DOI: 10.15514/ISPRAS-2026-38(6)-27

CC BY

Модели прозрачного ящика для графовых данных: таксономия, иллюстративный анализ и перспективы формирующегося направления

[1,2] *М.В. Соловьев, ORCID: 0009-0009-2243-273X <m.solovev@phystech.edu>*
[1,2] *М.Д. Дробышевский, ORCID: 0000-0002-1639-9154 <drobyshevsky@ispras.ru>*
[1,3] *А.Д. Моисейкин, ORCID: 0009-0009-4546-5681 <s02230192@gse.cs.msu.ru>*
[2,4] *Е.Ю. Полтавцев, ORCID: 0009-0008-6600-887X <poltavtsev.eu@phystech.edu>*
[2,4] *Е.Н. Потурило, ORCID: 0009-0007-5725-4801 <poturilo.en@phystech.edu>*
[2,4] *И.С. Халько, ORCID: 0009-0002-7373-4657 <khalko.is@phystech.edu>*

[1] *Институт системного программирования им. В.П. Иванникова РАН, 109004, Россия, г. Москва, ул. А. Солженицына, д. 25.*

[2] *Московский физико-технический институт, 141701, Россия, Московская область, г. Долгопрудный, Институтский пер., д. 9.*

[3] *Московский государственный университет имени М.В. Ломоносова, 119991, Россия, Москва, Ленинские горы, д. 1.*

[4] *Федеральный исследовательский центр «Информатика и управление» РАН, 119333, Россия, г. Москва, ул. Вавилова, д. 44.*

Аннотация. Несмотря на высокую эффективность графовых нейронных сетей в широком круге задач, связанных с графовыми структурами, их применение в ответственных областях остается ограниченным из-за недостаточной интерпретируемости. В этом контексте особый интерес представляют модели прозрачного ящика для графовых данных, поскольку процесс получения предсказания в таких моделях является прозрачным и не требует использования внешних постфактум-методов объяснения. Однако если для табличных данных подходы прозрачного ящика имеют устоявшуюся классификацию и обычно связываются с аддитивными, древовидными моделями и моделями на правилах, то для графовых данных подобная систематизация пока только формируется. В работе предлагается таксономия и систематизация подходов прозрачного ящика для графовых данных с выделением аддитивного, древовидного классов и класса моделей на правилах. В дополнение к таксономической части представлен иллюстративный анализ, демонстрирующий, что прозрачность механизма предсказания в моделях прозрачного ящика для графовых данных позволяет проследить логику принятия решения и использовать ее для анализа графовых данных. Кроме того, в статье обсуждаются ограничения существующих моделей прозрачного ящика для графов и перспективы дальнейшего развития этого направления. Проведенный анализ позволяет рассматривать такие модели как одно из формирующихся направлений интерпретируемого машинного обучения для графовых данных.

Ключевые слова: машинное обучение на графах; интерпретируемое машинное обучение; модели прозрачного ящика; графовые нейронные сети; аддитивные модели; деревья решений; модели на правилах.

Soloviov M.V., Drobyshevskiy M.D., Moiseikin A.D., Poltavtsev E.Y., Poturilo E.N., Khalko I.S. Glassbox Models for Graph Data: Taxonomy, Illustrative Analysis, and Perspectives on an Emerging Direction. *Trudy ISP RAN/Proc. ISP RAS*, 2026, vol. 38, issue 6, part 2, pp. 93-116.

Для цитирования: Соловьев М.В., Дробышевский М.Д., Моисейкин А.Д., Полтавцев Е.Ю., Потурило Е.Н., Халько И.С. Модели прозрачного ящика для графовых данных: таксономия, иллюстративный анализ и перспективы формирующегося направления. Труды ИСП РАН, 2026, том 38, вып. 6, часть 2, с. 93-116. DOI: 10.15514/ISPRAS-2026-38(6)-27.

Благодарности. Результаты получены с использованием услуг Центра коллективного пользования Института системного программирования им. В.П. Иванникова РАН – ЦКП ИСП РАН.

Glassbox Models for Graph Data: Taxonomy, Illustrative Analysis, and Perspectives on an Emerging Direction

[1,2] *M.V. Soloviov, ORCID: 0009-0009-2243-273X <m.solovev@phystech.edu>*
[1,2] *M.D. Drobyshevskiy, ORCID: 0000-0002-1639-9154 <drobyshevsky@ispras.ru>*
[1,3] *A.D. Moiseikin, ORCID: 0009-0009-4546-5681 <s02230192@gse.cs.msu.ru>*
[2,4] *E.Y.Poltavtsev, ORCID: 0009-0008-6600-887X <poltavtsev.eu@phystech.edu>*
[2,4] *E.N. Poturilo, ORCID: 0009-0007-5725-4801 <poturilo.en@phystech.edu>*
[2,4] *I.S. Khalko, ORCID: 0009-0002-7373-4657 <khalko.is@phystech.edu>*

[1] *Ivannikov Institute for System Programming of the RAS, 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.*

[2] *Moscow Institute of Physics and Technology, 9, Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russia.*

[3] *Lomonosov Moscow State University, GSP-1, Leninskie Gory, Moscow, 119991, Russian Federation.*

[4] *Federal Research Center "Computer Science and Control" of the RAS, 44, Vavilova st., Moscow, 119333, Russia.*

Abstract. Despite the high effectiveness of graph neural networks across a wide range of tasks involving graph-structured data, their use in high-stakes domains remains limited due to insufficient interpretability. In this context, glassbox models are of particular interest, since the prediction process in such models is transparent and does not require external post-hoc explanation methods. However, while glassbox approaches for tabular data have an established classification and are commonly associated with additive, tree-based, and rule-based models, a comparable systematization for graph data is still emerging. We present a taxonomy and systematization of glassbox approaches for graph data, distinguishing additive, tree-based, and rule-based classes. We also present an illustrative analysis demonstrating that the transparency of the prediction mechanism in glassbox models for graph data makes it possible to trace the logic of decision-making and to use it for graph data analysis. The paper discusses the limitations of existing glassbox models for graph data and the prospects for further development of this direction. The analysis suggests that glassbox models for graph data can be regarded as a conceptually important and promising emerging direction of interpretable machine learning for graph data.

Keywords: graph machine learning; interpretable machine learning; glassbox models; graph neural networks; additive models; decision trees; rule-based models.

For citation: Soloviov M.V., Drobyshevskiy M.D., Moiseikin A.D., Poltavtsev E.Y., Poturilo E.N., Khalko I.S. Glassbox Models for Graph Data: Taxonomy, Illustrative Analysis, and Perspectives on an Emerging Direction. Trudy ISP RAN/Proc. ISP RAS, 2026, vol. 38, issue 6, part 2, pp. 93-116 (in Russian). DOI: 10.15514/ISPRAS-2026-38(6)-27.

Acknowledgements. The results were obtained using the services of the Ivannikov Institute for System Programming (ISP RAS) Data Center.

1. Введение

В последние годы графовые нейронные сети (graph neural network, GNN) зарекомендовали себя как мощный инструмент для решения задач, связанных с графовыми структурами [1,2]. Графы находят применение в различных областях, таких как анализ социальных сетей, молекулярное моделирование, рекомендательные системы и другие сферы, где данные естественным образом представлены в виде графов. GNN способны эффективно моделировать сложные взаимосвязи между объектами, используя информацию о структуре связей, что позволяет достигать высокой точности предсказаний и гибкости в работе с различными типами данных [3-5].

Однако, несмотря на успехи в ряде прикладных задач, широкое применение GNN в ответственных областях, таких как биомедицина, финансы или правовая аналитика, ограничено их низкой интерпретируемостью [6]. Эти модели часто рассматриваются как “черные ящики” [7], что делает процесс принятия решений непрозрачным для пользователя и затрудняет их использование в сценариях, требующих высокой степени доверия и объяснимости.

Существующие подходы к интерпретации графовых моделей можно условно разделить на два класса: постфактум-объяснения (post-hoc explanations) и самоинтерпретируемые модели (self-explainable models). Постфактум-методы, такие как GNNExplainer [8], PGExplainer [9] и SubgraphX [10], строят объяснения поверх уже обученной модели, однако не гарантируют согласованность с ее внутренней логикой [11-13]. Это означает, что объяснения, полученные с помощью таких методов, могут неполностью отражать то, как модель на самом деле принимает решения, поскольку они строятся на основе уже готового результата, а не на основе анализа внутренней логики модели. В отличие от постфактум-методов, модели с встроенной интерпретируемостью обеспечивают более стабильную и достоверную интерпретацию, поскольку механизм объяснения и сама модель проектируются совместно, что позволяет достичь более точных и надежных объяснений, чем при использовании универсальных постфактум-методов [14-17]. Тем не менее, даже такие модели в общем случае не являются полностью прозрачными, поскольку некоторые аспекты их работы, например, взаимодействия признаков, могут оставаться скрытыми для пользователя.

