УДК 004

Хаустова И.В.
студент группы ПИмд-2205а
Тольяттинский государственный университет
(г. Тольятти, Россия)

Научный руководитель:
Аникина О.В.
канд. тех. наук, доцент кафедры прикладной математики и информатики
Тольяттинский государственный университет
(г. Тольятти, Россия)

ИСПОЛЬЗОВАНИЕ ПОЛНОСТЬЮ ГОМОМОРФНОГО ШИФРОВАНИЯ ДЛЯ ЗАЩИТЫ ДАННЫХ В МАШИННОМ ОБУЧЕНИИ В ОБЛАКЕ

Аннотация: *в работе рассматривается проблема недостаточной защищённости конфиденциальных данных, используемых банком для машинного обучения в облаке. Для решения проблемы было проанализировано использование полностью гомоморфного шифрования в машинном обучении и аппроксимирование сигмоидной функции. Была решена задача кредитного скоринга с обеспечением полной безопасности данных, для чего с помощью библиотеки TenSEAL данные на клиенте были зашифрованы с помощью схемы CKKS и загружены на сервер, где на зашифрованных данных была обучена модель логистической регрессии, продемонстрировавшая такую же точность, как и модель, обученная на незашифрованных данных.*

Ключевые слова: *полностью гомоморфное шифрование, CKKS, логистическая регрессия, полиномиальная аппроксимация, конфиденциальное машинное обучение.*

Введение

В данной работе рассматривается ситуация, в которой банк, использующий в своей деятельности машинное обучение, столкнулся с нехваткой

вычислительной мощности, поэтому было принято решение арендовать для этих целей облачный сервер. Но так как данные передаются на сторонний сервер, возникает проблема обеспечения безопасности данных, так как, например, утечка данных может привести к репутационным потерям и штрафам за нарушение правил информационной безопасности.

Решить данную проблему позволяет технология полностью гомоморфного шифрования, позволяющая выполнять вычисления (сложение и умножение) над зашифрованными данными без их расшифровки, что обеспечивает наивысший уровень безопасности [2], так как содержание данных никогда не раскрывается.

Целью данной работы является разработка модели машинного обучения на зашифрованных данных на примере решения задачи кредитного скоринга, а новизна работы заключается в реализации клиент-серверного сценария использования библиотеки TenSEAL путём передачи на сервер уже зашифрованных на клиенте данных.

Для того, чтобы показать, насколько эффективно обучение модели логистической регрессии на зашифрованных данных, предварительно обучим эту же модель, но не на зашифрованных данных.

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

Большинство работ [2, 6, 7, 8, 9], посвященных использованию полностью гомоморфного шифрования для машинного обучения, описывают только шифрование вывода работы нейронной сети, и не касаются этапа обучения. Например, работа [7] 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 [9] 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 [8] с помощью библиотеки HELib была обучена нейронная

сеть на основе стохастического градиентного спуска (SGD), что продемонстрировало эффективность обучения на зашифрованных данных.

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

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

Первым решением проблемы поддержки неполиномиальных функций было использование полиномиальной замены, представленной в работе [5], в которой авторы предложили замену сигмоиды квадратичным полиномом, который, однако, может вызвать нестабильность во время обучения нейросети [3]. Более эффективный и часто используемый способ поддержки неполиномиальных функций заключается в аппроксимации неполиномиальных функций полиномами низкой степени.

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

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

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

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

Настройка параметров полностью гомоморфного шифрования и шифрование данных

В качестве библиотеки полностью гомоморфного шифрования выбрана TenSEAL, основанная на Microsoft SEAL, которая является наиболее используемой библиотекой полностью гомоморфного шифрования [2, 6, 7].

Прежде чем приступить к шифрованию данных, настроим схему и параметры шифрования с помощью специального объекта TenSEALContext

В качестве схемы шифрования используем CKKS, так как она позволяет производить вычисления с вещественными числами, параметры шифрования которой включают в себя степень полиномиального модуля N и размеры модуля коэффициента q.

Приняв желаемый уровень защищенности, эквивалентный AES, равный 128 битам, степень полиномиального модуля следует определить равной 8192, что позволит группировать до 4096 значений в одном зашифрованном тексте.

Так как требуемое количество операций умножения равно 6 (1 для скалярного произведения, 2 для аппроксимации сигмоидной функции и 3 для обратного распространения ошибки), а желаемый уровень защищённости всё так же равен 128 бит, то в качестве двоичного размера следует использовать 21 бит, так как $128 / 6 \approx 21$.

Тогда размеры модуля коэффициента – это список чисел [40, 21, 21, 21, 21, 21, 21, 40], обозначающий, что модуль коэффициента будет содержать 8 простых чисел: первое и последнее по 40 бит и остальные по 21 бит.

Для обеспечения безопасности данных на сервер нужно загружать уже зашифрованные на клиенте данные, для чего для каждой из выборок создадим

зашифрованные вектора, которые сериализуем, то есть представим в виде двоичных данных, запишем в отдельные файлы и архивируем.

Таким же образом сериализуем объект TenSEALContext для того, чтобы развернуть его на сервере и не настраивать параметры шифрования ещё раз.

Модель логистической регрессии, обучаемая на зашифрованных данных

Логистическую регрессию можно рассматривать как простую однослойную нейронную сеть, использующую сигмоидальную функцию активации (1).

