

УДК 004.942

DOI: [10.26102/2310-6018/2026.55.4.009](https://doi.org/10.26102/2310-6018/2026.55.4.009)

Графовые нейронные сети для предсказания характеристик сетей в архитектурах New IP и ManyNets

М.К. Поваров✉, К.В. Гаврилов, П.А. Корчагин, П.А. Пищулин, С.В. Малахов

*Поволжский государственный университет телекоммуникаций и информатики,
Самара, Российская Федерация*

Резюме. В архитектурах New IP и ManyNets (ITU-T Network 2030) возрастает потребность в предсказании характеристик сетей, в том числе задержки пути, без тяжелой симуляции; неочевидно, при каких условиях графовые нейронные сети превосходят простые расчетные методы и как модели обобщаются на графы иного размера. Цель – оценить применимость графовой модели к задаче задержки пути на синтетических графах с формулой, учитывающей нагрузку на ребрах, и обобщение на графы большего размера. Применен сравнительный эксперимент на графах Эрдеша–Реньи: модель на основе графовой свертки сопоставлена с базовым методом; два эксперимента – целевая задержка с учетом нагрузки и тест на графах с 15 и 20 узлами после обучения на графах с 15 узлами. Результаты: в первом эксперименте базовый метод дал MAE 1,85 и MAPE 7,89 %, графовая модель – 9,91 и 59,20 %; во втором при переходе теста с 15 на 20 узлов MAE графовой модели снизилась примерно на 7 %, базового метода выросла на ~8 %. Сделан вывод о применимости подхода на синтетических данных как первого шага к моделям оценки характеристик сетей для архитектур New IP и ManyNets. Материалы полезны специалистам при выборе и валидации методов предсказания задержки и планировании экспериментов на синтетических топологиях.

Ключевые слова: графовые нейронные сети, предсказание характеристик сетей, New IP, ManyNets, предсказание задержки, синтетические сетевые топологии, графы Эрдеша–Реньи, качество обслуживания, топология сети, графовая свертка.

Для цитирования: Поваров М.К., Гаврилов К.В., Корчагин П.А., Пищулин П.А., Малахов С.В. Графовые нейронные сети для предсказания характеристик сетей в архитектурах New IP и ManyNets. *Моделирование, оптимизация и информационные технологии*. 2026;14(4). URL: <https://moitvvt.ru/ru/journal/article?id=2276> DOI: 10.26102/2310-6018/2026.55.4.009

Graph neural networks for predicting network characteristics in New IP and ManyNets architectures

М.К. Povarov✉, К.В. Gavrillov, P.A. Korchagin, P.A. Pishchulin, S.V. Malakhov

*Povolzhskiy State University of Telecommunications and Informatics, Samara,
the Russian Federation*

Abstract. In New IP and ManyNets architectures (ITU-T Network 2030), the need to predict network characteristics, including path delay, without heavy simulation grows; it remains unclear when graph neural networks outperform simple computational methods and how such models generalize to different graph sizes. This article aims to assess applicability of a graph neural network to the path delay task on synthetic graphs with a formula accounting for link load, and to evaluate generalization to larger graphs. A comparative experiment on Erdős–Rényi graphs was applied: a graph convolution-based model was compared with a baseline method; two experiments were conducted: a load-aware target latency experiment and a test on graphs with 15 and 20 nodes after training on graphs with 15 nodes. Results (single run): in the first experiment the baseline gave MAE 1.85 and MAPE 7.89 %, the graph model 9.91 and 59.20 %; in the second, when moving from 15- to 20-node test graphs, the graph model's MAE decreased by about 7 % and the baseline's increased by about 8 %. The approach is concluded applicable

on synthetic data as a first step toward models for predicting network characteristics in New IP and ManyNets architectures. The materials are of practical value for specialists when choosing and validating delay prediction methods and planning experiments on synthetic topologies.

Keywords: graph neural networks, network characteristics prediction, New IP, ManyNets, delay prediction, synthetic network topologies, Erdős–Rényi graphs, quality of service, network topology, graph convolution.

