DOI: 10.15514/ISPRAS-2026-38(5)-16

Глубокое обучение на табличных данных: от надежной оценки к передовым результатам на академических наборах задач

И.В. Рубачёв, *ORCID: 0000-0001-8023-8147 <irubachev@hse.ru>*

НИУ Высшая школа экономики,
101978, Россия, г. Москва, ул. Мясницкая, д. 20.

Аннотация. Надежный прогресс в глубоком обучении на табличных данных начинается с трудной задачи сравнения. Если сравнить простые полносвязные нейросетевые модели (MLP), специализированные архитектуры для табличных данных и методы градиентного бустинга над деревьями решений (GBDT) на академических наборах данных в едином и тщательном протоколе, то хорошо настроенные MLP-подобные модели зачастую трудно превзойти более продвинутыми методами, а GBDT остается сильным конкурентом и задает высокую планку для области. В этой работе мы используем такой строгий протокол оценки, чтобы продемонстрировать два механизма, которые улучшают табличные нейросетевые модели относительно сильных MLP-подобных методов и GBDT. Первый механизм связан с улучшенной методологией обучения: двухэтапным обучением с искажением входных данных, вспомогательными функциями потерь и более длительным обучением. Второй механизм связан с использованием непараметрической компоненты, которые во время предсказания используют представления и метки похожих объектов из обучающей выборки. На рассмотренных открытых академических наборах данных оба направления устойчиво улучшают качество тщательно настроенных параметрических нейросетей и часто достигают уровня качества GBDT или превосходят его. Эти результаты показывают, что глубокое обучение на табличных данных может быть конкурентоспособным с методами на основе решающих деревьев при строгой оценке относительно сильных простых моделей.

Ключевые слова: табличные данные; глубокое обучение; данные; предобучение; непараметрические компоненты; градиентный бустинг.

Для цитирования: Рубачёв И.В. Глубокое обучение на табличных данных: от надежной оценки к передовым результатам на академических наборах данных. Труды ИСП РАН, 2026, том 38, вып. 5, с. 287-304. DOI: 10.15514/ISPRAS-2026-38(5)-16

Tabular Deep Learning: From Reliable Evaluation to State-of-the-Art Results on Academic Benchmarks

I.V. Rubachev, *ORCID: 0000-0001-8023-8147 <irubachev@hse.ru>*

National Research University, Higher School of Economics,
20, Myasnitskaya st., Moscow, 101978, Russia.

Abstract. Reliable progress in tabular deep learning starts with a difficult comparison problem. When simple fully connected multilayer perceptrons (MLPs), specialized tabular architectures, and gradient-boosted decision trees are evaluated under a unified and thorough protocol on academic benchmarks, well-tuned MLP models are already hard to improve upon, while GBDT models remain strong competitors that set a high bar for the field. In this work, we use this demanding evaluation setting to demonstrate two mechanisms that improve tabular neural networks beyond strong MLP-based models and often bring them to the level of tree-based methods. The first one is an improved training methodology: two-stage training with input corruptions, auxiliary self-prediction objectives, and longer optimization schedules. The second one is retrieval: neural models equipped with a nonparametric component that, at prediction time, uses representations and labels of similar training objects. On the considered public academic benchmarks, both directions consistently improve tuned parametric neural networks. Moreover, some variants of the two approaches often reach or exceed GBDT quality. These results show that tabular deep learning can be competitive with tree-based methods when evaluated against strong simple models under a rigorous protocol.

Keywords: tabular data; deep learning; benchmarking; pretraining; retrieval; gradient boosting.

For citation: Rubachev I.V. Tabular Deep Learning: From Reliable Evaluation to State-of-the-Art Results on Academic Benchmarks. Trudy ISP RAN/Proc. ISP RAS, 2026, vol. 38, issue 5, pp. 287-304. DOI: 10.15514/ISPRAS-2026-38(5)-16

1. Введение

Табличные данные остаются одним из основных форматов прикладного машинного обучения. Во многих задачах – кредитном скоринге, страховании, электронной коммерции, рекламе, логистике и медицинской аналитике – объекты естественно представлены строками таблицы с числовыми, категориальными и бинарными признаками. Несмотря на успех глубокого обучения в компьютерном зрении и обработке естественного языка, в табличных задачах долгое время доминировали методы градиентного бустинга над деревьями решений (Gradient-boosted decision trees, GBDT), такие как XGBoost, LightGBM и CatBoost [1-3]. Эти методы остаются главной точкой отсчета и сильной альтернативой глубокому обучению на табличных данных благодаря сочетанию высокого качества, устойчивости и сравнительно умеренных вычислительных затрат.

В начале 2020-х годов интерес к глубокому обучению на табличных данных существенно вырос. Было предложено множество специализированных архитектур: SNN, NODE, TabNet, AutoInt, DCNv2, GrowNet, SAINT, NPT и другие [4-11]. Однако интерпретировать результаты этих работ было непросто. Разные статьи использовали разные наборы данных, разные процедуры предобработки и разные бюджеты настройки гиперпараметров. В такой ситуации заявленное преимущество новой архитектуры могло отражать не столько ее свойства, сколько недостаточно хорошую настройку моделей, использовавшихся для эмпирических сравнений, или отсутствие полноценного сравнения с методом GBDT.

Поэтому использование единого и надежного протокола оценки является необходимой отправной точкой для изучения табличного глубокого обучения. В такой постановке становится видно, что хорошо настроенные MLP-подобные модели зачастую трудно превзойти более продвинутыми методами, а GBDT остается сильным конкурентом и задает высокую планку для области. Этот результат важен не только как корректировка выводов о

ранних архитектурах, но и как основа для дальнейшего прогресса: новые методы должны улучшать уже сильные простые нейросетевые модели и сравниваться с хорошо настроенными методами GBDT.

Далее мы рассматриваем два подхода, которые позволяют табличному глубокому обучению достигать высокого качества на академических наборах данных в такой строгой постановке. Первый подход связан с улучшенной методологией обучения: двухэтапным обучением с искажением входных данных, вспомогательными функциями потерь, более длительным обучением и использованием целевой переменной на первом этапе. Второй подход связан с использованием непараметрической компоненты: во время предсказания модель использует представления и метки похожих объектов из обучающей выборки. Эти подходы устроены по-разному, но оба позволяют эффективнее использовать доступные обучающие данные.

На рассмотренных открытых академических наборах данных оба направления устойчиво улучшают качество тщательно настроенных параметрических нейросетей. Более того, некоторые варианты этих подходов часто достигают качества GBDT или превосходят его. Таким образом, строгая оценка не только показывает силу простых моделей и методов на основе решающих деревьев, но и позволяет выделить конкретные механизмы, благодаря которым глубокое обучение на табличных данных становится конкурентоспособным с GBDT в академической постановке.

2. Вклад

1. Мы формулируем строгую академическую постановку оценки табличного глубокого обучения. Она включает единые разбиения данных, согласованную предобработку, настройку гиперпараметров по валидационной выборке, раннюю остановку, усреднение по нескольким случайным инициализациям, ансамблирование и сравнение с хорошо настроенными методами GBDT. В такой постановке хорошо настроенные полносвязные нейросетевые модели, или многослойные перцептроны (Multilayer Perceptron, MLP), оказываются сильнее, чем следовало из части ранних работ, а GBDT остается главной точкой отсчета для оценки новых методов.

2. Мы показываем, что высокого качества на академических наборах данных можно достигать двумя различными способами. Первый способ: улучшенная методология обучения: двухэтапное обучение с искажением входных данных, вспомогательными функциями потерь и использованием целевой переменной. Второй способ: модели с непараметрической компонентой, которые при построении предсказания используют похожие объекты из обучающей выборки. Оба подхода улучшают сильные параметрические нейросетевые модели и часто достигают качества GBDT или превосходят его.

