



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

## РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

**НА ТЕМУ:**

**Классификация данных Википедии с использованием  
графовых нейронных сетей**

Студент ИУ5-34М  
(Группа)

\_\_\_\_\_. Л. В. Перлин  
(Подпись, дата) (И.О.Фамилия)

Руководитель

\_\_\_\_\_. Ю.Е. Гапанюк  
(Подпись, дата) (И.О.Фамилия)

2023 г.

**Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

---

УТВЕРЖДАЮ  
Заведующий кафедрой ИУ5  
(Индекс)  
В.И. Терехов  
(И.О.Фамилия)  
« 04 » сентября 2023 г.

**З А Д А Н И Е  
на выполнение научно-исследовательской работы**

по теме \_\_\_\_\_

Студент группы ИУ5-34М

Перлин Леонид Вадимович  
(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.)

ИССЛЕДОВАТЕЛЬСКАЯ

Источник тематики (кафедра, предприятие, НИР) КАФЕДРА

График выполнения НИР: 25% к \_\_\_\_ нед., 50% к \_\_\_\_ нед., 75% к \_\_\_\_ нед., 100% к \_\_\_\_ нед.

***Техническое задание***

проанализировать различные виды графовых нейронных сетей и разработать нейронную сеть для  
решения задачи классификации сетевых данных  
википедии

***Оформление научно-исследовательской работы:***

Расчетно-пояснительная записка на 20 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

Дата выдачи задания « 04 » сентября 2023 г.

**Руководитель НИР**

Ю.Е. Гапанюк  
(Подпись, дата) (И.О.Фамилия)

**Студент**

Л.В. Перлин  
(Подпись, дата) (И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

## Оглавление

<b>Оглавление .....</b>	<b>3</b>
<b>Глоссарий .....</b>	<b>4</b>
<b>Введение .....</b>	<b>5</b>
<b>Трудности использования графа как входных данных .....</b>	<b>8</b>
<b>Глобальные представления .....</b>	<b>9</b>
<b>Архитектура графовых нейронных сетей .....</b>	<b>10</b>
Концепция передачи сообщений .....	11
Сети внимания графов .....	12
<b>Задачи графовых нейронных сетей .....</b>	<b>15</b>
Задача на уровне графа .....	15
Задача на уровне узла .....	15
Задача на уровне связей .....	16
<b>Использование графовой нейронной сети для классификации статей из википедии. ....</b>	<b>17</b>
Формат данных википедии .....	17
Создание набора данных .....	18
Разработка нейронной сети .....	19
<b>Заключение .....</b>	<b>20</b>
<b>Список литературы .....</b>	<b>21</b>

## *Глоссарий*

Датасет – набор данных, коллекция данных

GNN (англ. Graph neural network) – Графовая нейронная сеть

GCN (англ. Graph convolution network) – Графовая сверточная сеть

GAT (англ. Graph attention network) – Графовая нейронная сеть с использованием механизмов внимания

## ***Введение***

Нейронные сети стали неотъемлемой частью жизни современных людей. Они используются в самых разных областях, от распознавания изображений и речи до машинного перевода и игры в шахматы. Нейронные сети для решения этих задач используют различные форматы входных данных, включая числовые данные, текстовые данные, изображения, видео, аудио, временные ряды. Недавно нейросети стали использоваться для обработки данных в формате графов. Граф представляет собой набор узлов, соединенных между собой ребрами. Узлы могут представлять собой объекты, такие как люди, места или вещи, а ребра могут представлять собой отношения между этими объектами. Многие данные реального мира описываются данной структурой лучше всего.

Обработка данных в формате графов имеет ряд особенностей, которые отличают ее от обработки данных в других форматах. Графы могут быть очень большими и сложными, графы могут содержать циклы, что затрудняет их анализ, графы могут содержать разнородные данные, что требует от нейросети особенной способности к обобщению, может существовать несколько различных представлений одного графа, что создает проблему с инвариантностью входных данных. Каждая задача требует особенного набора данных. Наборы данных для графовых нейронных сетей могут сильно отличаться в размерах, например: датасет с набором данных из википедии, представляет собой 1 граф с 12 миллионами узлов и 378 миллионами ребер, когда в свою очередь датасет с малыми молекулами содержит 134 тысячи графов с максимальным количеством в 9 узлов и 26 ребер для одного графа. В данной ситуации некоторые алгоритмы могут быть применимы к небольшому графу, но при этом абсолютно неэффективны в большом графе. В некоторых случаях необходимо помимо самих данных графа хранить общий контекст на весь граф или общее свойство всего графа. Несмотря на это, современные нейросети уже более 10 лет успешно обрабатывают графы. Появилось практическое применение в таких областях, как антибактериальное обнаружение, физическое моделирование, обнаружение поддельных

