DOI 10.37882/2223-2966.2026.03.21

ПРИМЕНЕНИЕ АЛГОРИТМОВ ГРАДИЕНТНОГО БУСТИНГА ДЛЯ ДЕТЕКЦИИ ФИНАНСОВОГО МОШЕННИЧЕСТВА

APPLICATION OF GRADIENT BOOSTING ALGORITHMS FOR DETECTING FINANCIAL FRAUD

N. Kurtash
S. Bryutova
S. Molodyakov

Summary. The current problem of automatic detection of fraudulent transactions in the banking sector, complicated by a strong imbalance in data classes, is considered. The evolution of boosting methods from the classic AdaBoost to the modern XGBoost is presented. A comparative analysis of the basic and optimized model configurations was performed on a combined dataset of 300,000 transactions with a 5 % fraud rate. Hyperparameter optimization was performed using a Bayesian approach (the TPE algorithm of the Optuna framework). The results showed an increase in F1-score by 57.8 % and Precision by 135.6 %, indicating a significant reduction in false positives. The dominant influence of tree depth and learning rate on classification quality was revealed. It was demonstrated that proper gradient boosting tuning allows for effective class separation without the use of additional data balancing methods.

Keywords: machine learning, gradient boosting, AdaBoost, XGBoost, fraud detection, unbalanced data, hyperparameter optimization.

Курташ Никита Сергеевич
Санкт-Петербургский политехнический университет Петра Великого
nikitakurtash@gmail.com
Брютова София Даниловна
Санкт-Петербургский политехнический университет Петра Великого
britovasofia@gmail.com
Молодяков Сергей Александрович
доктор технических наук, профессор,
Санкт-Петербургский политехнический университет Петра Великого
samolodyakov@mail.ru

Аннотация. Рассматривается актуальная проблема автоматического выявления мошеннических транзакций в банковской сфере, осложнённая сильной несбалансированностью классов данных. Представлена эволюция методов бустинга от классического AdaBoost до современного XGBoost. На комбинированном датасете из 300 000 транзакций с 5 % долей мошенничества проведён сравнительный анализ базовой и оптимизированной конфигураций модели. Оптимизация гиперпараметров выполнена с применением байесовского подхода (алгоритм TPE фреймворка Optuna). Результаты показали рост F1-score на 57,8 % и Precision на 135,6 %, что свидетельствует о значительном сокращении ложноположительных срабатываний. Выявлено доминирующее влияние глубины деревьев и скорости обучения на качество классификации. Продемонстрировано, что правильная настройка градиентного бустинга позволяет эффективно разделять классы без применения дополнительных методов балансировки данных.

Ключевые слова: машинное обучение, градиентный бустинг, AdaBoost, XGBoost, детекция мошенничества, несбалансированные данные, оптимизация гиперпараметров.

Введение

Детекция мошеннических операций — актуальная задача в финансовой сфере, решаемая в том числе с помощью машинного обучения. По данным Nilson Report [1], глобальные потери от мошенничества с платежными картами в 2023 году превысили 32 миллиарда долларов. Это делает задачу автоматического обнаружения подозрительных транзакций важной и практически значимой для исследования методов классификации.

Особую сложность задача детекции мошенничества представляет из-за сильной несбалансированности классов: доля мошеннических операций обычно составляет менее 1–5 % от общего объема транзакций. В таких условиях традиционные метрики качества, такие как accuracy, становятся малоинформативными, а модели склонны к предсказанию только мажоритарного класса. Это требует применения специализированных алгоритмов и метрик оценки качества.

Среди методов машинного обучения хорошо зарекомендовали себя ансамблевые алгоритмы, в частности методы бустинга. Алгоритм AdaBoost (Adaptive Boosting), предложенный Фройндом и Шапире в 1997 году, стал основой для развития целого семейства методов. Современные реализации градиентного бустинга — XGBoost, LightGBM и CatBoost — активно применяются для работы с табличными данными и часто показывают высокие результаты в соревнованиях по машинному обучению.

Целью данной работы является исследование эффективности алгоритмов градиентного бустинга в задаче детекции мошеннических транзакций, анализ влияния ключевых гиперпараметров на качество классификации и демонстрация потенциала оптимизации базовых конфигураций моделей.