3. Обзор литературы

Методы градиентного бустинга для табличных данных. В задачах обучения с учителем на табличных данных долгое время доминируют ансамбли деревьев решений, в первую очередь методы градиентного бустинга. На практике чаще всего используются методы XGBoost [1], LightGBM [2] и CatBoost [3], которые хорошо работают со смешанными типами признаков, устойчивы к различным масштабам числовых признаков, эффективно используют нелинейные зависимости и обычно требуют умеренных вычислительных затрат. Поэтому при оценке нейросетевых методов для табличных данных GBDT является не вспомогательным методом сравнения, а одной из главных точек отсчета.

Нейросетевые модели для табличных данных. Рост интереса к табличному глубокому обучению привел к появлению большого числа специализированных архитектур. Часть моделей развивает обычные полносвязные сети: например, метод SNN использует иные активации [4], DCNv2 добавляет явное моделирование взаимодействий признаков [8], а

GrowNet строит бустинг из слабых нейросетевых моделей [9]. Другое направление пытается перенести идеи деревьев решений в дифференцируемую форму, как в методе NODE [5]. Отдельная группа методов использует механизмы внимания: AutoInt применяет механизм self-attention для моделирования взаимодействий признаков [7], TabNet сочетает последовательную обработку признаков с обучаемыми масками [6], а SAINT и NPT используют внимание не только между признаками, но и между объектами [10, 11].

Несмотря на разнообразие архитектур, сравнивать результаты ранних работ между собой было трудно. В разных статьях использовались разные наборы данных, разные разбиения, разные процедуры предобработки и разные бюджеты настройки гиперпараметров. Кроме того, простые MLP-подобные модели и методы GBDT не всегда настраивались достаточно тщательно. В результате улучшение новой архитектуры могло быть связано не только с ее устройством, но и с особенностями экспериментального протокола. Последующие независимые эмпирические исследования также показали, что методы на основе деревьев решений часто остаются сильнее нейросетевых моделей на типичных табличных задачах [12-14]. Это делает надежную и единую постановку оценки необходимым условием для содержательных выводов.

Предобучение и вспомогательные задачи. В компьютерном зрении и обработке естественного языка предобучение стало стандартным способом улучшения качества моделей [15, 16]. Для табличных данных эта идея также изучалась в нескольких работах. VIME использует восстановление маски и значений искаженных признаков [17]; SCARF применяет контрастивное обучение с искажением входных признаков [18]; SubTab разбивает признаки объекта на подмножества и обучает согласованные представления [19]. Предобучение также использовалось как часть более сложных табличных архитектур, например в TabNet, SAINT и NPT [6, 10, 11].

При этом для табличных данных предобучение имеет важную особенность. В отличие от задач с изображениями и текстами, обычно нет большого внешнего корпуса, на котором можно предварительно обучить модель. Поэтому первый этап обучения чаще всего выполняется на том же наборе данных, на котором затем решается целевая задача. В такой постановке предобучение ближе к методологии обучения с дополнительной регуляризацией: модель обучается на искаженных входных данных, решает вспомогательные задачи и затем дообучается на исходной задаче. В данной работе мы рассматриваем именно такую постановку и отдельно изучаем, как на первом этапе можно использовать целевую переменную.

Методы с непараметрической компонентой. Идея использовать похожие обучающие объекты при построении предсказания восходит к локальному обучению и методу ближайших соседей [20]. Современные модели развивают эту идею с помощью обучаемых представлений: сначала объект переводится в пространство, где близость должна быть более согласована с целевой задачей, затем для него находятся похожие объекты из обучающей выборки, и их признаки или метки используются при предсказании.

Такие подходы успешно применялись в других областях. В обработке языка механизмы поиска похожих объектов позволяют языковым моделям использовать внешние документы или ближайшие примеры из обучающего множества [21-23]. В компьютерном зрении похожие идеи используются для улучшения представлений и предсказаний с помощью найденных примеров [24]. Для табличных задач близкими по духу являются DNNR [25], Deep Kernel Learning [26], Attentive Neural Processes [27], а также модели с вниманием между объектами, такие как SAINT и NPT [10, 11]. Однако само наличие непараметрической компоненты не гарантирует улучшения качества: важны способ построения представлений, метрика близости, использование меток найденных объектов и вычислительная стоимость поиска.

4. ЭКСПЕРИМЕНТЫ

4.1 Описание протокола экспериментов

Мы рассматриваем задачи обучения с учителем на табличных данных. Набор данных задается как

$$D = \{(x_i, y_i)\}_{i=1}^{n},$$

где объект x_i содержит числовые и, при наличии, категориальные и бинарные признаки, а y_i является целевой переменной. В экспериментах встречаются бинарная классификация, многоклассовая классификация и регрессия. Для классификации у нейросетевых моделей минимизируется кросс-энтропия, для регрессии – среднеквадратичная ошибка. Качество измеряется метрикой, принятой для соответствующего набора данных: accuracy, ROC-AUC, RMSE или log-loss.

Для каждого набора данных используется фиксированное разбиение на обучающую, валидационную и тестовую части. Обучающая часть используется для подгонки параметров модели, валидационная – для ранней остановки и выбора гиперпараметров, тестовая – только для финальной оценки. Все сравнения внутри одной таблицы проводятся при одинаковых разбиениях данных и одинаковой процедуре выбора гиперпараметров.

Для нейросетевых моделей используется единая предобработка. По умолчанию числовые признаки преобразуются квантильным преобразованием из библиотеки Scikit-learn [28]; в отдельных случаях, когда такая предобработка ухудшает качество, используется стандартизация или исходные признаки. Категориальные признаки кодируются one-hot кодированием либо с помощью обучаемых векторных представлений в зависимости от модели. Для регрессионных задач целевая переменная стандартизуется на основе обучающей выборки. Для методов GBDT такая предобработка числовых признаков обычно не применяется, поскольку деревья решений нечувствительны к монотонным преобразованиям масштаба.

Гиперпараметры настраиваются отдельно для каждой пары “метод – набор данных” по валидационной выборке. В большинстве экспериментов используется библиотека Optuna [29] с алгоритмом Tree-Structured Parzen Estimator. После выбора гиперпараметров каждая нейросетевая модель обучается с несколькими случайными инициализациями; в основных таблицах сообщается среднее качество на тестовой выборке. Для получения ансамблей обученные модели разбиваются на несколько непересекающихся групп, и предсказания моделей внутри каждой группы усредняются. Такой протокол позволяет оценивать не только качество отдельных моделей, но и качество ансамблей, что особенно важно при сравнении с GBDT, поскольку градиентный бустинг сам является ансамблевым методом.

Если не указано иное для конкретной группы экспериментов, нейросетевые модели обучаются оптимизатором AdamW, без изменения скорости обучения по заранее заданному расписанию и без дополнительных аугментаций помимо явно описанных ниже. Обучение останавливается, если качество на валидационной выборке не улучшается в течение 16 эпох. Для большинства пар “метод – набор данных” используются 100 итераций настройки гиперпараметров; для наиболее крупных наборов и вычислительно тяжелых методов бюджет может быть меньше.