Таким образом, наличие встроенного механизма объяснения не гарантирует полной прозрачности модели. В этом контексте особый интерес представляют модели прозрачного ящика (glassbox models) – модели, механизм предсказания которых допускает явное описание и не требует внешних постфактум-объяснений [7]. В строгом смысле модель прозрачного ящика должна допускать восстановление предсказания на основе явно заданной структуры модели без опоры на скрытые внутренние состояния, не входящие в описание этого предсказания.

Для табличных данных такие подходы хорошо изучены и обычно включают аддитивные, древовидные модели и модели на правилах, что отражено в ряде работ по интерпретируемому машинному обучению [18-19].

В отличие от табличного случая, где модели прозрачного ящика хорошо изучены и имеют устоявшуюся классификацию, для графовых данных подобная систематизация на сегодняшний день отсутствует, а сами подходы остаются фрагментарными и представлены ограниченным числом работ. Существующие работы в основном сосредоточены либо на постфактум-объяснении GNN, либо на более широком классе самоинтерпретируемых моделей, которые по своей сути не являются моделями прозрачного ящика.

В данной работе мы ставим целью предложить таксономию и систематизировать подходы прозрачного ящика для графовых данных. В качестве основы мы используем классификацию, принятую в интерпретируемом машинном обучении для табличных данных, где модели прозрачного ящика обычно разделяются на аддитивные модели, древовидные модели и

модели на правилах. Эта классификация используется как классификация формы прозрачного предсказания: через декомпозицию предсказания на вклады признаков, через последовательность правил разбиения или через явные логические и программные правила.

В дополнение к таксономической части мы приводим иллюстративный анализ, демонстрирующий, каким образом встроенная интерпретируемость моделей прозрачного ящика может быть использована для анализа графовых данных. В отличие от многих работ, где интерпретируемость ограничивается демонстрацией отдельных локальных объяснений для выбранных примеров, мы рассматриваем сценарии, в которых интерпретация используется как полноценный инструмент анализа. В частности, мы показываем, как интерпретируемость позволяет переходить от объяснения отдельных предсказаний к выявлению общих закономерностей в данных и поведении модели, что делает ее практически полезной, а не только иллюстративной. Кроме того, работа включает обсуждение ограничений существующих методов, анализ ключевых трудностей, а также рассмотрение перспективных направлений развития моделей прозрачного ящика для графовых данных.

2. Таксономия и систематизация моделей прозрачного ящика для графовых данных

В задачах интерпретируемого машинного обучения к числу наиболее известных моделей прозрачного ящика традиционно относят аддитивные модели, деревья решений и модели на правилах. Общей чертой этих подходов является прозрачная структура предсказания, позволяющая анализировать вклад входных данных без привлечения внешних постфактум-методов [20].

Однако большинство классических моделей прозрачного ящика изначально разрабатывались для табличных данных, в которых каждый объект представлен фиксированным набором признаков [20-22]. В случае графовых данных ситуация существенно усложняется: предсказание должно учитывать не только собственные признаки объекта, но и структуру связей, а также характеристики локального окружения [2, 23]. Это делает прямой перенос стандартных моделей прозрачного ящика на графы нетривиальной задачей и требует явного учета графового контекста в структуре модели. Вместе с тем базовые формы прозрачного предсказания сохраняют смысл и при переходе к графовым данным. В аддитивных моделях решение раскладывается на сумму отдельных вкладов, в деревьях решений задается последовательностью разбиений, а в моделях на правилах определяется набором логических или программных условий. Поэтому при адаптации этих подходов к графам меняется не сама форма прозрачного предсказания, а способ, которым в нее включаются связи между объектами и информация о локальном окружении.

При переносе такой классификации на графовые данные важно учитывать, что граф задает не только набор признаков объекта, но и отношения между объектами. Поэтому наряду с формой прозрачного предсказания необходимо учитывать уровень, на котором решается задача. Для графовых данных обычно выделяют три основных типа задач: задачи на уровне вершин, ребер и графов. В данной работе эта характеристика отражена в сводной табл. 1, где для каждого метода указаны типы задач, к которым он применим. Не менее важно, каким образом модель включает графовую структуру в механизм предсказания: через агрегацию по локальному окружению, использование путей и подграфов, структурные предикаты или правила над вершинами и ребрами. Эти аспекты рассматриваются далее при подробном описании отдельных методов. Таким образом, аддитивные модели, деревья решений и модели на правилах задают основную форму прозрачного предсказания, а уровень задачи и способ учета графовой структуры показывают, как эта форма реализуется в конкретных подходах прозрачного ящика для графовых данных.

В последние годы появились отдельные работы, направленные на адаптацию классических моделей прозрачного ящика к графовым данным [24-26]. Эти подходы, как правило, развивают идеи, лежащие в основе аддитивных моделей, древовидных моделей и моделей на правилах, однако пока не образуют единого и систематизированного направления. В частности, для графовых данных отсутствует устоявшаяся классификация моделей прозрачного ящика, аналогичная табличному случаю.

В данном разделе мы рассматриваем существующие подходы прозрачного ящика для графовых данных и группируем их в соответствии с формой прозрачного предсказания. В частности, последовательно рассматриваются:

- аддитивные модели (включая линейные модели как их частный случай),
- древовидные модели,
- модели на правилах.

Для каждого из классов обсуждаются существующие подходы, позволяющие учитывать графовую структуру, а также их интерпретационные свойства, ограничения и области применимости.

На текущий момент число работ, непосредственно относящихся к моделям прозрачного ящика для графовых данных невелико. В рамках проведенного анализа были выявлены подходы, соответствующие выделенным классам. Сводная информация по рассмотренным моделям представлена в табл. 1.

Табл. 1. Существующие модели прозрачного ящика для графовых данных.
Table 1. Existing glassbox models for graph data.

Модель	Класс	Год	Тип задач	Ключевая идея	Форма интерпретации
GNAM [24]	адди-тивная	2024	классификация вершин и графов	Расширение обобщенных аддитивных моделей (GAM) на графовые данные: предсказание представляется как сумма вкладов признаков с учетом локального графового окружения	Глобальная и локальная интерпретация через функции формы и явные вклады признаков и окрестности
TREE–G [25]	древо-видная	2024	классификация и регрессия на уровне вершин, ребер и графов	Адаптация дерева решений к графовым данным за счет специальных функций разбиения, учитывающих структуру графа, использующих как признаки, так и топологическую информацию	Интерпретация через путь в дереве и последовательность правил разбиения
PL4XGL [26]	модель на правилах	2024	классификация графов и вершин	Классификация задается с помощью программ на специальном языке, описывающих структурные паттерны в графе	Интерпретация в виде явных программных или логических правил, непосредственно участвующих в предсказании
SYMGRAPH [27]	модель на правилах	2026	классификация графов и вершин	Символическая модель графового обучения, в которой предсказание строится через структурные предикаты, объединяющие топологические шаблоны и семантические условия, с последующей агрегацией в виде счетчиков предикатов	Интерпретация в виде явных логических правил и предикатов, а также их комбинаций, определяющих итоговое решение модели

2.1 Аддитивные модели

Аддитивные модели относятся к числу основных интерпретируемых методов машинного обучения [18, 28], поскольку задают предсказание в виде суммы вкладов отдельных компонентов модели. Такая структура позволяет явно декомпозировать итоговое решение на эффекты отдельных признаков и, при необходимости, их взаимодействий. В отличие от сложных нелинейных архитектур, где вклад признаков формируется неявно во внутренних представлениях, в аддитивных моделях он задается через функции формы, каждая из которых параметризует вклад одного, либо пары признаков. Теоретически аддитивные модели могут учитывать взаимодействия более высокого порядка, однако на практике обычно ограничиваются одномерными и двумерными компонентами, поскольку их можно напрямую анализировать и визуализировать, тогда как функции более высокой размерности существенно теряют в интерпретируемости.

В общем виде аддитивная модель может быть записана следующим образом:

$$g(E[y]) = \beta_0 + \sum_{j=1}^{p} f_j\left(x_j\right) + \sum_{j<k} f_{jk}\left(x_j, x_k\right),$$

где y – целевая переменная, x_j – значение j-го признака, $f_j(\cdot)$ – одномерная функция формы, моделирующая вклад признака x_j, $f_{jk}(\cdot,\cdot)$ – двумерная функция формы, описывающая совместный вклад пары признаков x_j и x_k, β_0 – свободный член, а $g(\cdot)$ – функция связи, определяемая постановкой задачи. В случае регрессии обычно используется тождественная функция связи, а в задаче бинарной классификации (разделения объектов по принадлежности к двум взаимоисключающим классам – положительному и отрицательному) применяют функцию

$$g(p) = \log\left(\frac{p}{1-p}\right), \text{где } 0 < p < 1.$$

Здесь p представляет собой вероятность принадлежности объекта к положительному классу. При этом вероятность принадлежности объекта к отрицательному классу равна $(1-p)$.

Интерпретируемость таких моделей обусловлена тем, что каждый одномерный или двумерный компонент может быть проанализирован отдельно. Визуализация функций формы $f_j\left(x_j\right)$ и $f_{jk}\left(x_j, x_k\right)$ позволяет исследовать как индивидуальное влияние признаков на предсказание, так и характер их взаимодействия, включая возможные нелинейные эффекты. При этом явное представление модели в виде суммы отдельных компонентов сохраняет прозрачность ее структуры даже при учете попарных взаимодействий.

