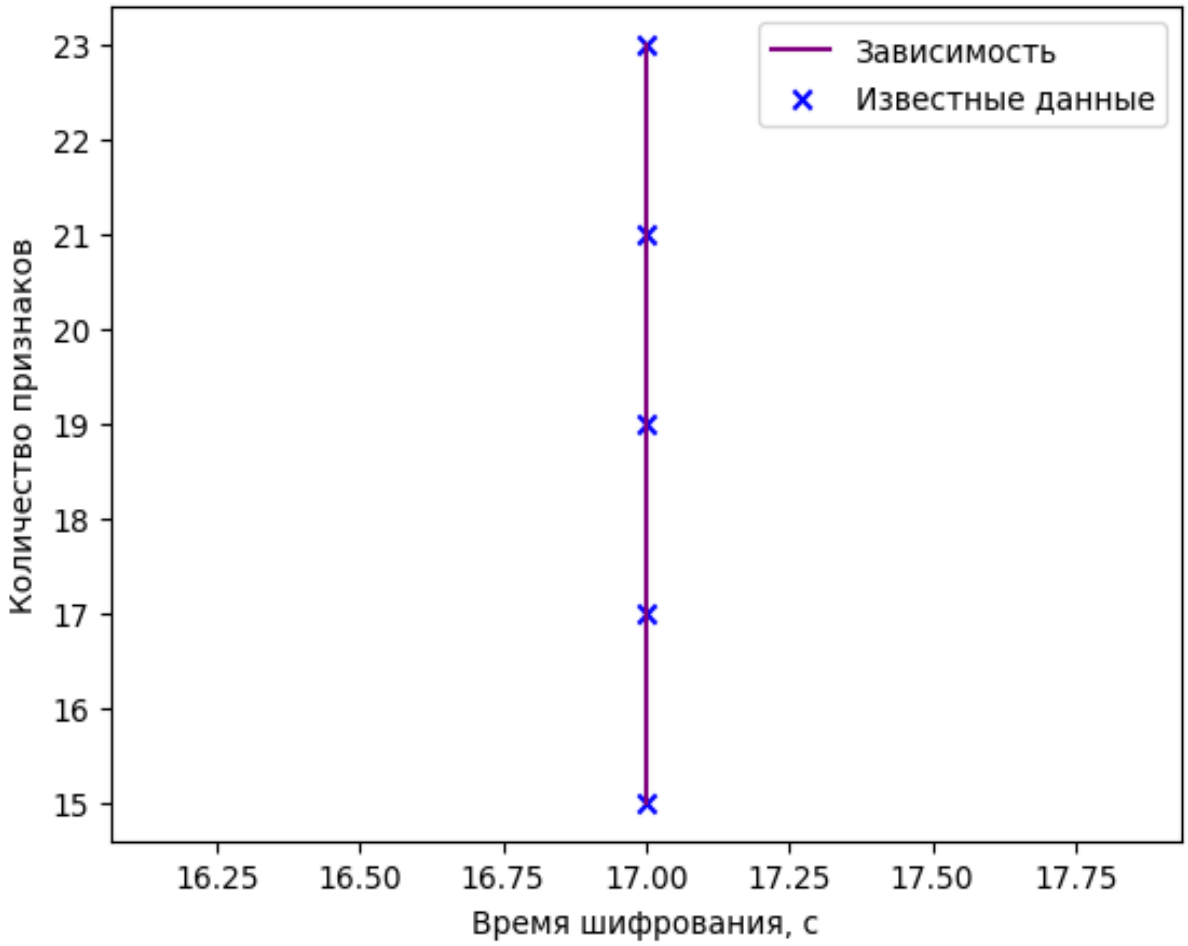

Рисунок 34 – График зависимости времени шифрования от количества признаков

При этом уменьшение количества признаков практически линейно уменьшает время обучения. Это обосновано тем, что в данном случае понижается сложность задачи оптимизации параметров модели. Также покажем это на рисунке 35.