Для проведения экспериментов используются публичные академические наборы данных из литературы по табличному глубокому обучению: Gesture Phase (GE), Churn Modelling (CH), California Housing (CA), House 16H (HO), Adult (AD), Diamond (DI), Otto Group Products (OT), Higgs (HI), Facebook Comments (FB), Black Friday (BL), Weather (WE), Covertype (CO), Microsoft (MI), Helena (HE), Jannis (JA), ALOI (AL), Epsilon (EP), Year (YE) и Yahoo (YA) [30, 31, 32]. Источники и характеристики наборов California Housing, Adult, Helena, Jannis, Higgs, ALOI, Epsilon, Year, Covertype, Yahoo и Microsoft приведены в разделе 4.2 и приложении B.1

работы [31]; источники остальных наборов — в приложении A работы [32] и приложении C.1 работы [30]. Поскольку рассматриваемые методы имеют разные вычислительные ограничения и поддерживают разные типы задач, эксперименты организованы в несколько групп.

Строгая базовая постановка оценки и сравнения в разделе о простых моделях и GBDT следует работе [31], а эксперименты с ансамблями TabR в разделе о моделях с непараметрической компонентой – работе [30]. В каждой группе методы сравниваются на одном и том же подмножестве наборов данных и в одном экспериментальном протоколе. Метрики зависят от задачи и указываются в таблицах стрелками в заголовках столбцов (ROC-AUC, Accuracy – ↑, RMSE, Log-Loss – ↓).

4.2 Обзор экспериментальных результатов

В разделе 4.3 сравниваются MLP-подобные модели, специализированные табличные архитектуры и GBDT в едином протоколе. Этот шаг важен методологически: он показывает, что тщательно настроенный MLP уже является сильным методом сравнения, а GBDT остается существенным конкурентом. Следовательно, дальнейшие улучшения должны оцениваться именно относительно этих сильных методов.

Основная часть экспериментов посвящена улучшенной методологии обучения. В разделе 4.4 изучается двухэтапное обучение: на первом этапе модель обучается на том же наборе данных с искажением входных данных и вспомогательными функциями потерь, а затем дообучается на исходной задаче на неискаженных данных. Отдельно рассматриваются варианты, в которых целевая переменная используется уже на первом этапе. Эти эксперименты показывают, что качество табличных нейросетей можно заметно улучшить без привлечения внешних данных.

В разделе 4.5 дополнительно оцениваются модели с непараметрической компонентой. Такие модели при построении предсказания используют не только параметры нейросети, но и похожие объекты из обучающей выборки, включая их представления и метки. Эта часть экспериментов показывает, что использование обучающих объектов во время предсказания также может давать устойчивые улучшения относительно параметрических нейросетевых моделей.

4.3 Строгая точка отсчета: простые модели и GBDT

Сначала мы фиксируем точку отсчета для последующих сравнений. Цель этого шага состоит не в том, чтобы выбрать одну лучшую архитектуру, а в том, чтобы понять, насколько сильными оказываются простые MLP-подобные модели, специализированные табличные архитектуры и GBDT при одинаковом экспериментальном протоколе.

Табл. 1 показывает результаты сравнения нейросетевых моделей. Хорошо настроенный MLP оказывается сильным методом сравнения: многие специализированные архитектуры не дают устойчивого улучшения относительно него. ResNet-подобная архитектура улучшает MLP в среднем, а FT-Transformer показывает лучший средний ранг среди рассмотренных нейросетевых моделей. При этом отдельные архитектуры могут выигрывать на отдельных наборах данных, но это не превращается в стабильное превосходство по всему набору задач.

Далее мы сравниваем основные нейросетевые модели с GBDT. Для более честного сравнения используются ансамбли: предсказания нескольких нейросетевых моделей усредняются, поскольку градиентный бустинг сам является ансамблевым методом. Результаты приведены в табл. 2. Они показывают, что даже сильные нейросетевые модели не вытесняют GBDT полностью. На ряде задач, например California Housing, Adult и Yahoo, XGBoost или CatBoost остаются лучше. На других задачах нейросетевые модели, особенно FT-Transformer, достигают более высокого качества.

Из этого сравнения следует два вывода. Во-первых, простые MLP-подобные модели должны быть настроены тщательно, иначе сравнение со специализированными архитектурами становится малоинформативным. Во-вторых, дальнейшие улучшения табличного глубокого обучения должны оцениваться относительно сильных простых нейросетевых моделей и GBDT. Именно в такой постановке далее рассматриваются два подхода: двухэтапное обучение с вспомогательными функциями потерь и модели с непараметрической компонентой.

Табл. 1. Сравнение нейросетевых моделей в строгой академической постановке. Значения усреднены по 15 случайным инициализациям. Жирным выделены лучшие или статистически неотличимые от лучших результаты. Последний столбец содержит средний ранг по наборам данных. Адаптировано из [31].
Table 1. Comparison of neural network models under a rigorous academic evaluation protocol. Results are averaged over 15 random initializations. Bold indicates the best results or those statistically indistinguishable from the best. The last column reports the average rank across datasets. Adapted from [31].

Method	CA ↓	AD ↑	HE ↑	JA ↑	HI ↑	AL ↑	EP ↑	YE ↓	CO ↑	YA ↓	MI ↓	rank (std)
TabNet	0.510	0.850	0.378	0.723	0.719	0.954	0.8896	8.909	0.957	0.823	0.751	7.5 (2.0)
SNN	0.493	0.854	0.373	0.719	0.722	0.954	0.8975	8.895	0.961	0.761	0.751	6.4 (1.4)
AutoInt	0.474	**0.859**	0.372	0.721	0.725	0.945	0.8949	8.882	0.934	0.768	0.750	5.7 (2.3)
GrowNet	0.487	**0.857**	–	–	0.722	–	0.8970	8.827	–	0.765	0.751	5.7 (2.2)
MLP	0.499	0.852	0.383	0.719	0.723	0.954	0.8977	8.853	0.962	0.757	0.747	4.8 (1.9)
DCN2	0.484	0.853	0.385	0.716	0.723	0.955	0.8977	8.890	0.965	0.757	0.749	4.7 (2.0)
NODE	0.464	**0.858**	0.359	0.727	0.726	0.918	0.8958	**8.784**	0.958	**0.753**	**0.745**	3.9 (2.8)
ResNet	0.486	0.854	**0.396**	0.728	0.727	**0.963**	0.8969	8.846	0.964	0.757	0.748	3.3 (1.8)
FT-T	**0.459**	**0.859**	0.391	**0.732**	**0.729**	0.960	**0.8982**	8.855	**0.970**	0.756	0.746	1.8 (1.2)

Табл. 2. Сравнение ансамблей GBDT и основных нейросетевых моделей. Результаты показывают, что GBDT остается сильной точкой сравнения даже после тщательной настройки нейросетевых моделей. Адаптировано из [31].
Table 2. Comparison of GBDT ensembles and baseline neural network ensembles. The results show that GBDT remains a strong baseline even after careful tuning of neural network models. Adapted from [31].