Теоретические основы алгоритмов бустинга

A. Классический алгоритм AdaBoost

Бустинг (boosting) представляет собой мета-алгоритм машинного обучения, основная идея которого — комбинирование множества слабых классификаторов в единую сильную модель. Алгоритм AdaBoost (Adaptive Boosting), предложенный Freund и Schapire [2], стал первым успешным алгоритмом бустинга и заложил теоретический фундамент для последующего развития этого направления.

В AdaBoost каждому образцу обучающей выборки присваивается вес, который адаптивно изменяется в процессе обучения. На каждой итерации строится новый слабый классификатор (обычно решающий пень — decision stump), после чего веса образцов пересчитываются: неверно классифицированные примеры получают больший вес, что заставляет последующие классификаторы фокусироваться на сложных случаях. Итоговое предсказание формируется как взвешенное голосование всех построенных классификаторов.

Математически AdaBoost минимизирует экспоненциальную функцию потерь. Для бинарной классификации с метками $y \in \{-1, +1\}$ итоговый классификатор имеет вид $H(x) = \text{sign}\left(\sum \alpha_t h_t(x)\right)$, где h_t — слабые классификаторы, α_t — их веса, определяемые качеством классификации на текущей итерации [3].

B. Градиентный бустинг и XGBoost

Градиентный бустинг, предложенный Friedman [4], обобщил идеи AdaBoost, интерпретировав бустинг как градиентный спуск в функциональном пространстве. Вместо изменения весов образцов, градиентный бустинг на каждой итерации аппроксимирует градиент функции потерь, что позволяет работать с произвольными дифференцируемыми функциями потерь.

XGBoost (eXtreme Gradient Boosting), разработанный Chen и Guestrin [5], представляет собой оптимизированную реализацию градиентного бустинга. Главными улучшениями стали регуляризация второго порядка для предотвращения переобучения, эффективная обработка разреженных данных и поддержка параллельных вычислений. В отличие от классического градиентного бустинга, XGBoost использует аппроксимацию функции потерь разложением Тейлора до второго порядка, что позволяет учитывать кривизну функции потерь.

Целевая функция XGBoost на итерации t имеет вид:

$$L^{(t)} = \sum_i \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \gamma T + \frac{1}{2}\lambda \sum_j w_j^2$$

где g_i и h_i — градиент и гессиан функции потерь соответственно, f_t — предсказание нового дерева, T — количество листьев, w_j — веса листьев, γ и λ — коэффициенты регуляризации. Параметр γ штрафует за сложность дерева (число листьев), а λ — за величину весов в листьях, что помогает контролировать переобучение.

C. Применение бустинга в задачах детекции мошенничества

Исследования последних лет демонстрируют высокую эффективность алгоритмов бустинга в разных задачах детекции в том числе и в детекции финансового мошенничества. В работе Ileberi и др. [6] комбинация AdaBoost с методом SMOTE позволила достичь точности 99,67 % на европейском датасете кредитных карт.

Randhawa и др. [7] исследовали устойчивость ансамбля на основе AdaBoost к зашумлённым данным. Авторы искусственно добавляли шум в обучающую выборку и измеряли качество классификации с помощью коэффициента корреляции Мэтьюса (MCC). Даже при добавлении 30 % шума модель сохранила MCC = 0,942, что говорит о высокой робастности бустинга — модель все так же верно классифицирует транзакции несмотря на значительное искажение данных.

Еще одним преимуществом алгоритмов бустинга является возможность анализа важности признаков для обеспечения интерпретируемости моделей [8].

Методология исследования

A. Характеристика набора данных

Для проведения экспериментов использовался датасет Credit Card Fraud Detection [11], предоставленный исследовательской группой ULB Machine Learning Group (Université Libre de Bruxelles) и размещённый на платформе Kaggle. Датасет содержит реальные транзакции европейских держателей кредитных карт за сентябрь 2013 года.

Набор данных включает 284 807 транзакций, из которых 492 (0,172 %) являются мошенническими. В целях обеспечения конфиденциальности исходные признаки были преобразованы методом главных компонент (PCA),

в результате чего получены 28 анонимизированных числовых признаков V1–V28. Признаки Time (время с момента первой транзакции) и Amount (сумма транзакции) не подвергались PCA-преобразованию. Целевая переменная Class принимает значение 1 для мошеннических транзакций и 0 для легитимных.