На практике функции формы в аддитивных моделях могут задаваться различными способами в зависимости от метода обучения. В частности, для их построения используются сплайны – кусочно-полиномиальные функции с заданными условиями гладкости в точках соединения [18], ансамбли неглубоких деревьев решений, обучаемые методом бустинга, при котором новые деревья последовательно корректируют ошибки уже построенного ансамбля, например в методе EBM (Explainable Boosting Machine) [29-30], а также нейросетевые подходы [31], в которых каждая функция формы параметризуется отдельной небольшой нейронной сетью.

Следует различать способ обучения функции формы и форму, в которой она используется для интерпретации. В аддитивных моделях прозрачность определяется прежде всего тем, что итоговое предсказание раскладывается на сумму явно выделенных вкладов отдельных признаков или их пар. При этом сами функции формы могут обучаться различными способами. Например, в Explainable Boosting Machine функции формы строятся на основе ансамблей неглубоких деревьев и после обучения могут быть представлены как набор кусочно-постоянных или кусочно-линейных зависимостей на области значений признака.

Такое представление позволяет непосредственно анализировать вклад признака в предсказание.

Нейросетевые функции формы также могут использоваться в аддитивных моделях, однако в этом случае прозрачность обеспечивается не внутренним устройством многослойного перцептрона (Multilayer Perceptron, MLP), а явным низкоразмерным представлением обученной функции формы. Если одномерная или двумерная функция формы визуализируется, табулируется или представляется в виде кусочно-линейной функции на области значений обучающей выборки, ее вклад может быть непосредственно проанализирован пользователем. В таком случае MLP выступает как инструмент обучения функции формы, тогда как интерпретируемым объектом является сама функция формы и ее вклад в аддитивное предсказание. Если для интерпретации используется приближенное представление функции формы, необходимо контролировать точность такой аппроксимации или использовать это представление как фактическую форму модели при вычислении предсказания.

Отдельно отметим, что **линейные модели** можно рассматривать как частный случай аддитивных. Действительно, если в качестве функций формы выбрать линейные функции $f_j(x_j) = w_j x_j$, то аддитивная модель принимает вид

$$\hat{y} = \beta_0 + \sum_{j=1}^{p} w_j\, x_j,$$

что соответствует стандартной линейной модели. Таким образом, линейная модель представляет собой наиболее простой случай аддитивного подхода, в котором вклад каждого признака задается постоянным коэффициентом.

2.1.1 Аддитивные модели для графовых данных. GNAM

Несмотря на высокую интерпретируемость аддитивных моделей, их классическая постановка предполагает, что каждый объект описывается фиксированным вектором признаков. Для графовых данных этого недостаточно, поскольку предсказание для вершины или графа должно учитывать не только собственные признаки объекта, но и информацию о его локальном окружении и структуре связей. В последние годы были предложены подходы, направленные на перенос аддитивных моделей прозрачного ящика на графовые данные, при котором сохраняется интерпретируемая структура предсказания, но одновременно учитывается информация о локальном графовом окружении. В исследовании [24] авторы вводят понятие *аддитивной графовой нейронной сети* (Graph Neural Additive Network, GNAN) – класса интерпретируемых аддитивных моделей для графовых данных. В рамках этого класса предлагается также конкретная *аддитивная графовая нейронная модель* (Graph Neural Additive Model, GNAM), демонстрирующая практическую реализуемость данного подхода.

Представление k-го признака вершины v_i в архитектуре GNAM задается следующим образом:

$$[h_i]_k = \sum_{j=1}^{N} \frac{1}{\#dist_i(j,i)} \cdot \rho_k\left(\frac{1}{1+dist(j,i)}\right) \cdot f_k\left([x_j]_k\right),$$

где N – число вершин в графе, $dist(j,i)$ – длина кратчайшего пути от вершины v_j до вершины v_i, $\#dist_i(j,i)$ – число вершин, находящихся от вершины v_i на расстоянии $dist(j,i)$, $[x_j]_k$ – значение k-го признака вершины v_j, $f_k(\cdot)$ – обучаемая функция формы для k-го признака, а $\rho_k(\cdot)$ – обучаемая функция, определяющая вклад признаков в зависимости от расстояния до рассматриваемой вершины.

Таким образом, представление признака формируется как сумма вкладов одноименного признака всех вершин графа, взвешенных с учетом расстояния до вершины v_i. При этом значение признака сначала преобразуется функцией формы $f_k(\cdot)$, а затем масштабируется функцией расстояния $\rho_k(\cdot)$. Нормировка на $\#dist_i(j,i)$ обеспечивает усреднение вкладов вершин, расположенных на одинаковом расстоянии от v_i.

Из этой записи видно, что для каждого признака обучается одна общая функция формы $f_k(\cdot)$, которая применяется ко всем вершинам независимо от их расстояния до рассматриваемой вершины. Пространственная структура графа при этом учитывается отдельно с помощью функции расстояния $\rho_k(\cdot)$, задающей относительный вес вкладов из различных окрестностей.

После вычисления вкладов отдельных признаков формируется представление вершины как сумма по всем признакам: $h_i = \sum_{k=1}^{K}[h_i]_k$, где K – число признаков вершины.

В задаче классификации графов представление всего графа получается агрегированием представлений всех вершин: $h_G = \sum_{i=1}^{N} h_i$.

Итоговое предсказание модели определяется в зависимости от типа задачи: для регрессии: $y = h$, для бинарной классификации: $y = \sigma(h)$, для многоклассовой классификации: $y = softmax(h)$, где h обозначает представление вершины или графа в зависимости от рассматриваемой постановки задачи.

Заметим, что данная конструкция сохраняет основную идею аддитивного моделирования: итоговое представление объекта раскладывается на сумму вкладов отдельных признаков. При этом, в отличие от классических аддитивных моделей для табличных данных, вклад каждого признака определяется не только его локальным значением, но и значениями того же признака в окрестности вершины. Благодаря этому GNAM можно рассматривать как естественное расширение аддитивных моделей на графовые данные.

Кроме того, описанный подход в той же логике применим и к линейным моделям. Если вместо произвольной функции формы $f_k(\cdot)$ использовать линейную зависимость вида $f_k(x) = w_k x$, то получается линейная модель для графовых данных, в которой вклад признака также зависит от его значений в локальном графовом окружении и взвешивается с учетом расстояния до рассматриваемой вершины. Таким образом, графовые линейные модели можно рассматривать как частный случай графовых аддитивных моделей.

2.2 Древовидные модели

В древовидных моделях интерпретируемость достигается за счет представления процесса принятия решения в виде последовательности явных правил. В классическом случае дерево решений рекурсивно разбивает пространство признаков на области, каждой из которых соответствует определенное предсказание [32]. Благодаря этому предсказание для конкретного объекта можно объяснить через путь от корня дерева к листу.

Формально предсказание дерева решений записывается как

$$\hat{y}(x) = \sum_{m=1}^{M} c_m\, I(x \in R_m),$$

где M – число листьев дерева, R_m – область пространства признаков, соответствующая m-му листу, c_m – предсказание в этом листе, а $I(x \in R_m)$ – индикатор принадлежности объекта x области R_m. Каждая такая область формируется последовательностью разбиений по отдельным признакам, например, $x_j \le t$, где x_j – значение j-го признака, а t – порог.

Интерпретируемость деревьев решений обусловлена тем, что логика предсказания задается в явной дискретной форме: отдельные ветви дерева можно рассматривать как самостоятельные правила. Это позволяет получать как локальные объяснения для конкретных объектов, так и глобальное представление о структуре модели.

Построение дерева обычно сводится к последовательному выбору разбиений, уменьшающих неоднородность данных в дочерних узлах. Например, в задаче классификации в качестве критерия может использоваться индекс Джини: $G(Q) = 1 - \sum_{c=1}^{C} p_c^2$, где Q – множество объектов в узле, C – число классов, а p_c – доля объектов класса c в множестве Q.

2.2.1 Древовидные модели для графовых данных. TREE-G

Классические деревья решений изначально были разработаны для табличных данных. В графовом случае предсказание должно учитывать не только признаки отдельных вершин или графов, но и структуру связей между ними. Поэтому перенос древовидных моделей на графовые данные требует специальных правил разбиения, учитывающих структуру графа. Одним из подходов к адаптации деревьев решений к графовым данным является TREE-G [25] – модификация классического дерева решений для задач классификации и регрессии на уровне вершин, ребер и графов.

Пусть граф задан как $G = (V, E)$, а матрица признаков вершин – как $X \in R^{|V| \times d}$. В отличие от обычного дерева, TREE-G использует разбиения не по исходным признакам напрямую, а по агрегированным признакам, учитывающим локальную структуру графа. Для этого вводится функция $\phi(v; j, \ell, M, S)$, вычисляющая агрегированное значение j-го признака в окрестности вершины v.