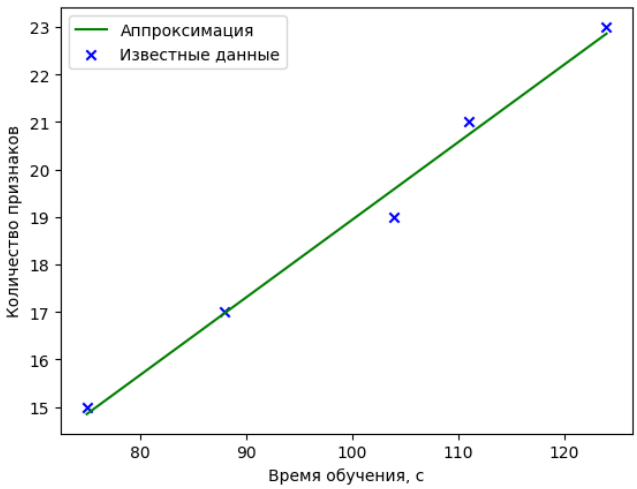

Рисунок 35 – График зависимости времени обучения от количества признаков

«Таким образом, время шифрования данных растёт линейно при увеличении количества данных, но остаётся постоянным при увеличении количества признаков. В то время как время обучения растёт как при увеличении количества данных, так и при увеличении количества признаков, при этом скорость изменяется значительно сильнее при переходе от большого количества данных и признаков к малому.

В результате эксперимента было установлено, что увеличение количества данных ведёт к пропорциональному росту времени шифрования и менее значительному росту времени обучения. В то же время уменьшение количества признаков не оказывает никакого влияния на время шифрования, но существенно снижает время обучения» [6].

4.5 Оценка бизнес-эффективности

Технология полностью гомоморфного шифрования обеспечивает безопасность данных в облачном сервере. Однако кроме программного обеспечения безопасности данных решением проблемы, в том числе, выступает полный

отказ от использования облачного сервера и закупка собственного оборудования.

Тем не менее для реализации данного решения необходимы значительные затраты на закупку, установку и обслуживание оборудования. Кроме того, могут вырасти потребности в вычислительных мощностях, но быстро увеличить производительность собственного оборудования может быть сложнее, чем в облаке.

Оценим бизнес-эффективность использования полностью гомоморфного шифрования, показав в таблице 13 стоимость вычислений на некоторых конфигурациях Yandex DataSphere и стоимость оборудования, используемых для этого.

Таблица 13 – Сравнение стоимости использования полностью гомоморфного шифрования на облачном сервере со стоимостью отказа от его использования и закупки собственного оборудования

Имя конфигурации в Yandex DataSphere	Арендуемое оборудование	Закупка собственного оборудования		Стоимость вычислений в Yandex DataSphere, руб.		
		Стоимость, руб.	Общая стоимость, руб.	За 1 час, руб.	За месяц (42 часа), руб.	За месяц (720 часов)
c1.8	Процессор Intel Xeon Gold 6330	~600 000	~615 000 + 50 000 (дополнительное оборудование)	34.6	1 451.5	24 883.2
	Оперативная память 64 ГБ DDR4 ECC	~15 000				
g1.2	Процессор Intel Xeon E5-2697A v4 (2 шт.)	~800 000	~3 840 000 + 50 000 (дополнительное оборудование)	1 002.2	42 092	721 612.8
	Оперативная память 192 ГБ DDR4 ECC	~40 000				
	Видеокарта NVIDIA Tesla V100 PCIe (2 шт.)	~3 000 000				

Продолжение таблицы 13

Имя конфигурации в Yandex DataSphere	Арендуемое оборудование	Закупка собственного оборудования		Стоимость вычислений в Yandex DataSphere, руб.		
		Стоимость, руб.	Общая стоимость, руб.	За 1 час, руб.	За месяц (42 часа), руб.	За месяц (720 часов)
g2.4	Процессор AMD EPYC 7742 (2 шт.)	~1 800 000	9 900 000 + 50 000 (дополнительное оборудование)	2 004.5	84 188.2	1 443 456
	Оперативная память 476 ГБ DDR4 ECC	~100 000				
	Видеокарта NVIDIA Ampere A100 (4 шт.)	~8 000 000				

Очевидно, что кроме процессора, оперативной памяти и видеокарты обязательно должны быть куплены материнская карта, накопитель, блок питания и корпус, и установлена система охлаждения. В среднем такие затраты могут составить от 10 000 до 70 000 рублей, они были приняты равными 50 000 рублей.

Так как полностью гомоморфное шифрование оказывает значительное влияние на производительность, то обучение и дообучение моделей, а также дополнительные эксперименты для нужд «Квартплата 24», предусматривающие использование мощного оборудования, в среднем будут занимать около 2 часов в день или 42 часа в месяц. Однако так же в таблице 4.5 показана стоимость аренды облачного сервера при круглосуточном использовании в течение месяца.