For citation: Povarov M.K., Gavrilov K.V., Korchagin P.A., Pishchulin P.A., Malakhov S.V. Graph neural networks for predicting network characteristics in New IP and ManyNets architectures. *Modeling, Optimization and Information Technology*. 2026;14(4). (In Russ.). URL: <https://moitvvt.ru/journal/article?id=2276> DOI: 10.26102/2310-6018/2026.55.4.009

Введение

Эволюция сетевых архитектур в сторону сетей нового поколения и гетерогенных сред повышает требования к предсказанию характеристик сетей (в том числе качества обслуживания и задержки пути) без опоры на тяжелые симуляции или измерения в реальном времени. В рамках ITU-T развиваются концепции Network 2030: New IP – протокольная рамка с гибкой адресацией и детерминированной доставкой для гетерогенных сетей; ManyNets – модель взаимодействия множества разнородных сетей, объединяемых через шлюзы. В New IP задержка критична для детерминированной доставки и выполнения SLA; в ManyNets оценка задержки на путях между сетями (в том числе через шлюзы) нужна для оркестрации и выбора маршрутов. В таких архитектурах быстрая оценка характеристик по топологии и конфигурации каналов представляет практический интерес. Цифровые двойники сетей рассматриваются как инструмент управления и оркестрации в сетях IMT-2020 и последующих поколений.

Графовые нейронные сети позволяют обучаться на графовой структуре и атрибутах узлов и ребер и обобщаться на непредъявленной при обучении топологии [1, 2]. Идея обработки графов нейросетевыми архитектурами восходит к моделям, отображающим граф и узел в векторное представление [3]; современные подходы включают схемы распространения сообщений по ребрам графа и агрегации в узлах [4], графовую свертку [2] и более выразительные архитектуры, связанные с тестом изоморфизма Вайсфейлера-Лемана [5]. В телекоммуникационной области модель RouteNet демонстрирует обобщение на произвольные топологии и конфигурации маршрутизации при предсказании задержки и потерь пакетов [1]; последующие работы показывают существенное улучшение предсказания задержки при масштабировании графовых моделей [6]. Челлендж по графовым нейросетям для сетей в рамках ITU стимулировал разработку решений для предсказания задержки по топологии, маршрутизации и матрице трафика. Модели задержки в сетях опираются на классические постановки очередей и обзоры по задержкам в сетях [7]; репозитории реальных топологий (например, Internet Topology Zoo) используются для валидации [8]. Ограничения реальных датасетов и сложность воспроизведения условий стимулируют эксперименты на синтетических графах с заданной формулой задержки пути, что дает контролируемую среду для оценки применимости графовых нейросетей к задаче предсказания задержки.

В данной работе рассматривается задача: при заданной топологии и атрибутах каналов (задержка и пропускная способность на ребрах, нагрузка на ребрах в виде числа путей через ребро) предсказывать задержку пути между парой узлов. Целевая задержка задается формулой, учитывающей как базовую задержку ребер, так и вклад нагрузки (конкуренции потоков). Цель исследования – оценить применимость графовой

нейронной сети к предсказанию такой задержки на синтетических графах и исследовать обобщение на графы большего размера. Задачи:

- а) генерация данных с формулой задержки пути, включающей нагрузку на ребрах;
- б) реализация модели на основе графовой свертки и базового метода (сумма задержек ребер по кратчайшему пути без учета нагрузки);
- в) проведение двух экспериментов – сравнение с базовым методом при учете нагрузки и тест на графах с большим числом узлов;
- г) анализ результатов и формулировка выводов.

Материалы и методы

Схема исследования. В рамках темы применения графовых нейронных сетей для предсказания характеристик сетей в архитектурах New IP и ManyNets в качестве представительной задачи выбрано предсказание задержки пути по топологии и атрибутам каналов: задержка относится к ключевым характеристикам, востребованным при детерминированной доставке (New IP), обеспечении качества обслуживания и оркестрации в гетерогенных и многосетевых сценариях, в том числе при оценке путей через шлюзы (ManyNets). Эксперимент на синтетических графах позволяет контролируемо оценить применимость графовой модели к этой задаче до переноса на реальные или симулированные топологии таких архитектур.