Условие разбиения можно записать как

$$\phi(v; j, \ell, M, S) \leq \tau,$$

где j – индекс признака, ℓ – длина путей, исходящих из вершины v, по которым производится агрегация, M – тип маски, определяющий допустимые типы ходов в графе, S – подмножество вершин, а τ – порог.

Параметр M задает тип маски, то есть определяет, какие именно пути учитываются при вычислении агрегированного признака. В работе для TREE-G рассматриваются, в частности, маски Source-walks и Cycle-walks. В случае Source-walks учитываются пути длины ℓ, исходящие из рассматриваемой вершины v, тогда как Cycle-walks ограничивает рассмотрение циклическими путями, начинающимися и заканчивающимися в вершине v.

Агрегированные признаки удобно выражать через матрицу смежности. Для вершины v_i и признака k признак ℓ-й окрестности имеет вид

$$\left[\phi_i^{(\ell)}\right]_k = \sum_{j=1}^{|V|} [A^\ell]_{ij}\, x_{jk},$$

где A – матрица смежности графа.

Таким образом, TREE-G строит правила разбиения на основе признаков, уже учитывающих графовый контекст. Это позволяет сохранить интерпретируемость дерева и одновременно использовать информацию о локальном окружении вершины.

Обучение TREE-G выполняется жадным образом: на каждом шаге выбирается разбиение, минимизирующее критерий ошибки, $\theta^* = argmin_\theta L_{\text{split}}(\theta)$, где $\theta = (j, \ell, M, S, \tau)$. Как и в классическом случае, процесс продолжается до выполнения условий остановки, например ограничения глубины дерева или минимального размера листа. TREE-G может использоваться и в ансамблевых схемах по аналогии со случайным лесом, однако в этом случае интерпретируемость обычно снижается по сравнению с одиночным деревом.

2.3 Модели на правилах

Модели на правилах относятся к числу классических моделей прозрачного ящика, поскольку в них предсказание задается с помощью явных логических, символьных или программных

правил [7]. В отличие от аддитивных моделей, где интерпретируемость обеспечивается разложением предсказания на вклады признаков, и древовидных моделей, где она достигается через последовательность разбиений, в моделях на правилах решение формулируется непосредственно в виде набора условий вида “если …, то …”. Такая форма делает процесс принятия решения прозрачным и позволяет явно проследить, какие свойства объекта привели к конкретному предсказанию.

В общем случае модель на правилах можно представить как набор правил $R = \{r_1, \dots, r_M\}$, где каждое правило r_m состоит из условия применимости и соответствующего вывода [33].

Формально правило можно записать в виде

$$r_m: \varphi_m(x) \Rightarrow y_m,$$

где $\varphi_m(x)$ – логическое условие на объект x, а y_m – предсказываемая метка или решение. Тогда предсказание модели может определяться, например, первым сработавшим правилом, правилом с максимальным приоритетом или лучшим правилом по некоторой функции качества:

$$\hat{y}(x) = y_{m^*}, m^* = arg \max_{m:\varphi_m(x)=\text{true}} s_m,$$

где s_m – качество, вес или приоритет правила r_m.

Интерпретируемость моделей на правилах обусловлена тем, что каждый компонент предсказания задается в явной символьной форме. Это позволяет анализировать как отдельные правила, так и всю систему правил целиком, а также проверять их корректность, полноту и согласованность с предметной областью. В отличие от постфактум-подходов, объяснение здесь не строится поверх уже обученной модели, а непосредственно совпадает с механизмом предсказания.

2.3.1 Модели на правилах для графовых данных. PL4XGL и SYMGRAPH

Классические модели на правилах в основном разрабатывались для табличных данных, где объект задается фиксированным набором признаков. Для графовых данных этого недостаточно, поскольку решение должно зависеть не только от атрибутов отдельных вершин или ребер, но и от структуры связей, типов соседства и более сложных структурных паттернов.

Одним из первых подходов на основе правил, специально ориентированных на графовые данные, является PL4XGL [26] – программно-ориентированная модель объяснимого графового обучения, в которой предсказание и объяснение задаются программами на специальном языке описания графов GDL (Graph Description Language). В отличие от постфактум-методов, строящих объяснение после получения предсказания черным ящиком, PL4XGL использует интерпретируемую модель как основную модель классификации: результат определяется той программой GDL, которая лучше всего описывает входной графовый объект.

В PL4XGL модель представляет собой множество кандидатных программ с метками классов и оценками качества:

$$M \subseteq L \times P \times [0,1],$$

где L – множество меток, P – множество программ GDL, а число из отрезка $[0,1]$ характеризует качество соответствующей пары “метка-программа”. Каждая программа описывает некоторый структурный паттерн в графе и одновременно выступает объяснением предсказания.

Формально поиск предсказания в PL4XGL можно записать как выбор лучшей программы, применимой к рассматриваемому графовому объекту:

$$Search(M, c) = arg \max_{(\ell,p,s)\in M} \begin{cases} s, & \text{if} c \in [p], \\ -1, & \text{otherwise.} \end{cases}$$

где c – классифицируемый объект (вершина, ребро или граф), p – семантика программы p, то есть множество объектов, описываемых этой программой, ℓ – предсказываемая метка, а s – качество программы. Тем самым модель одновременно возвращает и метку класса, и саму программу как объяснение. Существенно, что объяснение в данном случае является корректным по построению: модель принимает решение именно на основе той программы, которая затем предъявляется пользователю.

Язык GDL позволяет задавать правила через символьные описания вершин, ребер и их признаков. В общем виде программа состоит из набора описаний и целевого символа:

$$p = (d_1, \dots, d_K, t),$$

где d_i задают ограничения на вершины или ребра графа, а t указывает, какой объект считается целевым – вершина, ребро или весь граф. Благодаря этому PL4XGL можно рассматривать как модель прозрачного ящика на правилах для графовых данных, в которой предсказание задается набором явных программных правил над структурой графа.

Другой недавний подход – SYMGRAPH [27], в котором графовое обучение формулируется как символическая модель, использующая структурные предикаты и логические правила вместо стандартного механизма передачи сообщений в графовых нейросетях. В этой модели локальные паттерны описываются предикатами, зависящими от структурной роли элементов в окрестности вершины, а итоговое решение строится на их логической агрегации.

На локальном уровне предикат можно записать как

$$p_j(v) \Leftrightarrow \Psi_j\big(\psi(O_1, X), \dots, \psi(O_B, X)\big),$$

где $O_1, \dots, O_B$ – топологические орбиты в локальной окрестности вершины v, $\psi(O_i, X)$ – булевы или пороговые характеристики признаков внутри соответствующей орбиты, а Ψ_j – логическая функция, определяющая, удовлетворяет ли окрестность вершины шаблону p_j. Тем самым предикаты в SYMGRAPH играют роль явных структурных правил первого уровня.

Для задачи классификации графов глобальное решение в SYMGRAPH выражается через количественную дизъюнктивную нормальную форму (Q-DNF):

$$\Phi(G) = \bigvee_{m=1}^{M} \Big(\bigwedge_{(j,op,\kappa)\in R_m} (C_j(G) op \kappa) \Big) \Rightarrow y,$$

где $C_j(G) = \sum_{v\in V_G} I\left(p_j(v)\right)$ – число срабатываний предиката p_j в графе G, а каждое правило R_m задает набор количественных ограничений на частоты предикатов. Таким образом, итоговое решение строится уже не по непрозрачным латентным представлениям, а по явным логическим комбинациям структурных паттернов.

С точки зрения интерпретируемости SYMGRAPH близок к подходам прозрачного ящика на основе правил, поскольку использует символические предикаты и логические формулы как основу модели. Однако здесь следует учитывать, что в статье рассматриваются разные способы задания глобального классификатора: в варианте с *деревом решений* (Decision Tree) итоговое решение остается прозрачным и основанным на правилах, тогда как вариант со *случайным лесом* (Random Forest) сами авторы описывают как полупрозрачный (“gray-box”) ансамбль, а не как полностью прозрачную модель. Поэтому SYMGRAPH корректнее трактовать не как безусловную модель прозрачного ящика во всех реализациях, а как символическую модель на правилах, которая в своей наиболее прозрачной конфигурации соответствует классу моделей прозрачного ящика, но в ансамблевых вариантах частично от него отходит.

В итоге оба рассмотренных подхода демонстрируют возможные направления переноса моделей на правилах на графовые данные. PL4XGL делает это через язык описания графов и синтез программ, а SYMGRAPH – через систему структурных предикатов и логических формул над ними. В обоих случаях ключевая идея состоит в том, что объяснение не является внешней надстройкой над моделью, а непосредственно совпадает с механизмом принятия решения. Именно это свойство позволяет рассматривать такие методы как наиболее близкие к классическому пониманию моделей прозрачного ящика на основе правил в графовом контексте.