Method	CA ↓	AD ↑	HE ↑	JA ↑	HI ↑	AL ↑	EP ↑	YE ↓	CO ↑	YA ↓	MI ↓
Default hyperparameters											
XGBoost	0.462	**0.874**	0.348	0.711	0.717	0.924	0.8799	9.192	0.964	0.761	0.751
CatBoost	**0.428**	0.873	0.386	0.724	0.728	0.948	0.8893	8.885	0.910	0.749	0.744
FT-Transformer	0.454	0.860	**0.395**	**0.734**	**0.731**	**0.966**	**0.8969**	**8.727**	**0.973**	**0.747**	**0.742**
Tuned hyperparameters											
XGBoost	0.431	0.872	0.377	0.724	0.728	–	0.8861	8.819	0.969	**0.732**	0.742
CatBoost	**0.423**	**0.874**	0.388	0.727	0.729	–	0.8898	8.837	0.968	0.740	**0.741**
ResNet	0.478	0.857	0.398	0.734	0.731	0.966	0.8976	8.770	0.967	0.751	0.745
FT-Transformer	0.448	0.860	**0.398**	**0.739**	**0.731**	**0.967**	**0.8984**	**8.751**	**0.973**	0.747	0.743

4.4 Двухэтапное обучение, аугментации и вспомогательные функции потерь

4.4.1 Постановка

В этой части мы рассматриваем двухэтапное обучение без внешних данных. Первый этап выполняется на том же наборе данных, на котором затем решается целевая задача. После первого этапа модель дообучается на исходной задаче на неискаженных данных. Такая постановка отличается от предобучения в компьютерном зрении и обработке языка, где часто используется большой внешний корпус. Здесь первый этап скорее является частью методологии обучения: модель обучается на искаженных входных данных, решает вспомогательные задачи и получает инициализацию, более подходящую для последующего дообучения.

Основные модели в этой части – MLP и MLP с векторными представлениями числовых признаков. В этих моделях каждый числовой признак сначала отображается в вектор, после чего представления всех признаков конкатенируются и подаются в MLP. Мы используем два варианта таких моделей: MLP-PLR с периодическими преобразованиями числовых признаков и MLP-T-LR с кусочно-линейными кодировками. Эти модели важны для оценки, потому что обычный MLP уже является сильным методом сравнения, а векторные представления числовых признаков дополнительно повышают качество MLP-подобных моделей.

Ключевой технический элемент первого этапа – искажение входных данных. Для объекта

$$x = (x^1, \dots, x^m)$$

строится искаженная версия $\hat{x}$. Часть признаков заменяется значениями из того же столбца обучающей выборки:

$$\hat{x}^j = \{x^j, b^j = 0, \tilde{x}^j, b^j = 1,$$

где b^j показывает, был ли искажен j-й признак, а $\tilde{x}^j$ выбирается из эмпирического распределения соответствующего признака. Такая процедура сохраняет допустимые значения отдельных признаков, но разрушает часть зависимостей между признаками. Поэтому она подходит как для аугментаций, так и для построения вспомогательных задач.

4.4.2 Функции потерь первого этапа

Мы сравниваем несколько функций потерь первого этапа. Пусть $f(\hat{x})$ – представление объекта, полученное основной частью модели, g – голова целевой задачи, а h – голова вспомогательной задачи.

Контрастивная функция потерь. В контрастивной постановке исходный объект и его искаженная версия считаются положительной парой, а остальные объекты мини-батча – отрицательными. Модель обучается сближать представления x и $\hat{x}$ относительно представлений других объектов.

Реконструкция. В задаче реконструкции модель должна восстановить исходный объект по искаженному:

$$\boldsymbol{L_{rec} = \ell_{rec}(h(f(\hat{x})), x)}.$$

Эта функция потерь заставляет представление сохранять информацию о признаках и зависимостях между ними.

Предсказание маски. В задаче предсказания маски модель должна определить, какие признаки были искажены:

$$\boldsymbol{L_{mask} = BCE(h(f(\hat{x})), b)}.$$

Эта задача требует от модели распознавать несогласованность между признаками и поэтому хорошо согласуется с табличной структурой данных.

Обучение с учителем на искаженных входах. Простейший способ использовать целевую переменную на первом этапе – обучать модель предсказывать y по искаженному объекту:

$$\boldsymbol{L_{sup} = \ell_y(g(f(\hat{x})), y)}.$$

После этого модель дообучается на исходной задаче на неискаженных данных.

Комбинированные функции потерь. Также рассматриваются комбинации обучения с учителем и вспомогательной задачи, например

$$\boldsymbol{L_{rec+sup} = L_{rec} + L_{sup}}.$$

В таких вариантах первый этап одновременно использует целевую переменную и сохраняет вспомогательную задачу, связанную со структурой входных данных.

4.4.3 Сравнение функций потерь без явного использования целевой переменной

Табл. 3 сравнивает контрастивную функцию потерь, реконструкцию и предсказание маски. Результаты показывают, что простые вспомогательные функции потерь часто не уступают контрастивному обучению и во многих случаях оказываются предпочтительнее. Для MLP предобучение почти всегда улучшает качество относительно обучения с нуля. Для MLP-PLR и MLP-T-LR улучшения также встречаются, но они менее однородны, поскольку сами модели с векторными представлениями числовых признаков уже являются сильными.

Табл. 3. Сравнение функций потерь первого этапа. Значения усреднены по 15 случайным инициализациям. Жирным выделены статистически значимые лучшие результаты внутри соответствующей модели и набора данных. Стрелки показывают направление лучшего качества.
Table 3. Comparison of first-stage loss functions. Results are averaged over 15 random initializations. Bold indicates statistically significant best results for each model and dataset. Arrows indicate whether higher or lower values are better.

Method	GE ↑	CH ↑	CA ↓	HO ↓	OT ↓	HI ↑	FB ↓	AD ↑	WE ↓	CO ↑	MI ↓
MLP											
no pretraining	0.635	0.849	0.506	3.156	0.479	0.801	5.737	0.908	1.909	0.963	0.749
contrastive	0.672	**0.855**	0.455	**3.056**	0.469	**0.813**	5.697	0.910	1.881	0.960	0.748
rec	0.662	0.853	**0.445**	**3.044**	**0.466**	0.805	**5.641**	**0.910**	**1.875**	**0.965**	**0.746**
mask	**0.691**	**0.857**	0.454	3.113	0.472	**0.814**	**5.681**	**0.912**	1.883	0.964	0.748
MLP-PLR											
no pretraining	0.668	**0.858**	0.469	**3.008**	0.483	0.809	**5.608**	0.926	1.890	0.969	0.746
rec	0.667	0.852	**0.439**	**3.031**	**0.472**	0.808	**5.571**	**0.926**	**1.877**	**0.971**	**0.745**
mask	**0.685**	**0.863**	**0.434**	**3.007**	0.477	**0.818**	**5.586**	**0.927**	1.911	**0.970**	0.748
MLP-T-LR											
no pretraining	0.634	**0.866**	0.444	3.113	0.482	0.805	5.520	0.925	1.897	0.968	0.749
rec	**0.652**	0.857	**0.424**	3.109	**0.472**	0.808	**5.363**	0.924	**1.861**	**0.969**	**0.746**
mask	**0.654**	**0.868**	**0.424**	**3.045**	**0.472**	**0.818**	5.544	**0.926**	1.916	**0.969**	0.748

Главный вывод из таблицы состоит в том, что для табличных данных не обязательно использовать контрастивную функцию потерь. Реконструкция и предсказание маски проще устроены, не требуют второй версии каждого объекта в качестве отдельного представления для контрастивного сравнения и не используют отрицательные примеры из мини-батча. При этом они дают сопоставимые или лучшие результаты.

4.4.4 Использование целевой переменной на первом этапе

Далее мы рассматриваем методы, в которых целевая переменная используется уже на первом этапе. Это естественно для табличной постановки, поскольку первый этап выполняется на том же размеченном наборе данных. Использование целевой переменной реализуется двумя способами.

Первый способ – добавить обучение с учителем на искаженных входах. В этом случае модель на первом этапе решает целевую задачу по $\hat{x}$, а затем дообучается на исходных объектах x. Второй способ – изменить вспомогательную задачу так, чтобы она явно зависела от y. Например, голова реконструкции или предсказания маски получает не только представление $f(\hat{x})$, но и представление целевой переменной:

$$\hat{p} = h([f(\hat{x}), e(y)]),$$

где $e(y)$ – one-hot кодирование для классификации или масштабированное значение для регрессии.