Исследование включало:

- а) генерацию синтетических графов;
- б) расчет для каждого графа путей и целевой задержки;
- в) преобразование данных в формат, пригодный для обучения графовой модели;
- г) обучение модели и расчет предсказаний базового метода;
- д) оценку метрик на тестовой выборке и анализ обобщения на графы другого размера.

Генерация графов и атрибутов. Использовалась модель случайных графов Эрдеша–Реньи [9]: граф с n узлами, каждое ребро присутствует независимо с вероятностью p . Размер $n = 15$ при обучении выбран как компромисс между достаточным числом путей для обучения и вычислительными затратами; вероятность ребра $p = 0,2$ обеспечивает связность графа при умеренной плотности; число графов (50) и число пар источник – сток на граф (50–100) задают пилотный объем данных для воспроизводимой оценки метода. Основные параметры приведены в Таблице 1. На каждом ребре задавались: задержка d_e (равномерное распределение в диапазоне от 1 до 20 мс), пропускная способность c_e (от 10 до 100 Мбит/с). Для каждого графа формировалось 50–100 пар узлов (источник – сток); для каждой пары строился кратчайший путь по весам d_e . По всем таким путям вычислялась нагрузка F_e на каждом ребре – число путей, проходящих через данное ребро; значение F_e сохранялось в атрибутах ребра.

Целевая задержка пути и базовый метод. Целевая задержка пути задавалась формулой:

$$D(P) = \sum_{e \in P} \left(d_e + \alpha \frac{F_e}{c_e} \right), \quad (1)$$

где P – путь (множество ребер между источником и стоком); $e \in P$ – ребро, входящее в путь; d_e – задержка на ребре e , мс; c_e – пропускная способность ребра e , Мбит/с; F_e – нагрузка на ребре (число путей, проходящих через ребро); α – безразмерный коэффициент.

Таким образом, учитывались и базовая задержка канала, и вклад нагрузки (очереди), пропорциональный отношению нагрузки к пропускной способности. Базовый метод не использовал нагрузку: кратчайший путь между парой узлов строился по весам d_e (так же, как и при расчете целевой переменной для обучающих примеров); предсказание задержки пути равнялось $\sum_{e \in P} d_e$ по этому пути. Сравнение с ним показывает, дает ли графовая модель выигрывает за счет учета нагрузки и структуры графа.

Разбиение данных и признаки. Разбиение выполнялось по графам: 70 % – обучающая выборка, 15 % – валидационная, 15 % – тестовая. Использовался фиксированный генератор случайных чисел для воспроизводимости. Признаки узлов для графовой модели включали:

- а) степень узла;
- б) средние по инцидентным ребрам значения F_e и c_e ;
- в) при наличии – среднюю нагрузку F_e по инцидентным ребрам.

Данные преобразовывались в формат графов с реберными атрибутами, совместимый с библиотекой PyTorch Geometric [10].

Модель на основе графовой свертки. Применялась архитектура на основе графовой свертки (два-три слоя свертки на графе) [2]: представления узлов обновлялись по локальной структуре и признакам соседей. Для каждого примера (путь в графе) выполнялась агрегация представлений узлов пути (например, усреднение), после чего следовал линейный слой с выходом – предсказание \hat{D} . Обучающая функция потерь – среднеквадратичная ошибка между предсказанным и целевым значением $D(P)$. Оптимизация – метод Adam. Реализация выполнена с использованием PyTorch и PyTorch Geometric [10]. Схема преобразования «граф → графовая свертка → агрегация по пути → предсказание» иллюстрируется на Рисунке 3.

Метрики и эксперименты. Качество предсказаний оценивалось средней абсолютной ошибкой (MAE) и средней абсолютной процентной ошибкой (MAPE). Для выборки из N примеров с целевыми значениями y_i и предсказаниями \hat{y}_i использовались формулы:

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|, \quad (2)$$

$$MAPE = \frac{100\%}{N} \sum_{i=1}^N \frac{|y_i - \hat{y}_i|}{|y_i|}, \quad (3)$$

где N – число примеров в выборке; y_i – целевое значение задержки для i -го примера; \hat{y}_i – предсказанное значение задержки для i -го примера.