новостей, системы прогнозирования и рекомендаций трафика. Например, антибиотик халицин [1], был найден с помощью графовых нейронных сетей.

Графовая нейронная сеть принадлежит к классу искусственных нейронных сетей, предназначенных для обработки данных, представленных в виде графов. [2] Она является важной частью области "геометрического глубокого обучения", где существующие архитектуры нейронных сетей могут быть поняты как GNN, работающие с соответствующе определенными графами.

В более общем контексте "геометрического глубокого обучения" определенные существующие архитектуры нейронных сетей можно интерпретировать как GNN, действующие на соответственно определенных графах. Например, сверточный слой нейронной сети, в контексте компьютерного зрения, может рассматриваться как GNN, примененный к графам, у которых узлы представляют пиксели, и только соседние пиксели связаны ребрами в графе. Один из способов визуализации связности графа - это его матрица смежности. Мы упорядочиваем узлы, в данном случае каждый из 25 пикселей в простом изображении размером 5x5 пикселей в виде смайлика, и заполняем матрицу записью, если два узла имеют общее ребро. Данный пример представлен на рисунке 1 ниже.

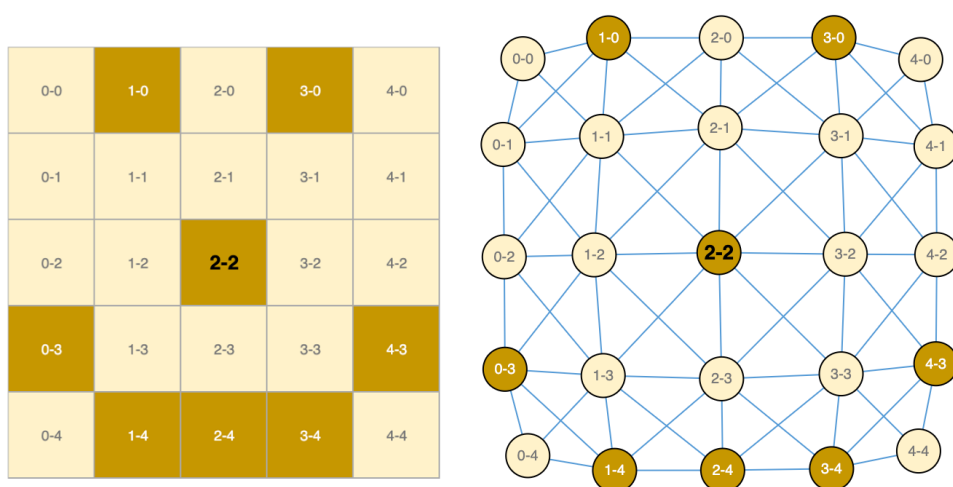


Рисунок 1. Представление растрового изображения в виде графа

Слой трансформера в обработке естественного языка может быть воспринят как GNN, примененный к полным графам, у которых узлы являются словами или токенами в тексте естественного языка.

Ключным элементом дизайна GNN является использование парной передачи сообщений [3], благодаря чему узлы графа итеративно обновляют свои представления, обмениваясь информацией со своими соседями. С момента их появления было предложено несколько различных архитектур GNN, которые реализуют различные варианты передачи сообщений, начиная с рекурсивных или конструктивных сверточных подходов. К настоящему времени открытым исследовательским вопросом является, возможно ли определить архитектуры GNN, "выходящие за пределы" передачи сообщений, или каждая GNN может быть построена на передаче сообщений по соответствующе определенным графам.

Задачей данной научно-исследовательской работы является рассмотреть особенности графов в рамках графовых нейронных сетей и варианты практического их применения на примере классификации данных из Википедии.