Для экспериментов датасет был дополнен синтетическими записями до 300 000 транзакций с сохранением структуры признаков и увеличением доли мошеннических операций до 5 % (15 000 записей). Такое соотношение позволяет исследовать поведение модели в условиях умеренного дисбаланса классов. Данные были разделены на обучающую (80 %) и тестовую (20 %) выборки методом стратифицированного разделения. Все признаки были стандартизированы.

B. Архитектура модели и гиперпараметры

В качестве основного алгоритма использовался XGBoost. Он был выбран из-за высокой производительности на табличных данных, встроенной поддержки несбалансированных классов через параметр scale_pos_weight, наличия механизмов регуляризации и возможности GPU-ускорения.

Исследуемые гиперпараметры модели представлены в таблице 1.

Таблица 1.

Пространство поиска гиперпараметров

Параметр	Диапазон	Описание
n_estimators	200–1500	Количество деревьев в ансамбле
learning_rate	0.01–0.3	Скорость обучения
max_depth	3–12	Максимальная глубина деревьев
min_child_weight	1–10	Минимальная сумма весов в листе
subsample	0.6–1.0	Доля образцов для каждого дерева
colsample_bytree	0.6–1.0	Доля признаков для каждого дерева
gamma	0–5	Минимальное улучшение для разбиения

C. Процедура оптимизации

Для поиска оптимальной конфигурации гиперпараметров использовался фреймворк Optuna [9] с алгоритмом Tree-structured Parzen Estimator (TPE). TPE относится к методам байесовской оптимизации, он на основе результатов предыдущих экспериментов алгоритм строит вероятностные модели для хороших и плохих конфигураций параметров, а затем выбирает следующую точку для исследования так, чтобы максимизировать вероятность улучшения.

В качестве целевой функции была выбрана максимизация метрики F1-score. Эта метрика представляет собой гармоническое среднее между Precision (точностью) и Recall (полнотой) и вычисляется как $F1 = 2 \cdot (Precision \cdot Recall)/(Precision+Recall)$. F1-score хорошо подходит для несбалансированных данных, так как он учитывает и способность модели не давать ложных срабатываний, и способность находить все положительные примеры.

В процессе исследования было проведено 100 экспериментов с различными комбинациями гиперпараметров. Optuna автоматически управляет процессом поиска: первые 10 экспериментов выполняются со случайными параметрами для накопления начальной статистики, после чего алгоритм TPE начинает направленный поиск в перспективных областях пространства параметров.

D. Метрики оценки качества

Оценка качества моделей проводилась с использованием набора разных метрик. Так, Accuracy отражает общую долю правильных предсказаний. Precision показывает, какая доля транзакций, помеченных как мошеннические, действительно является таковыми. Recall измеряет, какую долю реальных мошеннических транзакций удалось обнаружить. F1-score объединяет Precision и Recall в одну метрику. ROC-AUC характеризует способность модели разделять классы при различных порогах классификации.

Результаты экспериментов

A. Сравнительный анализ моделей

Базовая модель XGBoost с параметрами по умолчанию (100 деревьев, скорость обучения 0.1) сравнивалась с оптимизированной моделью. В ходе оптимизации лучший результат был достигнут на 61-й итерации. Найденная конфигурация сильно отличается от базовой: модель использует значительно больше деревьев (1254 вместо 100) при меньшей скорости обучения (0.026 вместо 0.1), а также более глубокие деревья (глубина 10). Такое сочетание позволяет модели постепенно и аккуратно обучаться на сложных паттернах данных.

Результаты сравнения представлены в таблице 2.

Целевая метрика F1-score, которую мы максимизировали в ходе оптимизации, выросла с 0.5284 до 0.8336, что составляет +57.75 %. Вместе с этим значение Precision выросло более чем в 2,3 раза (с 0.3949 до 0.9303).

B. Анализ качества классификации

Для оценки качества разделения классов была рассчитана площадь под ROC-кривой (AUC). У базовой моде-

Таблица 2.