Эксперимент 1. Целевая задержка с учетом нагрузки на ребрах. Сравнивались модель на основе графовой свертки и базовый метод по MAE и MAPE на тестовой выборке графов размера $n = 15$. Фиксировались время обучения и число параметров модели.

Эксперимент 2. Обучение проводилось только на графах с $n = 15$. Тестирование выполнялось на графах того же размера ($n = 15$) и на отдельно сгенерированных графах с $n = 20$. Цель – оценить изменение MAE и MAPE при обобщении на больший размер графа.

Реализация и воспроизводимость. Использовались Python, библиотеки NetworkX для генерации графов, NumPy, PyTorch и PyTorch Geometric для модели. Семена генератора случайных чисел (numpy, torch) фиксировались для воспроизводимости результатов. Приведенные ниже метрики соответствуют одному запуску (один фиксированный seed); для оценки разброса и устойчивости выводов целесообразно

повторять прогоны при нескольких значениях seed с приведением среднего и стандартного отклонения MAE/MARE.

Таблица 1 – Параметры генерации данных и обучения
 Table 1 – Data generation and training parameters

Параметр	Значение
Модель графа	Эрдеша–Реньи
Число узлов n (обучение)	15
Число узлов n (тест обобщения)	20
Вероятность ребра p	0,2
Число графов	50
Пар (источник – сток) на граф	50–100
Диапазон d_e , мс	[1; 20]
Диапазон c_e , Мбит/с	[10; 100]
Коэффициент α в формуле $D(P)$	2,0
Доля train/val/test	70 %/15 %/15 %
Слоёв графовой свёртки	3
Размер скрытого слоя	64
Число эпох	100
Скорость обучения (Adam)	$1 \cdot 10^{-3}$

Результаты

Эксперимент 1: сравнение при целевой задержке с учётом нагрузки. При целевой задержке пути $D(P) = \sum_{e \in P} \left(d_e + \alpha \frac{F_e}{c_e} \right)$ и базовом методе (сумма d_e по кратчайшему пути по весам d_e) на тестовой выборке получены метрики, приведенные в Таблице 2. Результаты соответствуют одному запуску с фиксированным seed.

Таблица 2 – Средняя абсолютная ошибка и средняя абсолютная процентная ошибка на тесте
 Table 2 – Mean absolute error and mean absolute percentage error on test set

Метод	MAE	MARE, %
Базовый метод	1,85	7,89
Графовая модель	9,91	59,20

Время обучения составило около 749 с; число обучаемых параметров модели – 8705. При данной настройке (объем данных, архитектура, один запуск) модель на основе графовой свертки не превзошла базовый метод по MAE и MARE.

Для иллюстрации процесса обучения на Рисунке 1 приведены кривые функции потерь на обучающей выборке и средней абсолютной ошибки на валидации по эпохам. Снижение обеих кривых к концу обучения свидетельствует о сходимости; разрыв между обучающей и валидационной кривыми дает представление о степени переобучения при заданных объеме данных и числе эпох.