Дополнительно можно изменить схему искажения входа. Вместо выбора замены из безусловного распределения $p(x^j)$ значение признака выбирается из условного распределения $p(x^j \mid \tilde{y})$, где $\tilde{y}$ отличается от исходной целевой переменной. Для регрессии целевая переменная предварительно дискретизуется. Интуитивно такая схема делает искажения более связанными с целевой задачей.

Результаты приведены в табл. 4. Обучение с учителем на искаженных входах оказывается сильным вариантом для MLP. Комбинация реконструкции и обучения с учителем наиболее стабильна среди вариантов с реконструкцией. Для предсказания маски наиболее полезным оказывается вариант, в котором вспомогательная задача явно учитывает целевую переменную.

Эти результаты показывают, что целевую переменную полезно использовать не только на этапе финального дообучения, но и на первом этапе. При этом нет одной универсальной функции потерь, которая была бы лучшей во всех случаях. На практике имеет смысл рассматривать несколько простых вариантов: реконструкцию, предсказание маски, обучение с учителем на искаженных входах, а затем варианты с учетом целевой переменной для наиболее подходящей вспомогательной задачи.

4.4.5 Сравнение с GBDT

Табл. 5 сравнивает ансамбли нейросетевых моделей после двухэтапного обучения с ансамблями GBDT. Результаты показывают, что двухэтапное обучение существенно улучшает MLP. Более того, MLP с такими функциями потерь часто достигает качества GBDT или превосходит его на отдельных наборах данных. При добавлении векторных представлений числовых признаков качество улучшается дальше: MLP-PLR и MLP-T-LR с двухэтапным обучением оказываются сильными методами в этой академической постановке.

Этот результат демонстрирует, что при строгой оценке табличные нейросетевые модели могут становиться конкурентоспособными с GBDT не только за счет изменения архитектуры, но и за счет изменения методологии обучения.

4.4.6 Вычислительные затраты

Двухэтапное обучение требует больше времени, чем обычное обучение с нуля, поскольку первый этап добавляет значительное число шагов оптимизации. Чтобы оценить эту цену, мы сравниваем обучение MLP с нуля и двухэтапное обучение с функцией потерь rec + sup при разных ограничениях на число итераций первого этапа. Результаты приведены в табл. 6.

Из таблицы видно, что двухэтапное обучение часто требует существенно больше времени, особенно на малых наборах данных, где обычное обучение MLP занимает секунды. При этом абсолютные значения времени остаются умеренными для большинства рассмотренных

академических задач. Увеличение числа итераций первого этапа часто дополнительно улучшает качество, поэтому на практике возникает обычный компромисс между качеством и вычислительными затратами. Важная практическая деталь состоит в том, что первый этап можно выполнить один раз, а затем строить ансамбль за счет нескольких независимых дообучений из одного предобученного состояния.

4.5 Модели с непараметрической компонентой

После экспериментов с двухэтапным обучением мы дополнительно рассматриваем другой способ улучшить табличные нейросетевые модели: использовать похожие обучающие объекты непосредственно при построении предсказания. Такая модель остается нейросетевой, но содержит непараметрическую компоненту: для целевого объекта она находит несколько близких объектов из обучающей выборки и использует информацию о них, включая их метки. Описание TabR и соответствующие экспериментальные результаты в этом разделе следуют работе [30].

В качестве основной модели этого типа мы используем TabR [30]. Это MLP-подобная модель с одним kNN-подобным модулем внутри. Сначала объект и кандидаты из обучающей выборки переводятся в обучаемое пространство представлений. Затем для целевого объекта находятся ближайшие кандидаты, а их метки и различия между представлениями используются для уточнения представления целевого объекта перед финальным предсказанием. В экспериментах также используется простая конфигурация TabR-S, в которой архитектура специально ограничена (используется меньше слоев и не используются векторные представления числовых признаков), чтобы проверить вклад самой непараметрической компоненты.

Табл. 4. Варианты двухэтапного обучения с использованием целевой переменной. Обозначение «+ target» означает учет целевой переменной во вспомогательной задаче, «+ sup» – дополнительную голову целевой задачи на искаженных входах.
Table 4. Variants of two-stage training that use the target variable. "+ target" denotes inclusion of the target variable in the auxiliary task; "+ sup" denotes an additional supervised prediction head applied to corrupted inputs.

Method	GE ↑	CH ↑	CA ↓	HO ↓	OT ↓	HI ↑	FB ↓	AD ↑	WE ↓	CO ↑	MI ↓	Avg. Rank
MLP												
no pretraining	0.635	0.849	0.506	3.156	0.479	0.801	5.737	0.908	1.909	0.963	0.749	5.5 ± 1.4
mask	0.691	0.857	0.454	3.113	0.472	0.814	5.681	0.912	1.883	0.964	0.748	3.8 ± 1.4
rec	0.662	0.853	0.445	**3.044**	0.466	0.805	5.641	0.910	1.875	0.965	0.746	3.6 ± 1.5
sup	0.693	0.856	0.441	3.077	**0.459**	0.814	5.689	0.914	1.883	0.968	0.748	3.0 ± 1.0
mask + target	0.683	0.857	0.434	**3.056**	0.468	**0.819**	5.633	0.914	1.876	0.965	0.748	2.9 ± 1.3
rec + target	0.659	0.853	0.454	**3.044**	0.463	0.806	5.636	0.909	1.884	0.965	**0.745**	3.7 ± 1.9
mask + sup	0.693	0.857	0.436	3.099	**0.458**	0.817	5.685	0.915	1.873	0.967	0.748	2.7 ± 1.2
rec + sup	0.684	0.854	0.436	**3.012**	**0.456**	0.815	5.672	0.911	**1.862**	0.967	0.747	2.6 ± 1.5
MLP-PLR												
no pretraining	0.668	0.858	0.469	**3.008**	0.483	0.809	5.608	0.926	1.890	0.969	0.746	3.5 ± 1.7
mask	0.685	**0.863**	0.434	**3.007**	0.477	0.818	5.586	0.927	1.911	0.970	0.748	2.8 ± 1.7
rec	0.667	0.852	0.439	**3.031**	0.472	0.808	5.571	0.926	1.877	**0.971**	**0.745**	2.6 ± 1.2
sup	**0.710**	0.859	0.433	3.136	0.479	0.811	5.521	0.924	1.873	**0.971**	0.748	2.5 ± 1.2
mask + target	0.694	0.862	**0.425**	**3.023**	0.474	**0.821**	5.537	**0.929**	1.911	0.969	0.749	2.5 ± 1.9
rec + target	0.688	0.860	0.445	**3.064**	0.475	0.812	5.507	0.927	1.887	**0.971**	0.748	2.7 ± 1.3
mask + sup	**0.711**	**0.866**	0.441	3.129	0.480	0.813	5.480	0.925	1.875	0.969	**0.745**	2.5 ± 1.4
rec + sup	**0.709**	0.858	0.433	**3.059**	0.465	0.807	5.571	0.927	**1.865**	**0.971**	**0.745**	1.9 ± 1.2