Сравнение метрик базовой и оптимизированной моделей

Метрика	Базовая	Оптимизированная	Изменение
Accuracy	0.9160	0.9822	+7.23%
Precision	0.3949	0.9303	+135.61%
Recall	0.7986	0.7551	−5.45%
F1-score	0.5284	0.8336	+57.75%
ROC-AUC	0.9022	0.9158	+1.51%

ли AUC составил 0.9022, у оптимизированной — 0.9158. Хотя разница в абсолютных значениях невелика (+1.5 %), обе модели демонстрируют хорошую способность разделять классы.

Матрицы ошибок (рис. 1) позволяют детально проанализировать распределение предсказаний моделей.

Анализ матриц ошибок подсвечивает главное практическое улучшение: количество ложноположительных срабатываний сократилось с 4328 до 200, то есть в 21.6 раза. Это означает, что оптимизированная модель значительно реже ошибочно блокирует легитимные транзакции, что напрямую влияет на удобство пользователей и снижает нагрузку на службу поддержки. При этом модель по-прежнему обнаруживает большинство мошеннических транзакций: 2670 из 3536 (75.5 %).

C. Анализ важности признаков

Одним из преимуществ градиентного бустинга является возможность оценки важности признаков. На рисунке 2 представлены 15 наиболее важных признаков оптимизированной модели.

Наиболее важными признаками оказались V25, V23, V18 и V24. Относительно равномерное распределение важности среди топ-15 признаков показывает, что модель использует информацию из различных источников, не опираясь чрезмерно на один из признаков.

Обсуждение результатов

Полученные результаты демонстрируют эффективность градиентного бустинга в задаче детекции мошенничества и подтверждают важность настройки гиперпараметров. Наиболее значимым результатом является рост Precision более чем в 2,3 раза (на 135.6 %). Из всех транзакций, помеченных системой как подозрительные, подавляющее большинство действительно являются мошенническими. Для финансовых организаций это критически важно — каждое ложноположительное срабатывание означает заблокированную легитимную транзакцию, недовольного клиента и затраты на ручную проверку. Незначительное снижение Recall (−5.45 %) является приемлемым компромиссом. Модель по-прежнему обнаруживает 75.5 % мошеннических транзакций, при этом существенно сокращая ложные срабатывания.

Анализ оптимальной конфигурации выявил важную особенность: лучшие результаты достигаются при низкой скорости обучения (0.026) в сочетании с большим количеством деревьев (1254). Это совпадает с наблюдением, что постепенное обучение с большим числом итераций способствует лучшей генерализации модели [10].