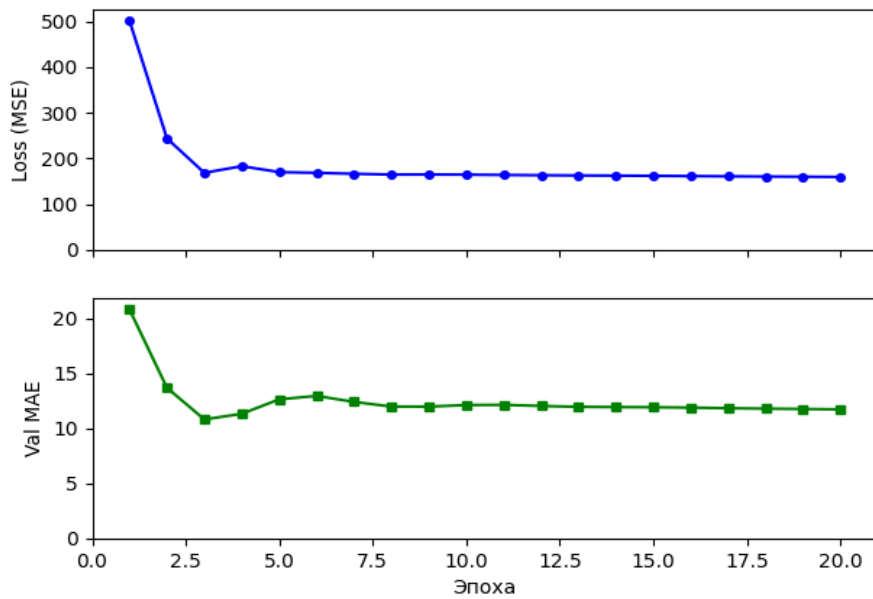


Рисунок 1 – Финальный кадр анимации обучения: функция потерь и валидационная MAE по эпохам.

Figure 1 – Final frame of training animation: loss and validation MAE over epochs.

На Рисунке 2 показан пример одного из сгенерированных графов с выделенным кратчайшим путем между выбранной парой узлов (источник – сток). Такие пути использовались как объекты предсказания: для каждого пути вычислялась целевая задержка по формуле (1), а модель и базовый метод выдавали свое предсказание для сравнения с ней (в данной визуализации путь 5 из 8).

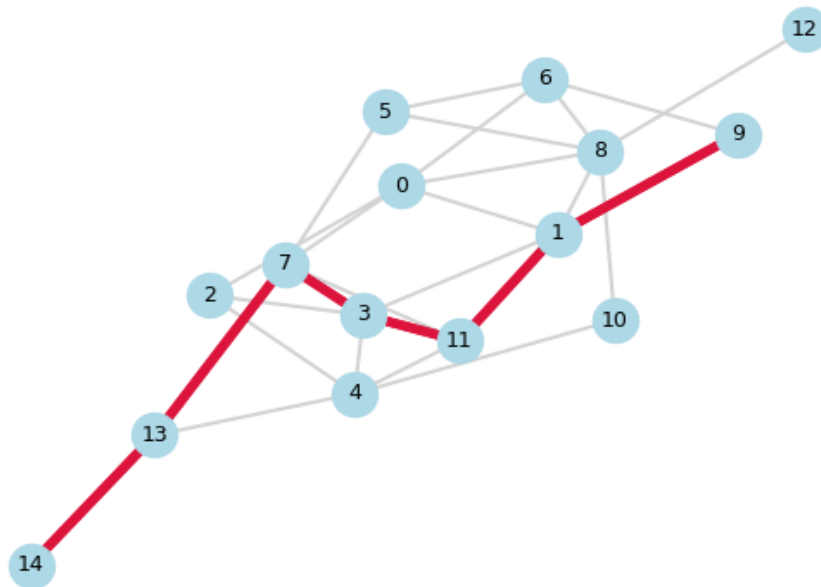


Рисунок 2 – Пример подсвеченного пути на графе – кратчайший путь между парой узлов
 Figure 1 – Example of a highlighted path on the graph: the shortest path between a node pair

Схема применяемой модели показана на Рисунке 3. Входной граф с признаками узлов и ребер последовательно обрабатывается слоями графовой свертки; для каждого примера (пути) выполняется агрегация представлений узлов пути, после чего линейный

слой выдает скалярное предсказание задержки. Это соответствует описанию в разделе «Материалы и методы».

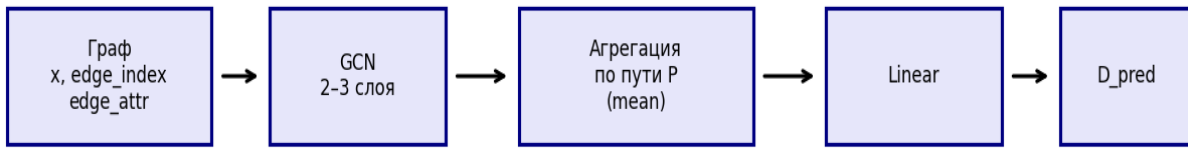


Рисунок 3 – Схема преобразования: граф с признаками узлов и ребер → слои графовой свертки → агрегация по пути → линейный слой → предсказание задержки \hat{D} .