2.4 Прозрачность модели и сложность объяснения

При рассмотрении моделей прозрачного ящика важно различать формальную прозрачность механизма предсказания и практическую интерпретируемость получаемого объяснения. Мы относим модель к классу прозрачного ящика, если ее предсказание может быть восстановлено из явно заданной структуры модели без обращения к скрытым внутренним состояниям. Для аддитивных моделей такой структурой является сумма вкладов функций формы, для деревьев решений – путь от корня к листу, а для моделей на правилах – набор логических или программных условий, непосредственно участвующих в предсказании.

Однако выполнение этого критерия само по себе не гарантирует, что объяснение будет легко восприниматься человеком. Аддитивная модель с большим числом признаков и взаимодействий может порождать слишком много функций формы для анализа; дерево решений большой глубины может терять наглядность; набор правил или программа могут быть формально точными, но слишком длинными и сложными для практического использования. Поэтому интерпретируемость таких моделей зависит не только от типа предиктора, но и от сложности его явного представления. В связи с этим при использовании моделей прозрачного ящика следует учитывать не только качество предсказания, но и характеристики сложности объяснения: число функций формы, глубину дерева, число листьев, количество правил, длину программы или логической формулы, а также степень семантической понятности используемых признаков и структурных предикатов. Эти характеристики позволяют отделить формальную прозрачность модели от ее практической интерпретируемости.

2.5 Качество предсказания и вычислительная сложность моделей прозрачного ящика

Ограничения на форму модели, необходимые для сохранения прозрачности, потенциально могут снижать ее выразительную способность и качество предсказания. Однако результаты рассмотренных работ показывают, что такой компромисс не является универсальным. В частности, модель GNAM [24] показала лучший или второй результат в девяти из 13 рассмотренных задач, а модель TREE-G [25] превзошла сравниваемые сети GNN в семи из десяти задач классификации графов. Таким образом, на части задач прозрачная структура модели не приводит к существенному снижению качества и может быть конкурентоспособна с GNN.

Более выраженный компромисс наблюдается для правиловых моделей. Подход PL4XGL [26], например, достигает 100.0% точности на наборе данных MUTAG против 91.0% у лучшей из сравниваемых GNN и сопоставимого качества на наборах BBBP и BACE, однако уступает GNN на наборах Cora (80.0% против 86.7%) и Citeseer (63.8% против 79.1%). Это показывает, что цена прозрачности зависит от того, насколько выразительности используемого языка правил достаточно для конкретной задачи. Сходная ситуация наблюдается в подходе SYMGRAPH [27]: вариант с деревом решений сохраняет прозрачность, тогда как

использование *случайного леса* может повысить качество, но переводит модель в класс неполностью прозрачных (“gray-box”).

С вычислительной точки зрения прозрачность также не обязательно означает увеличение стоимости. Для модели TREE-G сложность обучения возрастает из-за поиска структурных разбиений и зависит от числа признаков и размера рассматриваемой окрестности [25]. В подходе PL4XGL синтез программ может быть вычислительно дорогим, однако полученная программа одновременно является предсказателем и объяснением, поэтому отдельный post-hoc этап отсутствует; авторы сообщают как минимум о 35-кратном ускорении процедуры “классификации и объяснения” (classification+explanation) по сравнению с использованием метода интерпретации SubgraphX графовой сети (GNN+SubgraphX) [26].

Таким образом, существующие результаты не подтверждают универсального правила, согласно которому повышение прозрачности неизбежно приводит к снижению точности или росту вычислительной стоимости. Компромисс определяется сложностью задачи, выразительностью прозрачной модели и способом учета графовой структуры.

3. Иллюстративный анализ

Данный раздел дополняет таксономическую часть и показывает практическую ценность встроенной интерпретируемости моделей прозрачного ящика для графовых данных. Такой акцент важен, поскольку в литературе интерпретируемость часто декларируется как свойство модели, но реже рассматривается в контексте конкретных аналитических сценариев, где она действительно помогает выявлять закономерности в данных.

В этой связи в работе приводится иллюстративный анализ с аддитивной моделью прозрачного ящика на наборе данных MUTAG. На этом примере (рис. 1) демонстрируется, что интерпретируемость может выступать не только как средство объяснения отдельных предсказаний, но и как инструмент анализа данных: выявления значимых признаков, структурных закономерностей и проверки согласованности логики модели с предметной областью.

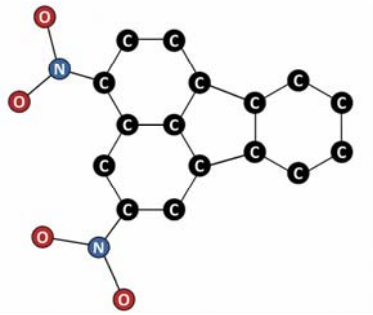


Рис. 1. Пример молекулы из набора данных MUTAG.
Fig. 1. Example molecule from the MUTAG dataset.

Подобный подход соответствует более широкой традиции интерпретируемого машинного обучения, где модели прозрачного ящика используются не только как прозрачная альтернатива “черным ящикам”, но и как средство получения нового знания. В частности, для аддитивных моделей показательным примером является работа [34], где интерпретируемая структура модели позволила выявить нетривиальные и практически значимые закономерности в медицинских данных. В настоящей работе аналогичная идея переносится на графовый контекст: нас интересует не только качество предсказания, но и то, какую дополнительную информацию о графовых данных позволяют извлечь модели прозрачного ящика.

3.1 Выявление значимых признаков в графовых данных с помощью аддитивных моделей (MUTAG)

MUTAG [35] – это химический графовый набор данных, широко используемый в задачах классификации графов. Каждый граф в этом наборе данных представляет собой молекулу, где вершины соответствуют атомам, а ребра – химическим связям между ними. Целевая переменная указывает на мутагенность соединения, то есть способность молекулы вызывать генетические мутации. Данная характеристика является бинарной: графы разделены на два класса – мутагенные и немутагенные молекулы. Каждый узел графа снабжен дискретным признаком, соответствующим типу атома (C, N, O, F, I, Cl, Br).

3.1.1 Архитектура модели

Для иллюстративного анализа была обучена нейросетевая аддитивная модель, близкая по структуре к GNAM (см. раздел 2.1.1) с рядом следующих модификаций:

1. Для построения признакового представления вершины использовалась идея, близкая к модели GNAM (см. раздел 2.1.1), однако с учетом бинарной природы исходных признаков. Поскольку признаки вершин принимают значения $0/1$, в рассматриваемой модели агрегация по графу переносится внутрь функции формы: сначала для вершины v_i и признака k вычисляется агрегированный признак
$$[\text{ж}_i]_k = \sum_{j \in N(i)} 1\left(\left[x_j\right]_k \neq 0\right),$$
где $N(i)$ – множество соседей вершины v_i, а затем функция формы применяется уже к этому агрегированному значению. Таким образом, $[\text{ж}_i]_k$ отражает количество соседних вершин, обладающих признаком типа k. При этом, в отличие от исходной модели GNAM, учитывалась только первая окрестность, поскольку экспериментально было установлено, что использование более дальних окрестностей не приводит к улучшению качества, но существенно увеличивает вычислительную сложность.
2. Дополнительно учитываются попарные взаимодействия признаков. Множество таких взаимодействий задается следующим образом:
$$I = \{(i,j) \vee i < j, i, j \in \{1, \dots, K\}\},$$
где K – число признаков. Такое определение исключает дублирование пар (i,j) и (j,i), а также не включает тривиальные случаи $i = j$, которые уже учтены в одномерных функциях формы.
3. Функции формы $f_k(x_k)$ и $f_{k\ell}(x_k, x_\ell)$ параметризуются нейронными сетями. Для попарных функций используется модифицированная версия MLP с gating-механизмом: пусть $x = (x_i, x_j)$ – вход пары признаков, тогда выход функции формы определяется как
$$f_{ij}(x_i, x_j) = MLP(x_i, x_j) \cdot 1(x_i \neq 0 \wedge x_j \neq 0).$$
Таким образом, вклад взаимодействия учитывается только в случае, когда оба признака ненулевые. Это позволяет избежать дублирования эффектов: если один из признаков равен нулю, соответствующий вклад уже моделируется одномерной функцией формы.

Модель с вышеописанной архитектурой была обучена на наборе данных MUTAG. Оценка качества проводилась по 10 независимым случайным разбиениям выборки в соотношении 80%/20%: на каждом запуске 80% графов использовались для обучения, а оставшиеся 20% – для тестирования. По итогам экспериментов качество модели составило $F1_{train} = 89.7 \pm 0.8, F1_{test} = 87.5 \pm 1.1$.

3.1.2 Оценка важности признаков

Для анализа влияния признаков на результат работы модели рассматривается метрика глобальной важности признаков. Пусть D_{train} – обучающая выборка, C – число классов, а $f_c(x)$ – функция формы для некоторого признака (или пары признаков) и класса c.

В литературе, в частности в модели EBM (Explainable Boosting Machine) [29-30], используется следующий способ оценки важности:

$$I_1 = \sum_{c=1}^{C} E_x\,[|f_c(x)|].$$

Здесь математическое ожидание берется по области определения функции формы. Таким образом, данный подход характеризует среднюю амплитуду функции формы как математического объекта.