Тогда при использовании конфигурации c1.8, которая является оптимальной для решения задачи прогнозирования задолженности по услугам ЖКХ, закупка собственного оборудования покажет себя наиболее эффективной через:

$$\frac{615\,000\ +\ 50\,000}{1\,451.5} = 458.2 \text{ мес.}$$

Однако при условии круглосуточного использования облачного сервера закупка собственного оборудования покажет себя более целесообразной, так как его примерная окупаемость составит:

$$\frac{615\,000\ +\ 50\,000}{24\,883.2} = 26.7 \text{ мес.}$$

Таким же образом рассчитаем окупаемость оборудования по сравнению с использованием облачного сервера и полностью гомоморфного шифрования, как показано в таблице 14.

Таблица 14 – Окупаемость оборудования при разном количестве часов работы в месяц

Имя конфигурации в Yandex DataSphere	Окупаемость оборудования при разном количестве часов работы в месяц	
	42 часа в месяц	720 часов в месяц
c1.8	458.2 мес. (38,2 лет)	26.7 мес. (2,2 года)
g1.2	92.4 мес. (7.7 лет)	5.4 мес.
g2.4	118.2 мес. (9.8 лет)	6.9 мес.

При использовании оборудования всего 42 часа в месяц аренда облачного сервера значительно выгоднее, чем покупка собственного оборудования. Это связано с тем, что основные затраты при покупке являются фиксированными и не зависят от времени использования. Аренда же позволяет платить только за фактически использованные ресурсы. Однако если оборудование должно эксплуатироваться в течение целых суток на протяжении многих месяцев, то отказ от использования облачного сервера в пользу закупки собственного оборудования будет более выгодным. Также, если учитывать дополнительные затраты, например, на обслуживание, то даже в этом случае они не должны превышать сумму месячной аренды сервера.

Таким образом, программное обеспечение безопасности данных, обрабатываемых в облаке, является более дешёвым и простым способом в случае использования его ресурсов не больше нескольких часов в день. И хотя также нужно учитывать затраты на аренду облачного сервера, такие затраты будут существенно ниже, чем закупка собственного оборудования. Кроме того, для каждой из задач машинного обучения, как было продемонстрировано в пункте 4.4.3, существует своя оптимальная конфигурация облачного сервера, отличающаяся по стоимости. Другими словами, использование облачного сервера позволяет как быстро увеличить, так и уменьшить производительность для экономии денежных средств. Кроме прямых затрат, важным фактором является оценка косвенных выгод, таких как улучшение репутации компании и повышение конкурентоспособности на рынке за счёт уверенности клиентов в безопасности данных. Эти нематериальные выгоды сложно измерить в денежном эквиваленте, но они играют значительную роль в общей оценке бизнес-эффективности.

Выводы по четвёртой главе

Разработаны критерии оценки эффективности использования полностью гомоморфного шифрования в облачном машинном обучении. В соответствие с критериями оценена безопасность данных, качество обучения модели, производительность и бизнес-эффективность. Установлено, что полностью гомоморфное шифрование не влияет на точность обучения модели, но увеличивает размер файла с данными и время обучения. При этом время обучения модели линейно зависит от количества объектов и признаков в наборе данных, а время шифрования зависит только от количества объектов. Стоимость закупки собственного оборудования, а также затраты на его эксплуатацию и поддержку значительно превышает стоимость аренды облачного сервера в случае его использования до двух часов в день. Однако при круглосуточном использовании оборудования следует отказаться от аренды в пользу его закупки.

Заключение

В данной работе было проведено исследование использования полностью гомоморфного шифрования для защиты данных в облачном машинном обучении.

В ходе исследования были изучены принципы работы полностью гомоморфного шифрования, его преимущества и ограничения. Особое внимание уделено задачам LWE и RLWE, которые лежат в основе современных схем полностью гомоморфного шифрования. Это позволило понять, как оно обеспечивает безопасность данных, позволяя выполнять вычисления над зашифрованными данными без их расшифровки.

Также был проведён обзор популярных библиотек, таких как HELib, Microsoft SEAL, PALISADE и других. В результате анализа была выбрана библиотека TenSEAL, которая предоставляет удобный интерфейс для на языке Python и поддерживает схемы CKKS и BFV, что делает её подходящей для задач машинного обучения.