## *Трудности использования графа как входных данных*

Модели машинного обучения обычно работают с прямоугольными или сетчатыми массивами данных. Изначально неочевидно, как корректно преобразовать графы в формат, удобный для глубокого обучения. Графы содержат четыре типа информации, которые могут быть полезны при прогнозировании: узлы, ребра, глобальный контекст и связи. Первые три типа относительно легко представить: например, для узлов мы можем создать матрицу характеристик узлов  $N$ , присваивая каждому узлу индекс  $i$  и храня характеристики узла  $i$  в  $N$ . Несмотря на то, что эти матрицы содержат разное количество примеров, их можно обрабатывать без специальных методов.

Однако представление связности в графе представляет собой более сложную задачу. Возможным вариантом может быть использование матрицы смежности, так как она легко преобразуется в тензор. Однако у такого подхода есть недостатки. Число узлов в графе, как уже указывалось ранее, может достигать миллионов, и количество рёбер на узел может значительно различаться. Это часто приводит к созданию очень разреженных матриц смежности, которые неэффективно занимают значительное количество памяти.

Еще одна проблема заключается в том, что существует множество различных матриц смежности, способных кодировать одну и ту же связность. Нет гарантии, что эти разные матрицы приведут к одинаковым результатам при использовании глубоких нейронных сетей (то есть они не обладают инвариантностью к перестановкам). Некоторые графы можно эквивалентно описать с помощью разных матриц смежности. Они также могут быть описаны при любой другой возможной перестановке узлов.

Один из элегантных и эффективных способов представления разреженных матриц, который экономичен по использованию памяти, — это использование списков смежности. Эти списки описывают связь между узлами  $n_i$  и  $n_j$  через ребро  $ek$  в виде кортежа  $(i, j)$ , хранящегося в  $k$ -ой записи списка смежности. Такой подход позволяет избежать лишних вычислений и хранения



информации о несвязанных частях графа, учитывая, что количество рёбер ожидаемо будет значительно меньше, чем количество записей в матрице смежности [4].

### *Глобальные представления*

Узлы, удаленные друг от друга в графе, могут оказаться в неэффективной связи при передаче информации, даже если мы применим несколько итераций передачи сообщений. Для каждого узла с  $k$  слоями информация может распространяться только на  $k$  шагов вокруг него. Это может представлять проблему в случаях, когда задача зависит от узлов или групп узлов, находящихся на большом расстоянии друг от друга. Одним из возможных решений является предоставление всем узлам возможности обмениваться информацией друг с другом. Однако для больших графов это становится вычислительно сложной задачей и может применяться только для небольших графов, таких как графы молекул.

Для решения этой проблемы можно использовать глобальное представление графа, также известное как главный узел [5]. Этот глобальный вектор контекста связывается со всеми остальными узлами и ребрами в сети, играя роль моста для передачи информации между ними и обеспечивая комплексное представление всего графа. Это позволяет создать более точное и информативное представление графа, чем это возможно иными способами.

В этой ситуации, когда все атрибуты графа имеют изученные представления, мы можем использовать их при объединении. Мы можем условно использовать информацию о выбранном атрибуте относительно других, используя данные из соседних узлов, ребер и глобальные данные. Чтобы объединить информацию из всех этих источников для встраивания нового узла, мы можем просто скомбинировать их. Также мы можем сопоставить их с одним общим пространством при помощи линейной проекции и применить слой объектной модуляции, который можно рассматривать как механизм внимания по характеристикам.

## *Архитектура графовых нейронных сетей*

Архитектура общей графовой нейронной сети включает следующие основные слои [6]:

Перестановочная эквивариантность: Слой перестановочной эквивариантности отображает представление графа в обновленное представление того же графа. В литературе перестановочные слои реализуются через попарный обмен сообщениями между узлами графа. В понимании слоя передачи сообщений узлы обновляют свои представления, агрегируя сообщения, полученные от своих ближайших соседей. Таким образом, каждый слой передачи сообщений увеличивает зону восприятия GNN на один шаг.

Локальный пулинг [7]: Локальный слой пулинга уменьшает размер графа через субдискретизацию. Локальный пулинг используется для увеличения зоны восприятия GNN, аналогично слоям пулинга в сверточных нейронных сетях. Примеры включают пулинг k-ближайших соседей, топ-k пулинг и пулинг на основе само-внимания.