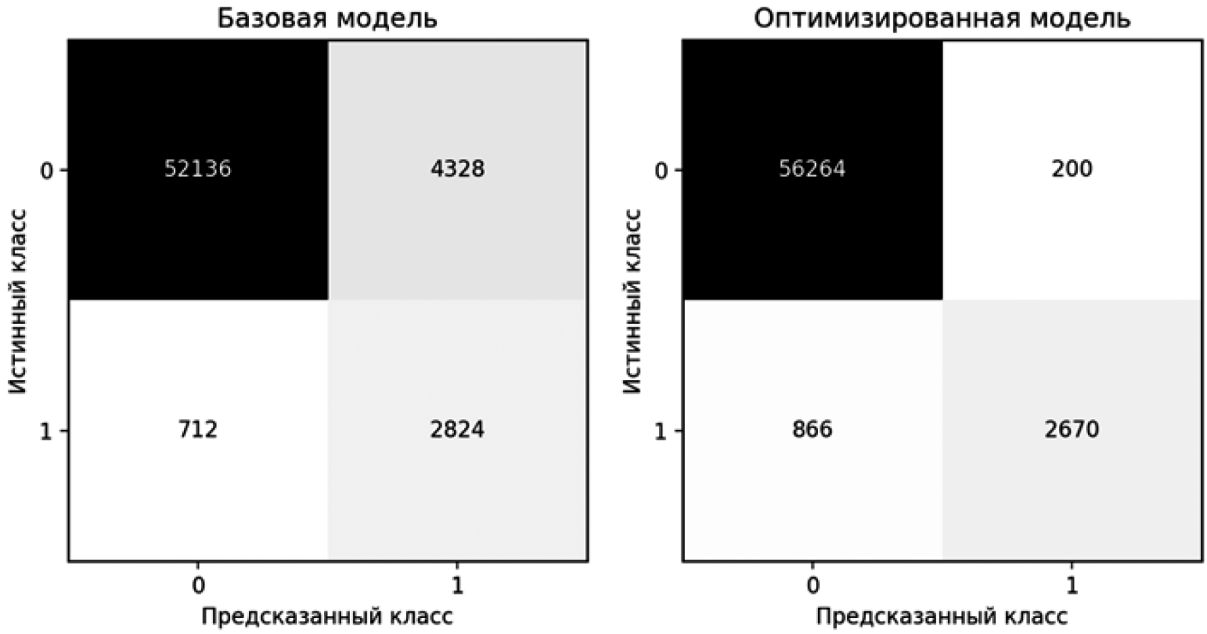

Рис. 1. Матрицы ошибок базовой и оптимизированной моделей

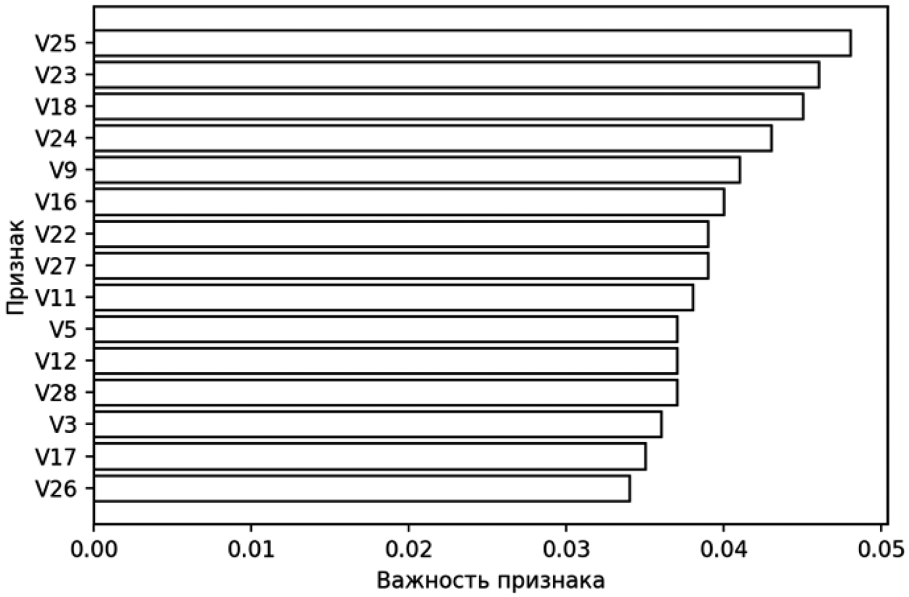

Рис. 2. Важность признаков оптимизированной модели (топ-15)

Глубина деревьев 10 говорит о том, что для данной задачи модели необходимо строить достаточно сложные правила классификации, чтобы уловить неочевидные паттерны мошеннического поведения.

Полученный F1-score = 0.8336 сопоставим с результатами Ileberi и др. [6] и Randhawa и др. [7], однако наше исследование проводилось без использования методов передискретизации, таких как SMOTE. Это важный результат: правильно настроенный градиентный бустинг способен эффективно работать с несбалансированными данными из коробки, без дополнительных этапов предобработки, которые усложняют пайплайн и могут исказить данные.

Заключение

В работе исследована эффективность алгоритмов градиентного бустинга в задаче детекции мошеннических транзакций. Экспериментально показано, что алгоритм XGBoost с оптимизированными гиперпараметрами существенно превосходит базовую конфигурацию. F1-score вырос на 57.75 %, а Precision — на 135.61 %.

По результатам исследования сформулированы следующие выводы. Во-первых, алгоритмы градиентного бустинга демонстрируют высокую эффективность в задачах детекции мошенничества на несбалансированных данных, что подтверждается значительным улучшением всех ключевых метрик качества. Во-вторых, настройка гиперпараметров оказывает критическое влияние на качество классификации: разница между базовой и оптимизированной конфигурациями составила более 50% по метрике F1-score. В-третьих, среди исследованных параметров наибольшее влияние на результат оказывают глубина деревьев и скорость обучения — именно их правильное сочетание позволяет модели находить сложные паттерны в данных без переобучения. В-четвёртых, при правильной настройке градиентный бустинг способен эффективно работать с несбалансированными данными без применения дополнительных методов балансировки классов, таких как SMOTE или взвешивание классов, что упрощает построение практических систем детекции.