Method	GE ↑	CH ↑	CA ↓	HO ↓	OT ↓	HI ↑	FB ↓	AD ↑	WE ↓	CO ↑	MI ↓	Avg. Rank
MLP-T-LR												
no pretraining	0.634	**0.866**	0.444	3.113	0.482	0.805	5.520	0.925	1.897	0.968	0.749	3.9 ± 1.7
mask	0.654	**0.868**	**0.424**	**3.045**	0.472	0.818	5.544	0.926	1.916	0.969	0.748	2.8 ± 1.7
rec	0.652	0.857	**0.424**	3.109	0.472	0.808	**5.363**	0.924	**1.861**	0.969	**0.746**	2.5 ± 1.4
sup	0.682	0.860	0.430	3.135	0.471	0.807	5.525	0.927	1.893	**0.971**	0.747	2.8 ± 1.5
mask + target	0.649	**0.865**	**0.421**	**3.058**	0.474	**0.820**	5.644	**0.929**	1.924	0.969	0.749	2.8 ± 2.1
rec + target	0.668	**0.864**	0.440	3.113	0.473	0.806	5.493	0.927	**1.862**	0.969	**0.746**	2.5 ± 1.4
mask + sup	0.676	0.858	0.429	3.199	0.468	0.814	5.510	0.926	1.869	**0.971**	0.748	2.5 ± 0.8
rec + sup	0.678	**0.865**	0.437	3.112	0.462	0.807	5.516	0.927	**1.862**	**0.970**	0.748	2.4 ± 1.2

Табл. 5. Сравнение ансамблей моделей с двухэтапным обучением и ансамблей GBDT. Результаты относятся к данной группе экспериментов; корректно сравнивать методы внутри таблицы.
Table 5. Comparison of ensembles of models trained in two stages with GBDT ensembles. The results pertain to this group of experiments; comparisons should be made within the table.

Method	GE ↑	CH ↑	CA ↓	HO ↓	OT ↓	HI ↑	FB ↓	AD ↑	WE ↓	CO ↑	MI ↓	Avg. Rank
CatBoost	0.692	**0.864**	0.430	3.093	0.450	0.807	5.226	0.928	1.801	0.967	**0.741**	2.6 ± 1.4
XGBoost	0.683	0.860	0.434	3.152	0.454	0.805	5.338	0.927	**1.782**	0.969	**0.742**	3.0 ± 1.5
MLP												
no pretraining	0.656	0.852	0.482	3.055	0.467	0.805	5.666	0.910	1.850	0.968	0.747	4.8 ± 1.1
mask + target	0.709	0.860	0.414	2.949	0.457	**0.828**	5.551	0.916	1.809	0.969	0.746	2.8 ± 1.2
rec + sup	0.709	0.859	0.419	2.951	**0.442**	0.817	5.531	0.913	1.801	0.973	0.745	2.5 ± 1.2
MLP-P-LR												
no pretraining	0.695	**0.864**	0.454	2.953	0.470	0.814	5.324	0.928	1.835	**0.974**	0.744	2.6 ± 1.2
mask + target	**0.719**	**0.866**	**0.407**	2.952	0.458	**0.828**	5.373	**0.930**	1.849	0.973	0.745	2.1 ± 1.2
rec + sup	**0.737**	0.862	0.424	2.964	0.449	0.811	**5.124**	**0.929**	1.813	**0.974**	0.744	2.0 ± 1.0
MLP-T-LR												
no pretraining	0.662	**0.868**	0.437	3.028	0.472	0.808	5.424	0.927	1.850	0.972	0.747	3.7 ± 1.1
mask + target	0.673	**0.868**	**0.410**	**2.894**	0.460	**0.827**	5.458	**0.930**	1.849	0.972	0.746	2.4 ± 1.4
rec + sup	0.705	**0.866**	0.425	3.057	**0.444**	0.814	5.422	0.927	1.811	**0.974**	0.746	2.5 ± 1.1

Табл. 6. Сравнение времени обучения MLP с нуля и двухэтапного обучения с функцией потерь rec + sup. Для каждого ограничения на число итераций первого этапа сообщается качество и среднее время обучения одной модели в секундах на GPU A100.
Table 6. Comparison of training time for MLP trained from scratch and two-stage training with the rec + sup loss. For each limit on the number of first-stage iterations, predictive performance and the average training time per model in seconds on an A100 GPU are reported.

Method	GE ↑	CH ↑	CA ↓	HO ↓	OT ↓	HI ↑	FB ↓	AD ↑	WE ↓	CO ↑	MI ↓
no pretraining	0.635	0.849	0.506	3.156	0.479	0.801	5.737	0.908	1.909	0.963	0.749
	29s	10s	25s	26s	38s	29s	159s	12s	57s	352s	211s
rec + sup	50k	0.679	0.857	0.441	3.064	0.462	0.813	5.650	0.910	1.879	0.966
	327s	227s	280s	277s	355s	256s	624s	403s	242s	593s	292s
rec + sup	100k	0.684	0.854	0.436	3.012	0.456	0.815	5.672	0.911	1.862	0.967
	661s	312s	533s	455s	570s	526s	740s	759s	439s	649s	472s
rec + sup	150k	0.692	0.859	0.435	3.012	0.456	0.816	5.629	0.910	1.866	0.968
	891s	338s	758s	509s	712s	862s	916s	1069s	663s	927s	665s

Опишем основной модуль поиска похожих объектов в TabR более формально. Пусть $\tilde{x} = E(x)$ – представление целевого объекта, а $\tilde{x}_i = E(x_i)$ – представления кандидатов из обучающей выборки. Модель строит ключи

$$k = W_K(\tilde{x}), \qquad k_i = W_K(\tilde{x}_i)$$

и измеряет близость через отрицательное евклидово расстояние

$$S(x, x_i) = -\| k - k_i \|^2.$$

После выбора m ближайших объектов их вклад задается выражением

$$V(x, x_i, y_i) = W_Y(y_i) + T(k - k_i),$$

где W_Y кодирует метку найденного объекта, а T – небольшая нейросетевая поправка, зависящая от различия представлений. Итоговый сигнал непараметрической компоненты получается взвешенным усреднением таких вкладов по найденным соседям и затем добавляется к представлению целевого объекта.

Табл. 7 сравнивает TabR с параметрическими нейросетевыми моделями и несколькими близкими подходами: метод k-ближайших соседей (k-Nearest Neighbors, kNN), метод дифференциальной регрессии ближайших соседей (Differential Nearest Neighbors Regression, DNNR), метод глубокого обучения на основе ядер (Deep Kernel Learning, DKL), нейронный процесс с механизмом внимания (Attentive Neural Process, ANP), нейросетевая модель на базе модели трансформер с непараметрической компонентой (Self-Attention and Intersample Attention Transformer, SAINT) и непараметрический трансформер (Non Parametric Transformer, NPT). Результаты показывают, что само наличие непараметрической компоненты не гарантирует улучшения качества: более ранние подходы с поиском похожих объектов и модели с вниманием между объектами часто дают качество, близкое к хорошо настроенным параметрическим моделям. TabR, напротив, во многих случаях превосходит MLP и MLP-PLR и показывает сильный средний ранг на этой группе наборов данных.

Табл. 7. Сравнение методов с непараметрической компонентой и параметрических нейросетевых моделей на академических наборах. Значения усреднены по 15 случайным инициализациям. Последний столбец содержит средний ранг. Адаптировано из [30].
Table 7. Comparison of methods with a nonparametric component and parametric neural network models on academic datasets. Results are averaged over 15 random initializations. The last column reports the average rank. Adapted from [30].