Была дана характеристика деятельности компании «Квартплата 24», в том числе сервиса по работе с должниками, также был изучен существующий процесс машинного обучения и выявлены узкие места, связанные с недостаточной защищённостью данных. После этого были сформулированы требования заинтересованных сторон, что позволило определить ключевые направления для внедрения полностью гомоморфного шифрования.

Для анализа реализации внедрения полностью гомоморфного шифрования в процессы машинного обучения были построены дерево целей проекта, предложен бизнес-план и матрица ролей для его реализации. Последовательности шагов, которую клиент проходит при взаимодействии с проектом была отражена с помощью карты пути клиента.

В качестве примера задачи, которая решается в компании «Квартплата 24» с помощью машинного обучение была выбрана задачи прогнозирования

задолженности по услугам ЖКХ. Для её решения была разработана модель логистической регрессии, адаптированная для работы с зашифрованными данными с использованием библиотеки TenSEAL. В частности, для адаптации модели для гомоморфных вычислений над зашифрованными данными была реализована аппроксимация сигмоидной функции полиномами низкой степени, полученная с помощью минимаксной аппроксимации. Также был разработан механизм шифрования и упаковки данных на клиенте для их последующей передачи на сервер.

Разработанная модель была оценена с точки зрения различных критериев, например, точности и производительности. Так точность модели, обученной на зашифрованных данных, совпала с точностью модели, обученной на незашифрованных данных, и составила 87 % в обоих случаях. Но оценка производительности выявила, что использование полностью гомоморфного шифрования значительно увеличивает время обучения и размер данных, например, обучение модели логистической регрессии на наборе данных, содержащих 3400 объекта и 23 признака составляет 297 секунд. Однако эти затраты компенсируются высоким уровнем безопасности.

Так как единственной альтернативой использованию полностью гомоморфного шифрования в защите данных в облачном машинном обучении является закупка собственного оборудования и исключение передачи данных на сторонний сервер, было проведено сравнение стоимости оборудования арендуемого сервера со стоимостью обучения за 1 час вычислений.

Это показало, что использование облачного сервера в сочетании с полностью гомоморфным шифрованием является более экономически выгодным решением по сравнению с закупкой собственного оборудования в случае его использования до нескольких часов в день, что делает его привлекательным для компаний, обучающих модель на сервере, но эксплуатирующих её на собственном оборудовании.

Так при эксплуатации облачного сервера до 2 часов в день или 42 часов в месяц, его использование более выгодно, чем закупка собственного оборудования, так как она окупится только через 38 лет.

Результаты работы могут быть применены компанией «Квартплата 24» для повышения безопасности данных в процессе машинного обучения, а также другими организациями, использующими облачные серверы для обработки конфиденциальной информации. Внедрение полностью гомоморфного шифрования позволит не только обеспечить защиту данных, но и укрепить доверие клиентов и партнёров, что является важным конкурентным преимуществом на рынке. При этом предложенное решение позволяет продолжить использование облачного сервера без необходимости закупки дорогостоящего оборудования.

Таким образом, в данной работе были успешно решены поставленные задачи, что позволило достичь цели исследования и доказать эффективность использования полностью гомоморфного шифрования для защиты данных в облачном машинном обучении.

Список используемой литературы и используемых источников

1. Басалова, Г. В. Основы криптографии : учебное пособие / Г. В. Басалова. – 3-е изд. – Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. – 282 с.

2. Варновский, Н. П., Шокуров, А. В. Гомоморфное шифрование. – Труды ИСП РАН. 2007.

3. Гаража, А. А., Герасимов, И. Ю., Николаев, М. В., Чижов И.В. Об использовании библиотек полностью гомоморфного шифрования. – International Journal of Open Information Technologies. 2021. №3.

4. Жиль, З. Курс криптографии / Земор Жиль ; перевод Шуликовская В. В. – Москва, Ижевск : Регулярная и хаотическая динамика, Институт компьютерных исследований, 2019. – 256 с.

5. Хаустова И. В., Аникина О. В. Использование полностью гомоморфного шифрования для защиты данных в машинном обучении в облаке // Вестник науки. 2024. № 4 (75). С. 482–491.

6. Хаустова И. В., Аникина О. В. Влияние размера данных на производительность полностью гомоморфного шифрования для защиты данных в облачном машинном обучении // Вестник науки. 2024. № 11 (80) том 2. С. 1160–1168.

7. Albrecht M., Cid C., and Faugère J. C., Fitzpatrick R., Perret L. On the Complexity of the BKW Algorithm on LWE. – Cryptology ePrint Archive, Paper 2012/636, 2012.