В данной работе мы вводим альтернативную метрику, учитывающую эмпирическое распределение признаков на обучающей выборке:

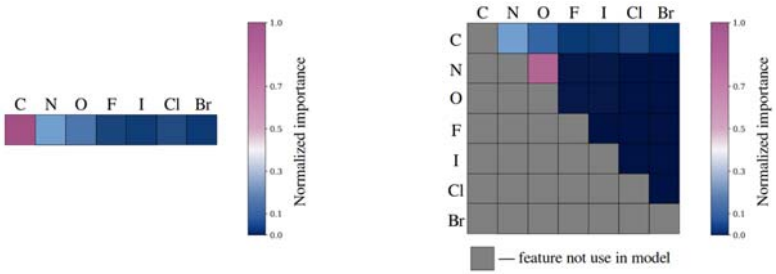
$$I_2 = |D_{train}| \cdot \sum_{c=1}^{C} E_{x\sim D_{train}}\,[|f_c(x)|].$$

Преимущество I_2 по сравнению с I_1 состоит в том, что I_2 в большей степени отражает репрезентативное влияние признака на выборке. Метрика I_1 может присваивать высокий приоритет признакам, для которых функция формы принимает большие значения лишь в редких точках пространства признаков. Это делает ее чувствительной к единичным или случайным объектам обучающей выборки. Напротив, I_2 позволяет выделять признаки, влияние которых существенно не только по величине, но и по распространенности среди объектов, что делает такую оценку более содержательной при анализе глобальной важности признаков.

На рис. 2 представлена тепловая карта важности признаков и их попарных взаимодействий, нормированная в диапазон $[0,1]$.

Из анализа следует, что наибольший вклад в предсказание вносят:

- признак, соответствующий количеству атомов углерода (C) в окрестности вершины,
- взаимодействие признаков, соответствующих атомам азота и кислорода (N, O).



(a) Важность признаков.
(a) Feature importance.

(b) Важность комбинаций признаков.
(b) Feature-combination importance.

Рис. 2. Тепловая карта важности признаков и комбинаций признаков.
Fig. 2. Heatmap of feature and feature-combination importance.

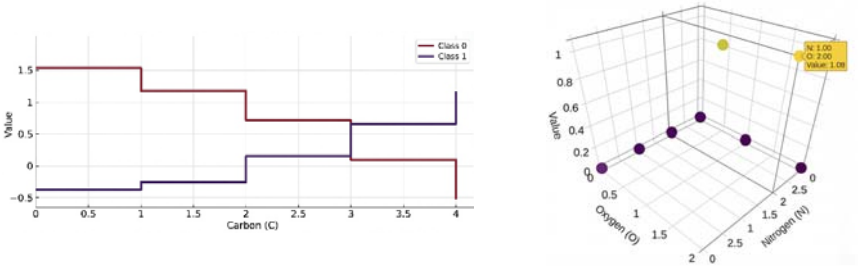
3.1.3 Упрощенная модель

Для проверки значимости выявленных факторов была обучена модель, использующая только указанные признаки и их взаимодействие. Если раньше модель состояла из 7 одномерных функций формы для каждого из признаков и 21 двумерной функции формы для парных взаимодействий, то упрощенная модель складывается из одной одномерной функции формы для признака, соответствующего количеству атомов углерода (C) в окрестности вершины и одной двумерной функции формы для взаимодействия признаков, соответствующих атомам азота и кислорода (N, O). Качество такой модели составило

$$F1_{train} = 86.9 \pm 1.2, F1_{test} = 85.6 \pm 1.3.$$

Незначительное снижение качества по сравнению с полной моделью свидетельствует о том, что данные признаки действительно вносят существенный вклад при определении мутагенности молекул.

Рассмотрим график функции формы для признака 0 (углерод, C), представленный на рис. 3(a).



(a) Функция формы для признака 0, углерод (C).
(a) Shape function for feature 0 (carbon, C).

(b) Взаимодействие признаков азота (N) и кислорода (O), класс 1.
(b) Nitrogen–oxygen interaction, class 1.

Рис. 3. Графики функций формы обученной модели для отдельного признака и пары признаков.
Fig. 3. Shape-function plots of the trained model for an individual feature and a pair of features.

Видно, что с увеличением количества атомов углерода в окрестности вершины возрастает вклад в предсказание класса 1 и уменьшается вклад в предсказание класса 0. При значениях признака 3 и 4 функция формы начинает отдавать предпочтение первому классу (ее значение становится выше, чем для класса 0). Это означает, что наличие большого числа атомов углерода в первой окрестности вершины положительно влияет на предсказание мутагенности молекулы. Такая закономерность наблюдается у молекул с плотно стыкованными углеродными кольцами. Далее рассмотрим график функции формы для попарного взаимодействия признаков 1 (азот, N) и 2 (кислород, O), представленный на рис. 3(b). Из рисунка следует, что комбинация одного атома азота и двух атомов кислорода в первой окрестности вершины дает положительный вклад в предсказание класса 1. Рассматриваемая комбинация соответствует ситуации, когда вершина, являющаяся атомом азота, имеет в своей первой окрестности два атома кислорода, что соответствует наличию нитрогруппы NO_2 в молекуле.

3.1.4 Вывод

В ряде работ, посвященных постфактум интерпретации графовых нейронных сетей, предполагается, что такие методы способны выявлять ключевые химические подструктуры, влияющие на мутагенность молекул, например, углеродные кольца и нитрогруппы. Однако,

как показано в исследовании [13], постфактум-методы часто демонстрируют значительные расхождения между собой и нестабильность при изменении параметров.

В отличие от подобных подходов, аддитивная модель обеспечивает интерпретируемость на уровне архитектуры, что позволяет выявлять значимые структурные элементы молекул. Проведенный анализ показал, что углеродные кольца (характеризуемые высоким числом атомов углерода в первой окрестности) и наличие нитрогрупп (совместное присутствие атомов N и O) вносят существенный вклад при определении мутагенности молекул. Эти результаты согласуются с химическими знаниями: ароматические углеродные кольца и нитрогруппы действительно известны своей способностью вызывать мутации.

4. Ограничения и перспективы

4.1 Структурные ограничения класса моделей

Одним из ключевых результатов проведенного анализа является наблюдение о крайне ограниченном числе подходов, удовлетворяющих строгому определению моделей прозрачного ящика для графовых данных. Это связано не с отсутствием интереса к интерпретируемым методам, а с фундаментальными ограничениями, накладываемыми самой постановкой задачи.

Во-первых, классические модели прозрачного ящика изначально разрабатывались для табличных данных, где каждый объект представлен фиксированным вектором признаков. В этом случае предсказание может быть явно разложено на вклад отдельных признаков (аддитивные модели), последовательность разбиений (деревья) или набор правил (модели на правилах). Для графовых данных, напротив, предсказание должно учитывать не только признаки объекта, но и структуру связей, а также характеристики локального окружения. Это требует явного включения графового контекста в модель, что существенно усложняет сохранение прозрачности механизма предсказания.

Во-вторых, строгий критерий модели прозрачного ящика существенно сужает множество допустимых подходов. Значительная часть существующих методов интерпретируемого обучения на графах относится либо к постфактум-объяснениям, либо к самоинтерпретируемым моделям, в которых интерпретация не полностью совпадает с вычислительным процессом. В частности, многие подходы используют дополнительные механизмы объяснения, не входящие непосредственно в процедуру предсказания, либо опираются на скрытые представления модели, которые не могут быть однозначно восстановлены из объяснения. В таких случаях, даже при наличии интерпретации, невозможно гарантировать воспроизведение предсказания только на основе объяснения, что нарушает критерий модели прозрачного ящика. Требование полной прослеживаемости предсказания приводит к тому, что лишь ограниченное число моделей удовлетворяет данному критерию.

В-третьих, ограниченность числа подходов объясняется структурой самих моделей прозрачного ящика. В частности, в случае аддитивных моделей разнообразие методов для табличных данных (например, Generalized Additive Models, Neural Additive Models, Explainable Boosting Machines) создается за счет различных способов аппроксимации функций формы – сплайнами, нейронными сетями или ансамблями деревьев. Однако все эти модели реализуют одну и ту же базовую идею аддитивного разложения предсказания.

Аналогично, в графовом случае появление новых моделей не связано с формированием принципиально новых семейств, а в основном сводится к адаптации уже существующих классов к графовой структуре. Любая модель, удовлетворяющая строгому критерию прозрачности, как правило, реализует один из этих принципов или их комбинацию. Это связано с тем, что требование полной прослеживаемости предсказания накладывает

существенные ограничения на форму модели: предсказание должно быть представимо в виде явной композиции интерпретируемых компонентов. В рамках интерпретируемого машинного обучения такие представления, как правило, сводятся к трем базовым схемам: аддитивному разложению, последовательности условий (деревья) или явным логическим правилам. В этом смысле ограниченность числа подходов носит не эмпирический, а структурный характер.

4.2 Ограничения, связанные с представлением признаков

Существенным ограничением применения моделей прозрачного ящика для графовых данных является зависимость их интерпретируемости от формы представления входных признаков. В большинстве существующих работ предполагается, что признаки обладают явным семантическим смыслом, например возраст, доход, степень вершины или тип атома. В таких случаях вклад каждого признака может быть непосредственно интерпретирован.