Для разработки модели логистической регрессии используем PyTorch, а в качестве набора данных – «Credit score classification», скачанный с Kaggle, который включает в себя практически полную информацию о заёмщиках. Также перед тем, как приступить к обучению модели на сервере, следует загрузить и восстановить на нём сериализованный TenSEALContext, содержащий схему и параметры шифрования, а также сериализованные зашифрованные векторы.

Так как решаемая задача кредитного скоринга относится к задачам бинарной классификации, то в качестве оптимизатора используем стохастический градиентный спуск, а в качестве функции потерь – бинарную потерю кросс-энтропии.

Для борьбы с переобучением используем L2 регуляризацию, и тогда функция потерь будет иметь следующий вид:

$$L = \frac{1}{m} \cdot \sum_{i=1}^{m} \left[y^{(i)} \cdot \log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) \cdot \log\left(1 - \hat{y}^{(i)}\right) \right] + \frac{\lambda}{2 \cdot m} \cdot \sum_{j=1}^{n} \theta_j^2 \qquad (3)$$

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;

λ – параметр регуляризации;

θ – вектор параметров модели.

Для обновления параметров применяется следующее правило:

$$\theta_j = \theta_j - \alpha \cdot [\frac{1}{m} \cdot \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) \cdot x^{(i)} + \frac{\lambda}{m} \cdot \theta_j] \quad (4)$$

Однако, учитывая ограничения полностью гомоморфного шифрования, примем $\alpha = 1$ и $\frac{\lambda}{m} = 0.05$, и тогда правило обновления параметров примет следующий вид:

$$\theta_j = \theta_j - [\frac{1}{m} \cdot \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) \cdot x^{(i)} + 0.05 \cdot \theta_j] \quad (5)$$

Анализ результатов

Основные результаты, полученные в ходе эксперимента по обеспечению безопасности данных при обучении модели логистической регрессии на сервере, приведены в таблице 1.

Таблица 1 – Характеристика модели логистической регрессии для решения задачи кредитного скоринга

Параметр	Значение
Вес исходного файла с набором данных (csv)	30 407 Кбайт
Вес одного зашифрованного вектора (hex)	429 Кбайт
Время шифрования одного тензора ([3523, 19])	59 секунд
Точность обучения модели на зашифрованных данных	86.27 %
Точность обучения модели на незашифрованных данных	86.27 %
Среднее время обучения модели на зашифрованных данных	346 секунд

Вес одного зашифрованного вектора составляет 429 Кбайт, в то время как файл, содержащий набор из 20000 строк, весит всего 30 407 Кбайт, что свидетельствует о значительном увеличении объема данных после шифрования, что может оказывать влияние на требования к хранению и передаче данных. Время шифрования одного тензора составляет 59 секунд и демонстрирует, что процесс шифрования является достаточно времязатратным.

При этом точность модели логистической регрессии, обученной на зашифрованных данных, составила 86%, что совпадает 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. Среднее время обучения модели на зашифрованных данных равно 346 секунд, что указывает на то, что обучение на зашифрованных данных занимает больше времени по сравнению с обучением на незашифрованных данных.

Заключение

В данной работе рассматривалась проблема обеспечения безопасности данных, используемых банком для машинного обучения, так как утечка или раскрытие данных могут привести к репутационным потерям и штрафам за нарушение правил информационной безопасности.

В рамках проведенного исследования было осуществлено решение задачи кредитного скоринга путём разработки модели логистической регрессии, обучаемой на данных, зашифрованных с использованием технологии полностью гомоморфного шифрования.

В результате обучения модели логистической регрессии на зашифрованных данных была достигнута точность 86.27 %, что совпало с точностью модели логистической регрессии, обученной на незашифрованных данных, и что демонстрирует применимость технологии полностью гомоморфного шифрования для обеспечения безопасности данных в машинном обучении, так как оно не влияет на качество прогноза, но при этом обеспечивает максимальный уровень защиты данных.

Тем не менее, необходимо отметить, что использование полностью гомоморфного шифрования влечёт за собой дополнительные вычислительные затраты, например, увеличение размера данных и времени обучения модели.

Направления для дальнейшего исследования могут включать в себя оптимизацию процесса отправки данных на сервер и получения данных с него, так как сейчас это выполняется вручную.

Список литературы:

1. Маршалко Г. Б., Труфанова Ю. А., Полиномиальные аппроксимации некоторых функций активации нейронных сетей, Информатика и автоматизация, 2022, выпуск 21, том 1, 161–180.

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

3. 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.

4. 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.

5. 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, Volume 48. – P. 201–210.

6. Kahya A. Machine Learning over Encrypted Data With Fully Homomorphic Encyption. – Master of Science, METU. 2022.

7. 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.

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

9. 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).

Khaustova I.V.
Togliatti State University
(Togliatti, Russia)

Anikina O.V.
Togliatti State University
(Togliatti, Russia)

USING FULLY HOMOMORPHIC ENCRYPTION TO PROTECT DATA IN MACHINE LEARNING IN THE CLOUD

***Abstract**: the paper addresses the issue of insufficient protection of confidential data used by banks for machine learning in the cloud. To solve this issue, the use of fully homomorphic encryption in machine learning and the approximation of the sigmoid function were analyzed. The problem of credit scoring was solved with full data security, for which the TenSEAL library was used to encrypt the data on the client side using the CKKS scheme and upload it to the server. On the server, a logistic regression model was trained on the encrypted data, demonstrating the same accuracy as a model trained on unencrypted data.*

***Keywords**: fully homomorphic encryption, CKKS, logistic regression, polynomial approximation, confidential machine learning*