Практическая значимость работы состоит в демонстрации возможности существенного повышения эффективности систем детекции мошенничества путём систематической оптимизации моделей машинного обучения. Предложенный подход может быть адаптирован для других задач классификации на несбалансированных данных.

В качестве направлений дальнейших исследований можно выделить сравнительный анализ различных алгоритмов бустинга (LightGBM, CatBoost), исследование комбинации градиентного бустинга с методами балансировки классов, а также адаптацию подхода для онлайн-обучения в потоковых данных.

ЛИТЕРАТУРА

1. Card Fraud Losses Reach $32.34 Billion [Электронный ресурс] // Nilson Report. — 2023. — December. — URL: https://nilsonreport.com (дата обращения: 15.12.2025).
2. Freund Y.A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting / Y. Freund, R.E. Schapire // Journal of Computer and System Sciences. — 1997. — Vol. 55, № 1. — P. 119–139. — DOI: 10.1006/jcss.1997.1504. — URL: https://www.sciencedirect.com/science/article/pii/S002200009791504X
3. Schapire R.E. Explaining AdaBoost / R.E. Schapire // Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik / eds. B. Schölkopf, Z. Luo, V. Vovk. — Berlin, Heidelberg: Springer, 2013. — P. 37–52. — DOI: 10.1007/978-3-642-41136-6_5. — URL: https://link.springer.com/chapter/10.1007/978-3-642-41136-6_5
4. Friedman J.H. Greedy Function Approximation: A Gradient Boosting Machine / J.H. Friedman // The Annals of Statistics. — 2001. — Vol. 29, № 5. — P. 1189–1232. — DOI: 10.1214/aos/1013203451. — URL: https://www.researchgate.net/publication/2424824.
5. Chen T. XGBoost: A Scalable Tree Boosting System / T. Chen, C. Guestrin // Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (San Francisco, August 13–17, 2016). — New York: ACM, 2016. — P. 785–794. — DOI: 10.1145/2939672.2939785. — URL: https://dl.acm.org/doi/10.1145/2939672.2939785.
6. Ileberi E. Performance Evaluation of Machine Learning Methods for Credit Card Fraud Detection Using SMOTE and AdaBoost / E. Ileberi, Y. Sun, Z. Wang // IEEE Access. — 2021. — Vol. 9. — P. 165286–165294. — DOI: 10.1109/ACCESS.2021.3134330. — URL: https://www.researchgate.net/publication/357077948.
7. Randhawa K. Credit Card Fraud Detection Using AdaBoost and Majority Voting / K. Randhawa, C.K. Loo, M. Seera [et al.] // IEEE Access. — 2018. — Vol. 6. — P. 14277–14284. — DOI: 10.1109/ACCESS.2018.2806420. — URL: https://www.researchgate.net/publication/323213023
8. Bentéjac C.A Comparative Analysis of Gradient Boosting Algorithms / C. Bentéjac, A. Csörgő, G. Martínez-Muñoz // Artificial Intelligence Review. — 2021. — Vol. 54, № 3. — P. 1937–1967. — DOI: 10.1007/s10462-020-09896-5. — URL: https://dl.acm.org/doi/abs/10.1007/s10462-020-09896-5
9. Akiba T. Optuna: A Next-generation Hyperparameter Optimization Framework / T. Akiba, S. Sano, T. Yanase [et al.] // Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (Anchorage, August 4–8, 2019). — New York: ACM, 2019. — P. 2623–2631. — DOI: 10.1145/3292500.3330701. — URL: https://arxiv.org/abs/1907.10902.
10. Hastie T. The Elements of Statistical Learning: Data Mining, Inference, and Prediction / T. Hastie, R. Tibshirani, J. Friedman. — 2nd ed. — New York: Springer, 2009. — 745 p. — (Springer Series in Statistics).
11. Credit Card Fraud Detection [Электронный ресурс] / ULB Machine Learning Group // Kaggle. — 2018. — URL: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud (дата обращения: 15.12.2025).