Figure 3 – Pipeline: graph with node and edge features → graph convolution layers → path aggregation → linear layer → delay prediction \hat{D} .

На Рисунке 4 представлен пример топологии того же типа, что использовалась в экспериментах: граф с подписанными на ребрах задержкой d_e и нагрузкой F_e . По таким данным для каждой пары узлов строился кратчайший путь и вычислялась целевая задержка; признаки узлов (в том числе усредненные по инцидентным ребрам d_e, c_e, F_e) подавались на вход графовой модели.

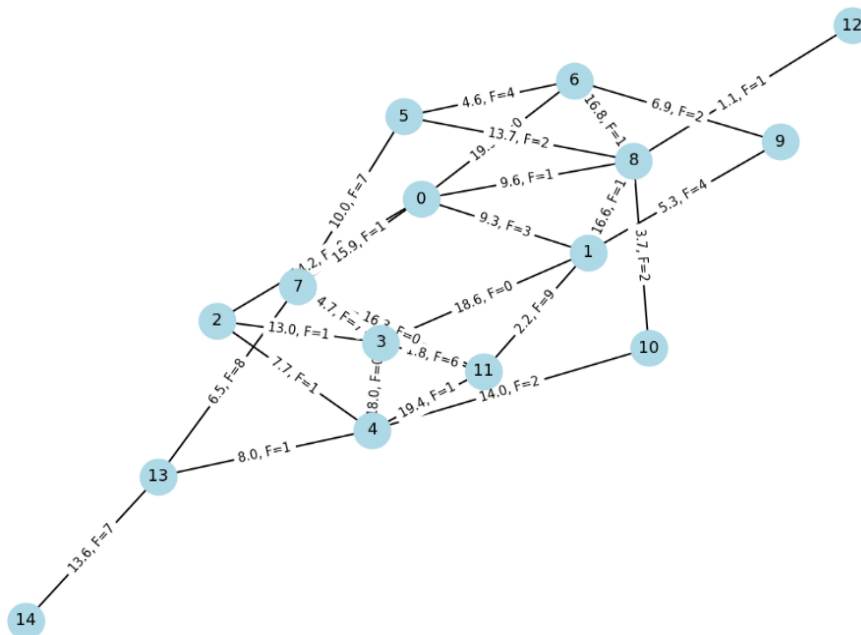


Рисунок 4 – Пример графа топологии с подписями на ребрах

Figure 4 – Example topology graph with edge labels

Эксперимент 2: обобщение на графы большего размера. Обучение выполнялось на графах с $n = 15$ узлами. В Таблицах 3 и 4 приведены результаты тестирования на графах размера $n = 15$ и $n = 20$ соответственно.

Таблица 3 – MAE и MAPE на тесте при графах размера $n = 15$

Table 3 – MAE and MAPE on test set for graphs with $n = 15$

Метод	MAE	MAPE, %
Базовый метод	1,85	7,89
Графовая модель	9,91	59,18

Таблица 4 – MAE и MAPE на тесте при графах размера $n = 20$
Table 4 – MAE and MAPE on test set for graphs with $n = 20$

Метод	MAE	MAPE, %
Базовый метод	2,01	9,99
Графовая модель	9,21	52,73

При переходе с теста на графах $n = 15$ на тест на графах $n = 20$ средняя абсолютная ошибка графовой модели уменьшилась примерно на 7 %, а средняя абсолютная ошибка базового метода увеличилась примерно на 8 %. Время обучения во втором эксперименте составило около 666 с. Сводка по обоим экспериментам (метод, MAE и MAPE на соответствующем тесте, при эксперименте 2 – размер тестовых графов) приведена в Таблице 5.