Во многих практических задачах исходные данные также обладают понятной семантикой. Например, в задачах на графах цитирования вершины могут содержать текстовые документы или абзацы, которые сами по себе являются интерпретируемыми объектами. Однако для применения современных методов машинного обучения такие данные, как правило, преобразуются в плотные векторные представления с использованием трансформерных моделей. После такого преобразования признаки теряют явную семантическую интерпретацию, поскольку отдельные компоненты вектора не соответствуют конкретным понятным человеку характеристикам.

В результате возникает разрыв между интерпретируемостью исходных данных и интерпретируемостью признакового пространства, на котором обучается модель. Модели прозрачного ящика в этом случае формально сохраняют прозрачность, однако объяснения становятся неинтерпретируемыми, поскольку оперируют абстрактными признаками.

Одним из возможных решений является использование более простых и интерпретируемых представлений, например модели “мешка слов”, в которой каждый признак соответствует частоте встречаемости конкретного слова. Однако такой подход, как правило, уступает современным трансформерным моделям по качеству решения задачи.

Таким образом, возникает фундаментальный компромисс между интерпретируемостью и качеством модели. В задачах, где высокое качество достигается за счет сложных, неинтерпретируемых представлений признаков, прозрачные, понятные модели оказываются менее конкурентоспособными. Это ограничивает область их применения и объясняет, почему они чаще используются в задачах, где признаки изначально имеют явную и устойчивую семантику.

4.3 Тренды и перспективные направления

Несмотря на ограниченное число существующих работ, можно выделить несколько направлений, в которых ожидается развитие моделей прозрачного ящика для графовых данных.

Во-первых, развитие, вероятно, будет происходить внутри уже существующих классов моделей, а не через появление принципиально новых типов. Основной акцент смещается на разработку методов, позволяющих учитывать графовую структуру при сохранении прозрачности модели, например через более сложные функции агрегации локального окружения или расширение допустимых типов взаимодействий между признаками.

Во-вторых, перспективным направлением является интеграция символических и программных подходов, позволяющих описывать структурные паттерны в графах. Такие методы могут расширить выразительную способность моделей при сохранении интерпретируемости.

В-третьих, важной задачей остается поиск баланса между интерпретируемостью и качеством. В ряде приложений допустимо некоторое снижение точности в обмен на прозрачность модели, однако для сложных задач с высокоразмерными и неинтерпретируемыми признаками данный компромисс остается существенным препятствием.

Наконец, ожидается развитие гибридных подходов, сочетающих модели прозрачного ящика с более сложными представлениями данных, например через предварительное извлечение интерпретируемых признаков или ограничение пространства признаков таким образом, чтобы сохранить их семантическую интерпретируемость.

5. Заключение

В данной работе предложена таксономия и проведена систематизация подходов прозрачного ящика для графовых данных с выделением аддитивных моделей, древовидных моделей и моделей на правилах. Проведенный анализ показывает, что, несмотря на ограниченное число непосредственно относящихся к этой области работ, сама идея переноса подходов прозрачного ящика на графовые данные является содержательно обоснованной и позволяет выделить уже формирующееся, хотя пока еще фрагментарное направление исследований.

Хотя графовые данные требуют учета структуры связей, локального окружения и топологических зависимостей, фундаментальные принципы построения интерпретируемых моделей остаются теми же. Как и в табличном случае, предсказание может быть организовано либо как сумма прозрачных вкладов признаков и окрестностей, либо как последовательность явных правил разбиения, либо как система логических или программных правил. Это позволяет рассматривать модели прозрачного ящика для графовых данных не как набор разрозненных исключений, а как естественное расширение уже сложившихся направлений интерпретируемого машинного обучения.

Проведенный анализ показал, что текущее число существующих моделей прозрачного ящика для графовых данных невелико, а само направление пока существенно уступает по степени разработанности как классическим интерпретируемым моделям для табличных данных, так и многочисленным постфактум-методам и самоинтерпретируемым подходам для графов. При этом именно модели прозрачного ящика представляют особый интерес, поскольку обеспечивают наиболее строгую форму интерпретируемости: логика предсказания в них выражена непосредственно в самой структуре модели и не требует обращения к внешним объясняющим процедурам.

Дополнительный вывод работы связан с практической значимостью таких моделей. Проведенный иллюстративный анализ показывает, что встроенная интерпретируемость может быть полезна не только как средство объяснения уже полученного предсказания, но и как инструмент анализа самих графовых данных. В этом случае модель позволяет выявлять структурные закономерности, понимать, какие признаки и характеристики окрестности действительно используются при принятии решения, а также соотносить логику модели со способом построения или природой исследуемого графа. Тем самым интерпретируемость выступает не как формальное свойство модели, а как источник дополнительного знания о задаче.

Проведенный анализ позволил обсудить ключевые ограничения и перспективы моделей прозрачного ящика для графовых данных. К числу основных ограничений можно отнести малое количество существующих подходов, фрагментарность текущего направления и ограниченную эмпирическую проработку таких моделей по сравнению с более широко представленными постфактум-методами и самоинтерпретируемыми моделями для графов. Вместе с тем рассмотренные подходы показывают, что модели прозрачного ящика для графовых данных обладают значительным потенциалом развития. Перспективы этого направления связаны с расширением числа моделей внутри выделенных классов, с

адаптацией классических интерпретируемых подходов к графовым данным, а также с развитием прикладных сценариев, в которых встроенная интерпретируемость используется не только для объяснения предсказаний, но и для выявления содержательных закономерностей в данных.

Таким образом, модели прозрачного ящика для графовых данных представляют собой небольшое, но значимое формирующееся направление на пересечении машинного обучения на графах и интерпретируемого машинного обучения. Несмотря на текущую ограниченность числа работ, уже существующие подходы показывают, что прозрачные модели для графовых данных возможны и способны сочетать учет графовой структуры с явной, проверяемой логикой предсказания. Это делает их перспективной основой как для дальнейших теоретических исследований, так и для практических приложений, где наряду с качеством предсказания требуется высокая степень доверия, объяснимости и контролируемости модели.

Список литературы / References

[1]. Wu Z., Pan S., Chen F., Long G., Zhang C., Yu P.S. A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems, 2020, vol. 32, no. 1, pp. 4-24.
[2]. Scarselli F., Gori M., Tsoi A.C., Hagenbuchner M., Monfardini G. The graph neural network model. IEEE transactions on neural networks, 2008, vol. 20, no. 1, pp. 61-80.
[3]. Wang Y., Wang J., Cao Z., Barati Farimani A. Molecular contrastive learning of representations via graph neural networks. Nature Machine Intelligence, 2022, vol. 4, no. 3, pp. 279-287.
[4]. Ying R., He R., Chen K., Eksombatchai P., Hamilton W.L., Leskovec J. Graph convolutional neural networks for web-scale recommender systems. In Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, 2018, pp. 974-983.
[5]. Kearnes S., McCloskey K., Berndl M., Pande V., Riley P. Molecular graph convolutions: Moving beyond fingerprints. Journal of computer-aided molecular design, 2016, vol. 30, pp. 595–608.
[6]. Lipton Z.C. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. Queue, 2018, vol. 16, no. 3, pp. 31-57.
[7]. Molnar C. Interpretable Machine Learning: A Guide for Making Black Box Models Explainable. Available at: https://christophm.github.io/interpretable-ml-book/, accessed 04.09.2026.
[8]. Ying Z., Bourgeois D., You J., Zitnik M., Leskovec J. Gnnexplainer: Generating explanations for graph neural networks. Advances in neural information processing systems, 2019, vol. 32.
[9]. D. Luo et al. Parameterized explainer for graph neural network. arXiv preprint, 2020 DOI: 10.48550/arXiv.2011.04573.
[10]. Yuan H., Yu H., Wang J., Li K., Ji S. On explainability of graph neural networks via subgraph explorations. In International conference on machine learning, PMLR, 2021, pp. 12241-12252.
[11]. Zhong Z., Jiang Y., Mottin D. On the robustness of post-hoc GNN explainers to label noise. arXiv preprint, 2023 DOI: 10.48550/arXiv.2309.01706.
[12]. Fang J. et al. Evaluating post-hoc explanations for graph neural networks via robustness analysis. Advances in neural information processing systems, 2023, vol. 36, pp. 72446-72463.
[13]. Krishna S., Han T., Gu A., Wu S., Jabbari S., Lakkaraju H. The disagreement problem in explainable machine learning: A practitioner's perspective. arXiv preprint, 2022. DOI: 10.48550/arXiv.2202.01602.
[14]. Rudin C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature machine intelligence, 2019, vol. 1, no. 5, pp. 206-215.
[15]. Dai E., Wang S. Towards self-explainable graph neural network. In Proceedings of the 30th ACM international conference on information & knowledge management, 2021, pp. 302-311.
[16]. Ji Y. et al. A comprehensive survey on self-interpretable neural networks. Proceedings of the IEEE, 2025.
[17]. Zhang Z., Liu Q., Wang H., Lu C., Lee C. Protgnn: Towards self-explaining graph neural networks. In Proceedings of the AAAI conference on artificial intelligence, 2022, pp. 9127-9135.
[18]. Hastie T.J. Generalized additive models. Statistical models in S, 2017, pp. 249-307.
[19]. Samek W., Montavon G., Vedaldi A., Hansen L.K., Müller K.-R. Explainable AI: Interpreting, explaining and visualizing deep learning. Springer Nature, 2019.
[20]. Rudin C., Chen C., Chen Z., Huang H., Semenova L., Zhong C. Interpretable machine learning: Fundamental principles and 10 grand challenges. Statistic Surveys, 2022, vol. 16, pp. 1-85.