Method	CH ↑	CA ↓	HO ↓	AD ↑	DI ↓	OT ↑	HI ↑	BL ↓	WE ↓	CO ↑	MI ↓	Avg. Rank
kNN	0.837	0.588	3.744	0.834	0.256	0.774	0.665	0.712	2.296	0.927	0.764	6.0 ± 1.7
DNNR [25]	–	0.430	3.210	–	0.145	–	–	0.704	1.913	–	0.765	4.8 ± 1.9
DKL [26]	–	0.521	3.423	–	0.147	–	–	0.699	–	–	–	6.2 ± 0.5
ANP [27]	–	0.472	3.162	–	0.140	–	–	0.705	1.902	–	–	4.6 ± 2.5
SAINT [10]	**0.860**	0.468	3.242	0.860	0.137	0.812	0.724	0.693	1.933	0.964	0.763	3.8 ± 1.5
NPT [11]	0.858	0.474	3.175	0.853	0.138	0.815	0.721	0.692	1.947	0.966	0.753	3.6 ± 1.0
MLP	0.854	0.499	3.112	0.853	0.140	0.816	0.719	0.697	1.905	0.963	0.748	3.7 ± 1.3
MLP-PLR	0.860	0.476	**3.056**	**0.870**	0.134	0.819	**0.729**	0.687	1.860	0.970	**0.744**	2.0 ± 1.0
TabRsimple	0.860	**0.403**	**3.067**	0.865	**0.133**	0.818	0.722	0.690	1.747	0.973	0.750	1.9 ± 0.7
TabR	**0.862**	**0.400**	3.105	**0.870**	**0.133**	**0.825**	**0.729**	**0.676**	**1.690**	**0.976**	0.750	1.3 ± 0.6

Табл. 8 сравнивает ансамбли TabR с ансамблями GBDT. На этой академической коллекции TabR дает заметные улучшения относительно настроенных GBDT на нескольких наборах

данных и остается конкурентоспособным на большинстве остальных. При этом GBDT по-прежнему является сильным и часто более дешевым методом. Поэтому этот результат следует интерпретировать как дополнительное свидетельство того, что непараметрическая компонента может улучшать табличные нейросетевые модели в строгой академической постановке, но не как замену сравнению с GBDT.

4.5.1 Набор задач «Why Trees»

Дополнительно мы проверяем модели с непараметрической компонентой на наборе задач из работы [12]. Этот набор задач важен для рассматриваемого вопроса, поскольку изначально использовался как пример академического набора задач, на котором методы на основе деревьев решений систематически превосходят нейросетевые модели. В нашей оценке используются задачи регрессии и классификации среднего размера ($\leq 50K$ объектов), а сравнение выполняется относительно XGBoost [1].

Результаты на рис. 1 показывают последовательное улучшение табличных нейросетевых моделей в этой постановке. Обычный MLP и FT-Transformer остаются близки к XGBoost или уступают ему, MLP-PLR улучшает положение за счет векторных представлений числовых признаков, а TabR и ModernNCA [33] дополнительно улучшают среднее качество за счет использования похожих объектов из обучающей выборки. Это подтверждает основной вывод данного раздела: аккуратно спроектированная непараметрическая компонента может быть полезной не только на отдельной коллекции наборов данных, но и на специально подобранном «GBDT-дружественном» академическом наборе данных.

В совокупности результаты по двухэтапному обучению и по моделям с непараметрической компонентой показывают, что улучшения табличного глубокого обучения в академической постановке возникают не только из-за усложнения архитектуры.

Табл. 8. Сравнение ансамблей TabR с ансамблями GBDT на академической коллекции. Результаты показывают, что аккуратно спроектированная непараметрическая компонента может улучшать качество относительно GBDT на ряде задач. Адаптировано из [30].
Table 8. Comparison of TabR ensembles and GBDT ensembles on the academic benchmark collection. The results show that a carefully designed nonparametric component can improve performance over GBDT on several tasks. Adapted from [30].

Method	CH ↑	CA ↓	HO ↓	AD ↑	DI ↓	OT ↑	HI ↑	BL ↓	WE ↓	CO ↑	MI ↓	Avg. Rank
Tuned hyperparameters												
[0.1cm] XGBoost	0.861	0.432	3.164	**0.872**	0.136	**0.832**	0.726	0.680	1.769	0.971	0.741	2.5 ± 0.9
CatBoost	0.859	0.426	3.106	**0.872**	0.133	0.827	0.727	0.681	1.773	0.969	**0.741**	2.5 ± 1.1
LightGBM	0.860	0.434	3.167	**0.872**	0.136	**0.832**	0.726	0.679	1.761	0.971	0.741	2.4 ± 0.9
TabR	**0.865**	**0.391**	**3.025**	**0.872**	**0.131**	**0.831**	**0.733**	**0.674**	**1.661**	**0.977**	0.748	1.3 ± 0.9
Default hyperparameters												
XGBoost	0.856	0.471	3.368	0.871	0.143	0.817	0.716	0.683	1.920	0.966	0.750	3.4 ± 0.9
CatBoost	0.861	0.432	3.108	**0.874**	0.132	0.822	**0.726**	0.684	1.886	0.924	**0.744**	2.1 ± 0.8
LightGBM	0.856	0.449	3.222	0.869	0.137	**0.826**	0.720	0.681	1.817	0.899	0.744	2.5 ± 0.9
TabR-S	**0.864**	**0.398**	**2.971**	0.859	**0.131**	0.824	0.724	0.688	**1.721**	**0.974**	0.752	2.0 ± 1.3

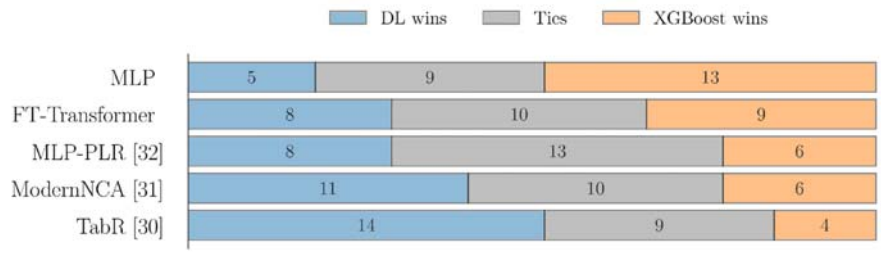

Рис. 1. Сравнение табличных нейросетевых моделей с XGBoost на задачах регрессии и классификации среднего размера ($\leq 50K$ объектов) из набора задач [12].
Метки на рисунке ссылаются на MLP-PLR [34], TabR [30] и ModernNCA [33].
Значения показывают среднее относительное положение методов относительно XGBoost: чем дальше метод расположен влево, тем чаще и сильнее он превосходит XGBoost на рассматриваемых задачах. Результаты для TabR адаптированы из [30].
Fig. 1. Comparison of tabular neural network models with XGBoost on medium-sized regression and classification tasks ($\leq 50K$ instances) from the benchmark in [12]. The labels refer to MLP-PLR [34], TabR [30], and ModernNCA [33]. Values indicate the average relative standing of the methods against XGBoost: the farther to the left a method appears, the more often and by a larger margin it outperforms XGBoost on these tasks. TabR results are adapted from [30].

Первый подход улучшает процесс обучения и качество представлений за счет искажений входа, вспомогательных задач и использования целевой переменной. Второй подход меняет форму предсказания, добавляя к параметрической модели информацию о похожих объектах из обучающей выборки.

5. Заключение

В этой работе мы рассмотрели глубокое обучение на табличных данных в строгой академической постановке оценки. Единые разбиения данных, согласованная предобработка, настройка гиперпараметров по валидационной выборке, ранняя остановка, несколько случайных инициализаций и сравнение с хорошо настроенными методами GBDT существенно меняют интерпретацию результатов. В такой постановке простые MLP-подобные модели оказываются сильными методами сравнения, а градиентный бустинг над деревьями решений остается важной точкой отсчета для новых подходов.