8. Albrecht M., Player R., Scott S. On the concrete hardness of learning with errors. – Cryptology ePrint Archive, Report 2015/046, 2015.

9. Albrecht M., Robert F, Florian G. On the efficacy of solving lwe by reduction to unique-svp. – Cryptology ePrint Archive, Report 2013/602, 2013.

10. Albrecht M., Chase M., Chen H. Homomorphic encryption standard. – Cryptology ePrint Archive, Report 2019/939, 2019.

11. Alperin-Sheriff J., Peikert C. Faster Bootstrapping with Polynomial Error. – Cryptology ePrint Archive, Paper 2014/094, 2014.

12. Arora S., Ge R. New algorithms for learning in presence of errors. Proceedings of the 38th International Colloquim Conference on Automata, Languages and Programming. – Volume Part I. – ICALP'11. – Berlin, Heidelberg : Springer-Verlag, 2011. – P. 403-415.

13. Boneh D., Goh E. J., Nissim K. Evaluating 2-dnf formulas on ciphertexts. Theory of Cryptography. – Springer Berlin Heidelberg, 2005. – P. 325–341.

14. Bourse F., Sanders O., Traoré J. Improved Secure Integer Comparison via Homomorphic Encryption. In Topics in Cryptology. – The Cryptographers' Track at the RSA Conference 2020, Lecture Notes in Computer Science, Volume 12006. – P. 391–416.

15. Chabanne H., Wargny A., Milgram J., Morel C., Prouff E. Privacy-preserving classification on deep neural network. – IACR Cryptol. ePrint Arch, 2017. – P. 1–35.

16. Chen H, Gilad-Bachrach R., Han K., Huang Z., Jalali A., Laine K., Lauter K. Logistic regression over encrypted data from fully homomorphic encryption. – BMC Medical Genomics, Volume 11, Article number 81. – P. 56–67.

17. Cheon J. H., Kim A., Kim M., Song Y. S. Homomorphic encryption for arithmetic of approximate numbers. – Cryptology ePrint Archive, Report 2016/421, 2016.

18. Chillotti I., Gama N., Georgieva M., Izabachène M. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. – Cryptology ePrint Archive, Report 2016/870, 2016.

19. Costache A., Laine K., Player R. Evaluating the effectiveness of heuristic worst-case noise analysis in fhe. – Cryptology ePrint Archive, Report 2019/493, 2019.

20. Costache A., Smart N. P. Which ring based somewhat homomorphic encryption scheme is best? – Cryptology ePrint Archive, Report 2015/889, 2015.

21. Crawford J. L. H., Gentry C., Halevi S., Platt D., Shoup V. Doing real work with FHE: The case of logistic regression. – Proc. 6th Workshop Encrypted Comput. Appl. Homomorphic Cryptogr, 2018. – 1–12.

22. Ducas L., Micciancio D. Fhew: Bootstrapping homomorphic encryption in less than a second. – Cryptology ePrint Archive, Report 2014/816, 2014.

23. El Gamal T. A public key cryptosystem and a signature scheme based on discrete logarithms. – Springer-Verlag, 1985. – P. 10–18.

24. Fan J., Vercauteren F. Somewhat practical fully homomorphic encryption. – Cryptology ePrint Archive, Report 2012/144, 2012.

25. Gama N., Izabachène P., Nguyen P.Q., Xie X. Structural Lattice Reduction: Generalized Worst-Case to Average-Case Reductions and Homomorphic Cryptosystems. – Lecture Notes in Computer Science book series, Volume 9666, 2016.

26. Gentry C. A Fully Homomorphic Encryption Scheme : Ph. D. thesis / Craig Gentry. – Stanford, CA, USA : Stanford University, 2009.

27. Gentry C., Sahai A., B. Waters B. Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based. – Lecture Notes in Computer Science, Volume 8042, 2013.

28. Gilad-Bachrach R., Dowlin N., Laine K., Lauter E. K., Naehrig M., Wernsing J. CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy. – JMLR Workshop and Conference Proceedings, Volume 48. – P. 201–210.

29. Guerra-Manzanares A., L. Lopez J.L., Maniatakos M., Shamout F. Privacy-Preserving Machine Learning for Healthcare: Open Challenges and Future Perspectives. – First International Workshop, TML4H 2023, Virtual Event, 2023. – P. 25–40.