Таблица 5 – Сводка метрик по экспериментам
Table 5 – Summary of metrics across experiments

Эксперимент	Тест (n узлов)	Метод	MAE	MAPE, %
1	15	Базовый метод	1,85	7,89
1	15	Графовая модель	9,91	59,20
2	15	Базовый метод	1,85	7,89
2	15	Графовая модель	9,91	59,18
2	20	Базовый метод	2,01	9,99
2	20	Графовая модель	9,21	52,73

Обсуждение

В первом эксперименте базовый метод, не учитывающий нагрузку на ребрах, показал меньшую MAE и MAPE, чем модель на основе графовой свертки. Это может быть связано с ограниченным объемом обучающей выборки (50 графов, разбиение 70/15/15), фиксированной архитектурой и одним запуском без усреднения по нескольким семенам. Графовая модель имела доступ к признакам нагрузки по инцидентным ребрам, но при данной настройке не научилась использовать их эффективнее, чем простой учет только d_e по кратчайшему пути. В литературе по предсказанию задержки с помощью графовых нейросетей сообщается о существенном улучшении при масштабировании данных и архитектуры [4, 9]; в нашем случае эксперимент носит пилотный характер на синтетических графах ограниченного размера.

Во втором эксперименте снижение MAE графовой модели при переходе на тест с $n = 20$ по сравнению с тестом с $n = 15$ указывает на то, что на графах большего размера предсказания модели в среднем оказались ближе к целевым значениям в абсолютных единицах; рост MAE базового метода при том же переходе согласуется с изменением структуры графа и распределения путей. Улучшение графовой модели при масштабировании можно интерпретировать как способность учитывать структуру графа при обобщении на топологии иного размера; ухудшение базового метода – как отсутствие такой способности, поскольку он опирается только на сумму d_e по кратчайшему пути, чувствительную к изменению графа. Таким образом, получена количественная оценка поведения обоих методов при обобщении на непредъявленный размер графа, что релевантно для сценариев New IP и ManyNets, где топологии и размер сетей могут варьироваться.

Ограничения работы: использование только синтетических данных; одна модель топологии (Эрдеша–Реньи); фиксированный коэффициент α в формуле задержки;

отсутствие реального трафика и потерь пакетов. Результаты одного запуска без оценки дисперсии по семенам не позволяют делать строгих статистических выводов; для устойчивых выводов целесообразно повторение прогонов при нескольких значениях генератора случайных чисел с приведением среднего и стандартного отклонения.

Заключение

Задача быстрой оценки задержки по топологии и нагрузке остается востребованной в архитектурах New IP и ManyNets и в целом в сетях нового поколения: предсказание характеристик сетей (в том числе качества обслуживания) без тяжелой симуляции необходимо для управления и оркестрации. В работе мы проверили, насколько графовая нейронная сеть пригодна для такой задачи на контролируемом полигоне – синтетических графах с известной формулой задержки пути.

Итоги экспериментов оказались неоднозначными, но поучительными. Простой базовый метод – сумма задержек ребер по кратчайшему пути без учета нагрузки – в наших условиях обошел модель на основе графовой свертки по средней абсолютной и процентной ошибке. Это напоминает, что переход к нейросетевым моделям не отменяет необходимости в достаточном объеме данных и продуманной настройке. В то же время при переносе на графы большего размера (обучение на 15 узлах, тест на 20) графовая модель слегка улучшила MAE (примерно на 7 %), тогда как у базового метода ошибка выросла (примерно на 8 %). То есть модель, обученная на малых графах, показала признаки обобщения по размеру, что важно для переноса подхода на более крупные топологии.

В целом работа подтверждает, что эксперименты на синтетических графах с заданной формулой задержки полезны: они дают прозрачную среду для сравнения методов и оценки обобщения. Такой полигон логично рассматривать как первый шаг к моделям предсказания характеристик сетей для архитектур New IP и ManyNets, сетей нового поколения и цифровых двойников. Дальнейшие шаги естественно связать с валидацией на реальных или симулированных топологиях (включая открытые репозитории), с варьированием типа графа и объема данных, с повторными прогонами по семенам для устойчивых выводов, а также с учетом потерь пакетов и сценариев с шлюзами и разнородными подсетями.

СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Rusek K., Suárez-Varela J., Almasan P., Barlet-Ros P., Cabellos-Aparicio A. RouteNet: Leveraging graph neural networks for network modeling and optimization in SDN. *IEEE Journal on Selected Areas in Communications*. 2020;38(10):2260–2270. <https://doi.org/10.1109/JSAC.2020.3000405>
2. Kipf Th.N., Welling M. Semi-supervised classification with graph convolutional networks. In: *5th International Conference on Learning Representations, ICLR 2017, 24–26 April 2017, Toulon, France*. 2017. URL: <https://openreview.net/forum?id=SJU4ayYgl>
3. Scarselli F., Gori M., Tsoi A.Ch., Hagenbuchner M., Monfardini G. The graph neural network model. *IEEE Transactions on Neural Networks*. 2009;20(1):61–80. <https://doi.org/10.1109/TNN.2008.2005605>
4. Gilmer J., Schoenholz S.S., Riley P.F., Vinyals O., Dahl G.E. Neural message passing for quantum chemistry. In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, 06–11 August 2017, Sydney, NSW, Australia*. PMLR; 2017. P. 1263–1272.

5. Xu K., Hu W., Leskovec J., Jegelka S. How powerful are graph neural networks? In: *International Conference on Learning Representations, ICLR 2019, 06–09 May 2019, New Orleans, LA, USA*. 2019. URL: <https://openreview.net/forum?id=ryGs6iA5Km>
6. Farreras M., Soto P., Camelo M., Fàbrega L., Vilà P. Improving network delay predictions using GNNs. *Journal of Network and Systems Management*. 2023;31(4). <https://doi.org/10.1007/s10922-023-09758-9>
7. Roy A., Pachuau J.L., Saha A.K. An overview of queuing delay and various delay based algorithms in networks. *Computing*. 2021;103(10):2361–2399. <https://doi.org/10.1007/s00607-021-00973-3>
8. Knight S., Nguyen H.X., Falkner N., Bowden Rh., Roughan M. The internet topology zoo. *IEEE Journal on Selected Areas in Communications*. 2011;29(9):1765–1775. <https://doi.org/10.1109/JSAC.2011.111002>
9. Erdos P., Renyi A. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*. 1960;5:17–61.
10. Fey M., Lenssen J.E. *Fast graph representation learning with PyTorch Geometric*. arXiv. URL: <https://arxiv.org/abs/1903.02428> [Accessed 5th March 2025].

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Поваров Максим Константинович, студент, Поволжский государственный университет телекоммуникаций и информатики, Самара, Российская Федерация.
e-mail: maxim.powarov@mail.ru

Maksim K. Povarov, Student, Povolzhskiy State University of Telecommunications and Informatics, Samara, the Russian Federation.

Гаврилов Кирилл Витальевич, студент, Поволжский государственный университет телекоммуникаций и информатики, Самара, Российская Федерация.
e-mail: clickhonk@gmail.com

Kirill V. Gavrilov, Student, Povolzhskiy State University of Telecommunications and Informatics, Samara, the Russian Federation.

Корчагин Павел Алексеевич, студент, Поволжский государственный университет телекоммуникаций и информатики, Самара, Российская Федерация.
e-mail: pasha.korchagin.03@mail.ru

Pavel A. Korchagin, Student, Povolzhskiy State University of Telecommunications and Informatics, Samara, the Russian Federation.

Пищулин Павел Александрович, студент, Поволжский государственный университет телекоммуникаций и информатики, Самара, Российская Федерация.
e-mail: pavelalex485@mail.ru

Pavel A. Pishchulin, Student, Povolzhskiy State University of Telecommunications and Informatics, Samara, the Russian Federation.

Малахов Сергей Валерьевич, кандидат технических наук, доцент кафедры информатики и робототехнических систем, Поволжский государственный университет телекоммуникаций и информатики, Самара, Российская Федерация.
e-mail: s.malakhov@psuti.ru

Sergey V. Malakhov, Candidate of Engineering Sciences, Associate Professor at the the Department of Informatics and Robotic Systems, Povolzhskiy State University of Telecommunications and Informatics, Samara, the Russian Federation.

*Статья поступила в редакцию 09.03.2026; одобрена после рецензирования 16.04.2026;
принята к публикации 24.04.2026.*

*The article was submitted 09.03.2026; approved after reviewing 16.04.2026;
accepted for publication 24.04.2026.*