На этом фоне мы рассмотрели два направления, которые дают устойчивые улучшения на открытых академических наборах данных. Первое направление связано с методологией обучения. Двухэтапное обучение на том же наборе данных, искажение входных признаков, вспомогательные функции потерь и использование целевой переменной на первом этапе позволяют заметно улучшать качество MLP-подобных моделей. Особенно сильные результаты достигаются при сочетании такого обучения с векторными представлениями числовых признаков.

Второе направление связано с использованием непараметрической компоненты. Такие модели строят предсказание не только по параметрам нейросети, но и с учетом похожих объектов из обучающей выборки, включая их представления и метки. В рассмотренных экспериментах этот подход также улучшает качество относительно сильных параметрических нейросетевых моделей и часто позволяет достигать качества GBDT или превосходить его.

Полученные результаты показывают, что глубокое обучение на табличных данных может быть конкурентоспособным с методами на основе решающих деревьев при строгой оценке относительно сильных простых моделей. При этом выводы данной работы относятся к

рассмотренным академическими наборами задач. Перенос таких результатов в более прикладные условия требует отдельной оценки с учетом временной динамики данных, процесса построения признаков и вычислительных затрат на обучение и настройку моделей.

Список литературы / References

[1]. Chen T., Guestrin C. XGBoost: A scalable tree boosting system. SIGKDD, 2016.
[2]. Ke G., Meng Q., Finley T., Wang T., Chen W., Ma W., Ye Q., Liu T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 2017.
[3]. Prokhorenkova L., Gusev G., Vorobev A., Dorogush A.V., Gulin A. CatBoost: Unbiased boosting with categorical features. NeurIPS, 2018.
[4]. Klambauer G., Unterthiner T., Mayr A., Hochreiter S. Self-normalizing neural networks. NIPS, 2017.
[5]. Popov S., Morozov S., Babenko A. Neural Oblivious Decision Ensembles for Deep Learning on Tabular Data. International Conference on Learning Representations (ICLR), 2020. URL: https://openreview.net/forum?id=r1eiu2VtwH.
[6]. Arik S.Ö., Pfister T. TabNet: Attentive Interpretable Tabular Learning. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, vol. 35, no. 8, pp. 6679–6687. DOI: 10.1609/aaai.v35i8.16826.
[7]. Song W., Shi C., Xiao Z., Duan Z., Xu Y., Zhang M., Tang J. Autoint: Automatic feature interaction learning via self-attentive neural networks. Proceedings of the 28th ACM international conference on information and knowledge management, 2019.
[8]. Wang R., Shivanna R., Cheng D.Z., Jain S., Lin D., Hong L., Chi E.H. DCN V2: Improved Deep & Cross Network and Practical Lessons for Web-scale Learning to Rank Systems. Proceedings of the Web Conference 2021, 2021, pp. 1785–1797. DOI: 10.1145/3442381.3450078.
[9]. Badirli S., Liu X., Xing Z., Bhowmik A., Doan K., Keerthi S.S. Gradient Boosting Neural Networks: GrowNet. arXiv preprint arXiv:2002.07971, 2020. DOI: 10.48550/arXiv.2002.07971.
[10]. Somepalli G., Goldblum M., Schwarzschild A., Bruss C.B., Goldstein T. SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training. arXiv preprint arXiv:2106.01342, 2021. DOI: 10.48550/arXiv.2106.01342.
[11]. Kossen J., Band N., Lyle C., Gomez A.N., Rainforth T., Gal Y. Self-attention between datapoints: Going beyond individual input-output pairs in deep learning. NeurIPS. 2021.
[12]. Grinsztajn L., Oyallon E., Varoquaux G. Why do tree-based models still outperform deep learning on typical tabular data? NeurIPS, the "datasets and benchmarks" track. 2022.
[13]. Shwartz-Ziv R., Armon A. Tabular data: Deep learning is not all you need.” Information Fusion, 2022.
[14]. Borisov V., Leemann T., Seßler K., Haug J., Pawelczyk M., Kasneci G. Deep neural networks and tabular data: A survey. IEEE Transactions on Neural Networks and Learning Systems, 2024.
[15]. Devlin J., Chang M.-W., Lee K., Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: Human language technologies, vol. 1, 2019.
[16]. He K., Chen X., Xie S., Li Y., Dollár P., Girshick R. Masked autoencoders are scalable vision learners. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022.
[17]. Yoon J., Zhang Y., Jordon J., van der Schaar M. VIME: Extending the success of self- and semi-supervised learning to tabular domain. NeurIPS, 2020.
[18]. Bahri D., Jiang H., Tay Y., Metzler D. SCARF: Self-supervised contrastive learning using random feature corruption. ICLR, 2021.
[19]. Ucar T., Hajiramezanali E., Edwards L. SubTab: Subsetting features of tabular data for self-supervised representation learning. Advances in Neural Information Processing Systems, 2021.
[20]. Bottou L., Vapnik V. Local learning algorithms. Neural Computation, 1992.
[21]. Khandelwal U., Levy O., Jurafsky D., Zettlemoyer L., Lewis M. Generalization through memorization: Nearest neighbor language models. ICLR, 2020.
[22]. Lewis P.S.H. et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. NeurIPS, 2020.
[23]. Borgeaud S. et al. Improving language models by retrieving from trillions of tokens. ICML, 2022.
[24]. Long A., Yin W., Ajanthan T., Nguyen V., Purkait P., Garg R., Blair A., Shen C., van den Hengel A. Retrieval augmented classification for long-tail visual recognition. CVPR, 2022.
[25]. Nader Y., Sixt L., Landgraf T. DNNR: Differential nearest neighbors regression. ICML, 2022.
[26]. Wilson A.G., Hu Z., Salakhutdinov R., Xing E.P. Deep kernel learning. AISTATS, 2016.

[27]. Kim H., Mnih A., Schwarz J., Garnelo M., Eslami S.M.A., Rosenbaum D., Vinyals O., Teh Y.W. Attentive neural processes. ICLR, 2019.
[28]. Pedregosa F. et al. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 2011.
[29]. Akiba T., Sano S., Yanase T., Ohta T., Koyama M. Optuna: A next-generation hyperparameter optimization framework. KDD, 2019.
[30]. Gorishniy Y., Rubachev I., Kartashev N., Shlenskii D., Kotelnikov A., Babenko A. TABR: Tabular deep learning meets nearest neighbors in 2023. ICLR, 2024.
[31]. Gorishniy Y., Rubachev I., Khrulkov V., Babenko A. Revisiting Deep Learning Models for Tabular Data. NeurIPS, 2021. URL: https://arxiv.org/abs/2106.11959.
[32]. Rubachev I., Alekberov A., Gorishniy Y., Babenko A. Revisiting Pretraining Objectives for Tabular Deep Learning. arXiv preprint arXiv:2207.03208, 2022. DOI: 10.48550/arXiv.2207.03208.
[33]. Ye H.-J., Yin H.-H., Zhan D.-C., Chao W.-L. Revisiting nearest neighbor for tabular data: A deep tabular baseline two decades later. ICLR, 2025.
[34]. Gorishniy Y., Rubachev I., Babenko A. On embeddings for numerical features in tabular deep learning. NeurIPS, 2022.

Информация об авторах / Information about authors

Иван Викторович РУБАЧЁВ – аспирант Национального исследовательского университета «Высшая школа экономики». Сфера научных интересов: машинное обучение на табличных данных, глубокое обучение, тестирование моделей.

Ivan Viktorovich RUBACHEV – postgraduate student at HSE University. Research interests: machine learning on tabular data, deep learning, model benchmarking.