30. Halevi S., Shoup V. Design and implementation of HElib: a homomorphic encryption library. – Cryptology ePrint Archive, Paper 2020/1481, 2020.

31. Hesamifard E., Takabi H., Ghasemi M. CryptoDL: Deep neural networks over encrypted data. [Электронный ресурс]. URL: https://arxiv.org/abs/1711.05189 (дата обращения: 05.03.2024).

32. Ioffe S., Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift [Электронный ресурс]. URL: https://arxiv.org/abs/1502.03167 (дата обращения: 05.03.2024).

33. Kahya A. Machine Learning over Encrypted Data With Fully Homomorphic Encyption. – Master of Science, Middle East Technical University. 2022.

34. Kim A., Papadimitriou A., Polyakov Y. Approximate Homomorphic Encryption with Reduced Approximation Error. – Cryptology ePrint Archive, Paper 2020/1118, 2020.

35. Liu J., Juuti M., Lu Y., Asokan N. Oblivious neural network predictions via minionn transformations. ACM SIGSAC Conference on Computer and Communications Security, October 30 – November 03, 2017. – P. 619–631.

36. Liu Y., Lyubashevsky V., Micciancio D. On Bounded Distance Decoding for General Lattices. – Lecture Notes in Computer Science, Volume 4110, 2006.

37. Martins P., Sousa L., Mariano A. A survey on fully homomorphic encryption: An engineering perspective. – ACM Computing Surveys, Volume 50, Number 6, 2018. – P. 83:1–83:33.

38. Micciancio D., Regev O. Lattice-based Cryptography. PostQuantum Cryptography. – Berlin, Heidelberg : Springer Berlin Heidelberg, 2009. – P. 147–191.

39. Mohassel P., Zhang Y. Secureml: A system for scalable privacy-preserving machine learning. – IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017. – P. 19–38.

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

41. Obla A., Gong X., Aloufi A., Hu P., Takabi D. Effective Activation Functions for Homomorphic Evaluation of Deep Neural Networks [Электронный

ресурс]. URL: https://repository.rit.edu/theses/10593 (дата обращения: 20.11.2024).

42. Pascal P. Public-key cryptosystems based on composite degree residuosity classes. Advances in Cryptology. – Springer Berlin Heidelberg, 1999. – P. 223–238.

43. Qayyum A., Haris M. Cryptanalysis of lwe and sis-based cryptosystems by using quantum annealing. – ПДМ. Приложение №16, 2023.

44. Regev O. On lattices, learning with errors, random linear codes, and cryptography. – Journal of the ACM, 2009. Vol. 56, iss. 6. – P. 1-40.

45. Riazi M. S., Weinert C., Tkachenko O., Songhori E. M., Schneider T., Koushanfar F. Chameleon: A hybrid secure computation framework for machine learning applications. – 13 ACM Asia Conference on Information, Computer and Communications Security, ACM, Jun 4–8, 2018.

46. Rivest R L, Adleman L, Dertouzos M L. On data banks and privacy homomorphisms. – Foundations of Secure Computation, Academia Press, 1978. – P. 169–179.

47. Rivest R. L., Shamir A., Adleman L. A method for obtaining digital signatures and public-key cryptosystems. – Commun. ACM, Volume 21, Number 2, 1978. – P. 120–126.

48. Schnorr C., Euchner M. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. – Mathematical Programming, 1994. – 08. – Vol. 66. – P. 181–199.

49. Smart N. P., Vercauteren F. Fully Homomorphic SIMD Operations. – Designs, Codes and Cryptography, Number 71, 2014 – P. 57–81.

50. 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: 05.03.2024).

51. Vizitiu A., Nita C. I., Puiu A., Suciu C., Itu L. M. Applying Deep Neural Networks over Homomorphic Encrypted Medical Data [Электронный ресурс]. URL: https://doi.org/10.1155/2020/3910250 (дата обращения: 05.03.2024).

52. Xu R., Nathalie Baracaldo N., Joshi J. Privacy-Preserving Machine Learning: Methods, Challenges and Directions [Электронный ресурс]. URL: https://arxiv.org/abs/2108.04417 (дата обращения: 05.03.2024).

53. Zvika B., Gentry C., Vaikuntanathan V. Fully homomorphic encryption without bootstrapping. – Cryptology ePrint Archive, Report 2011/277, 2011.

54. Zvika B. Fully homomorphic encryption without modulus switching from classical gapsvp. – Cryptology ePrint Archive, Report 2012/078, 2012.