[21]. Loh W.-Y. Classification and regression trees. Wiley interdisciplinary reviews: data mining and knowledge discovery, 2011, vol. 1, no. 1, pp. 14-23.
[22]. Yang H., Rudin C., Seltzer M. Scalable bayesian rule lists. In International conference on machine learning, PMLR, 2017, pp. 3921-3930.
[23]. Kipf T.N., Welling M. Semi-supervised classification with graph convolutional networks. arXiv preprint, 2016. DOI: 10.48550/arXiv.1609.02907.
[24]. Bechler-Speicher M., Globerson A., Gilad-Bachrach R. The intelligible and effective graph neural additive network. Advances in Neural Information Processing Systems, 2024, vol. 37, pp. 90552-90578.
[25]. Bechler-Speicher M., Globerson A., Gilad-Bachrach R. Tree-g: Decision trees contesting graph neural networks. In Proceedings of the AAAI conference on artificial intelligence, 2024, pp. 11032-11042.
[26]. Jeon M., Park J., Oh H. Pl4xgl: A programming language approach to explainable graph learning. Proceedings of the ACM on Programming Languages, 2024, vol. 8, no. PLDI, pp. 2148-2173.
[27]. Geng C. et al. Beyond message passing: A symbolic alternative for expressive and interpretable graph learning. arXiv preprint, 2026. DOI: 10.48550/arXiv.2602.16947.
[28]. Caruana R., Lou Y., Gehrke J. Intelligible models for classification and regression. In Proc. 23rd ACM SIGKDD conference, 2012, Beijing, China, Citeseer.
[29]. Lou Y., Caruana R., Gehrke J., Hooker G. Accurate intelligible models with pairwise interactions. In Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining, 2013, pp. 623-631.
[30]. Nori H., Caruana R., Bu Z., Shen J.H., Kulkarni J. Accuracy, interpretability, and differential privacy via explainable boosting. In International conference on machine learning, PMLR, 2021, pp. 8227-8237.
[31]. Agarwal R. et al. Neural additive models: Interpretable machine learning with neural nets. Advances in neural information processing systems, 2021, vol. 34, pp. 4699-4711.
[32]. Breiman L., Friedman J., Olshen R.A., Stone C.J. Classification and regression trees. Chapman; Hall/CRC, 2017.
[33]. Wang F., Rudin C. Falling rule lists. In Artificial intelligence and statistics, PMLR, 2015, pp. 1013-1022.
[34]. Caruana R., Lou Y., Gehrke J., Koch P., Sturm M., Elhadad N. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, 2015, pp. 1721-1730.
[35]. Debnath A.K., Lopez de Compadre R.L., Debnath G., Shusterman A.J., Hansch C. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. Correlation with molecular orbital energies and hydrophobicity. Journal of medicinal chemistry, 1991, vol. 34, no. 2, pp. 786-797.

Информация об авторах / Information about authors

Михаил Валентинович СОЛОВЬЕВ – программист-исследователь в области машинного обучения и искусственного интеллекта. Специализируется на разработке и оптимизации графовых нейронных сетей, трансформерных графовых моделей и системах объяснимого искусственного интеллекта (XAI). Его научные интересы охватывают сферу крупных языковых моделей (LLM) и систем поиска и генерации текстов, таких как LightRAG и GraphRAG. Исследует различные методы защиты и атак на модели машинного обучения. Активно занимается повышением интерпретируемости и прозрачности моделей в контексте сложных данных.

Mikhail Valentinovich SOLOVIEV is a software engineer and researcher in the field of machine learning and artificial intelligence. He specializes in the development and optimization of graph neural networks, transformer-based graph models, and explainable artificial intelligence (XAI) systems. His research interests cover large language models (LLM) and Retrieval-Augmented Generation (RAG) systems, such as LightRAG and GraphRAG. He explores various methods for model protection and attacks on machine learning models. He is actively working on enhancing the interpretability and transparency of models in the context of complex data.

Михаил Дмитриевич ДРОБЫШЕВСКИЙ – Кандидат физико-математических наук, научный сотрудник ИСП РАН. Сфера научных интересов: доверенный искусственный интеллект, объяснимый искусственный интеллект, графовые нейронные сети.

Mikhail Dmitrievich DROBYSHEVSKIY – Cand. Sci. (Phys.-Math.), Researcher at ISP RAS. Research interests: trustworthy AI, explainable AI, graph neural networks.

Андрей Денисович Моисейкин – разработчик и исследователь в области машинного обучения и искусственного интеллекта. Сфера его научных интересов включает графовые нейронные сети, генеративные модели и крупные языковые модели. Его работа направлена на изучение современных архитектур и методов, предназначенных для анализа сложных данных, извлечения закономерностей и построения интеллектуальных систем нового поколения, а также на разработку новых графовых фундаментальных моделей для решения междоменных задач.

Andrey Denisovich MOISEYKIN is a software engineer and researcher in the field of machine learning and artificial intelligence. His research interests include graph neural networks, generative models, and large language models. His work focuses on the study of modern architectures and methods designed for analyzing complex data, discovering patterns, and building next-generation intelligent systems, as well as on the development of new graph foundation models for solving cross-domain tasks.

Евгений Юрьевич ПОЛТАВЦЕВ – инженер-разработчик, специализирующийся на параллельном программировании, алгоритмах маршрутизации и высоконагруженных системах. Его научные интересы включают оптимизацию инференса моделей компьютерного зрения и применение методов оптимизации для бэкенд-систем. Также он активно работает над разработкой эффективных алгоритмов маршрутизации в условиях ограничений предметной области.

Evgeny Yurievich POLTAVTSEV is a backend developer specializing in parallel programming, routing algorithms, and high-load systems. His research interests focus on inference optimization of computer vision models, as well as the application of optimization methods in backend systems. He is also actively involved in the development of efficient algorithms for routing in constrained environments.

Евгений Николаевич ПОТУРИЛО – инженер-исследователь в области машинного обучения и искусственного интеллекта. Специализируется на разработке и оптимизации крупномасштабных языковых и мультимодальных моделей, систем рекомендаций и инфраструктуры распределенного обучения. Сфера научных интересов включает оптимизацию инференса языковых моделей, методы адаптации (LoRA и параметр-эффективное дообучение), квантование и сжатие нейросетей, знания и вывод на графах (GraphRAG, Knowledge Graphs), а также архитектуры компьютерного зрения для мультимодальных систем. Активно занимается исследованием интеграции онтологических баз знаний с языковыми моделями и построением эффективных систем Retrieval-Augmented Generation.

Evgeny Nikolaevich POTURYLO is a research engineer in the field of Machine Learning and Artificial Intelligence. He specializes in the development and optimization of large-scale language and multimodal models, recommendation systems, and distributed training infrastructure. His research interests include language model inference optimization, parameter-efficient fine-tuning methods (such as LoRA), neural network quantization and compression, graph-based reasoning (GraphRAG, Knowledge Graphs), and computer vision architectures for multimodal AI. He is also engaged in integrating ontological knowledge bases with language models and building efficient Retrieval-Augmented Generation (RAG) systems.

Илья Сергеевич ХАЛЬКО – инженер-разработчик, специализирующийся на алгоритмах цифровой обработки связи, методах помехоустойчивого кодирования и высокопроизводительных телекоммуникационных системах. Его научные интересы

включают применение методов функционального анализа, теории аппроксимации и спектральной теории операторов для обработки сигналов в каналах связи. Особое внимание он уделяет алгебраическим и вероятностным методам синтеза эффективных алгоритмов модуляции и демодуляции, а также вопросам робастности и сходимости итерационных процедур в условиях неопределенности. Активно разрабатывает алгоритмы адаптивной фильтрации и коррекции искажений, опираясь на аппарат вариационного исчисления и теории оценивания.

Ilya Sergeevich KHALKO is a communications engineer specializing in digital signal processing algorithms, error-correcting coding, and high-performance telecommunication systems. His research interests include the application of functional analysis, approximation theory, and spectral operator theory to signal processing in communication channels. He places particular emphasis on algebraic and probabilistic methods for the synthesis of efficient modulation and demodulation algorithms, as well as on the robustness and convergence of iterative procedures under uncertainty. He is actively involved in developing adaptive filtering and distortion correction algorithms based on the calculus of variations and estimation theory.