Глобальный пулинг: Глобальный пулинг, также известный как слой считывания, предоставляет фиксированное представление всего графа. Глобальный слой пулинга должен быть перестановочно-инвариантным, так что перестановки в порядке узлов и ребер графа не изменяют окончательный выход данного слоя [8]. Примеры включают поэлементную сумму, среднее или максимальное значение.

## Концепция передачи сообщений

Идея сетей передачи сообщений была представлена в 2017 году, и она сводится к трем основным этапам:

Каждый узел в графе вычисляет сообщение для каждого из своих соседей. Сообщения являются функцией узла, соседа и ребра между ними. Сообщения отправляются, и каждый узел агрегирует полученные сообщения, используя перестановочно-инвариантную функцию. Эта функция обычно представляет собой сумму или среднее, но может быть любой. Получив сообщения, каждый узел обновляет свои атрибуты как функцию его текущих атрибутов и агрегированных сообщений. Эта процедура происходит синхронно для всех узлов в графе, так что на каждом шаге передачи сообщений обновляются все узлы.

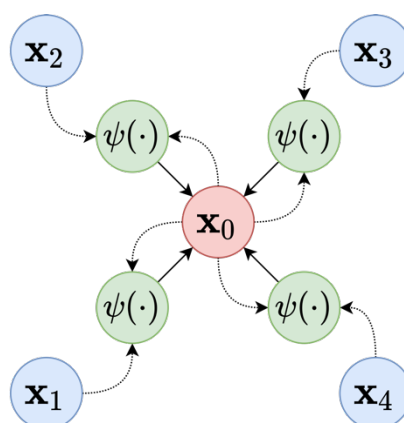


Рисунок 2. Передача сообщений

На рисунке 2 представлено обновление представления узла на уровне нейронной сети передачи сообщений. Узел  $x_0$  получает сообщения, отправленные всеми его непосредственными соседями  $x_1$  к  $x_4$ . Сообщения вычисляются с помощью функции сообщений  $\psi$ , которая учитывает особенности как отправителей, так и получателей.

## Сети внимания графов

Учитывая исследования, в том числе [9], GAT демонстрируют максимальную точность при решении задач классификации, поэтому заслуживают особенного внимания. Сети внимания графов или GAT представляют собой вариант графовых нейронных сетей, использующих механизмы внимания для обучения признаков на графах. Представленные в 2018 году, GAT предлагают более тонкий подход к агрегации информации о соседних узлах по сравнению с традиционными GNN.

В стандартных GNN, таких как графовые сверточные сети (Graph Convolutional Networks, GCN), обновление признаков узла обычно является средним значением признаков его соседей. Этот подход не различает вклады разных соседей.

GAT, напротив, назначают коэффициент внимания каждому соседу, указывая на важность признаков этого соседа для обновления признаков узла. Эти коэффициенты вычисляются с помощью общего механизма самовнимания, который рассчитывает оценку внимания для каждой пары узлов. Затем оценки нормализуются по всему окружению каждого узла с использованием функции SoftMax.

Подход, основанный на внимании, позволяет GAT назначать разные веса различным соседям, обеспечивая более гибкую и потенциально более выразительную модель. Он также обладает уровнем интерпретируемости, поскольку коэффициенты внимания можно рассматривать как показатель важности каждого соседа [10].

В графовых сетях внимания концепция использования нескольких "голов" внимания аналогична идее использования нескольких фильтров в сверточных нейронных сетях. Каждая "голова" внимания может потенциально научиться обращать внимание на различные типы информации об окружении узлов.

При использовании нескольких "голов" внимания слой GAT модифицируется следующим образом:

1. Линейное преобразование: каждая "голова" внимания  $k$  имеет свою собственную матрицу весов  $W^k$ . Входные признаки преобразуются отдельно для каждой "головы", что показано на формуле 1:

$$h_i^k = x_i W^k \quad (1)$$

2. Механизм самовнимания: каждая "голова" внимания вычисляет собственные коэффициенты внимания, что показано на формуле 2:

$$e_{im}^k = \text{LeakyReLU}((a^k)^T [h_i^k || h_m^k]) \quad (2)$$

3. Нормализация: Коэффициенты внимания нормализуются отдельно для каждой "головы", что показано на формуле 3:

$$\alpha_{im}^k = \text{SoftMax}_m(e_{im}^k) \quad (3)$$

4. Агрегация признаков: Выходные признаки вычисляются как взвешенная сумма признаков соседей, отдельно для каждой "головы", что показано на формуле 4:

$$h_i'^k = \sigma \left( \sum_k \alpha_{im}^k h_m^k \right) \quad (4)$$

5. Агрегация из нескольких "голов": Выходные признаки из всех "голов" затем агрегируют для формирования окончательных выходных признаков. Это можно сделать двумя способами, конкатенацией и с помощью среднего значения, что представлено на формулах 5 и 6:

$$h_i' = || h_i'^k \quad (5)$$

$$h_i' = 1/K \sum_i^k h_i'^k \quad (6)$$

## *Задачи графовых нейронных сетей*

### Задача на уровне графа

Задача на уровне графа состоит в том, чтобы спрогнозировать свойства всего графа или его фрагмента. В контексте данных из Википедии это задание может быть интерпретировано как кластеризация узлов в графе. Другими словами, сеть глубокого обучения может использовать информацию о свойствах узлов, например, текстовое содержимое статей и связях между ними, например, ссылки между статьями для прогнозирования общей тематики или концепции, объединяющей группу статей.

Это означает, что путем анализа данных, представляющих граф статей Википедии, мы можем обучить нейронную сеть для выявления паттернов и связей между узлами (статьями) в графе. На выходе мы можем получить прогнозы или кластеры, которые объединяют статьи по общей тематике или концепции, что поможет в понимании и структурировании информации из Википедии.

### Задача на уровне узла

Задача на уровне узла в графовых структурах заключается в предсказании свойств конкретного узла в графе. Если мы рассматриваем Википедию как графовую структуру, где статьи представлены как узлы, то задача на уровне узла может заключаться в классификации этих статей на основе их содержания или свойств.

В контексте Википедии статьи, которые выступают в качестве узлов в графе, могут быть классифицированы по различным тематикам, таким как научные, технические, исторические или художественные. Это означает, что мы можем использовать информацию, содержащуюся в статьях, чтобы предсказать или присвоить им различные метки или категории, отражающие их тематическое содержание или природу.

Применение методов машинного обучения, таких как нейронные сети, к графовым данным Википедии позволяет выявлять общие характеристики или темы, присущие группам статей. Это помогает в понимании структуры и содержания Википедии, делая ее более удобной для анализа и поиска информации по различным категориям и тематикам.

#### Задача на уровне связей

Задача на уровне связей в графовых структурах заключается в прогнозировании или анализе связей между узлами. Графовые нейронные сети в этом случае могут выявлять существующие связи в графе и их характеристики. Это позволяет понять, есть ли какое-либо взаимодействие или отношение между объектами, а также определить характер этой связи.

Применительно к Википедии, задача на уровне связей может касаться статей, которые имеют смысловую или тематическую связь друг с другом. Графовые нейронные сети могут анализировать тексты статей, определять схожесть или взаимосвязь между ними на основе содержания или контекста. Например, они могут выявлять статьи, которые имеют схожую тематику или содержат информацию о схожих событиях, локациях или людях. Это позволяет увидеть связи между различными статьями и лучше понять их смысловое взаимодействие в рамках графа Википедии.



## *Использование графовой нейронной сети для классификации статей из википедии.*

### Формат данных википедии

Структура данных Википедии может быть представлена как сетевая структура, где каждая статья является узлом сетевой структуры, а ссылки между статьями образуют связи. В Википедии есть главная страница, которая является корневым узлом. От корневого узла выходят новые звенья, представляющие категории и порталы, и каждая категория может содержать подкатегории. Статьи могут быть отнесены к одной или нескольким категориям. Каждая статья может также содержать ссылки на другие статьи Википедии, образуя тем самым гибкую и динамичную сетевую структуру. Эти ссылки могут быть как внутренними, то есть ссылками на другие статьи внутри Википедии, так и внешними, то есть ссылки на статьи вне Википедии. В рамках исследования рассматриваются исключительно внутренние ссылки.

В силу децентрализованного характера структура Википедии является довольно сложной и динамичной. Целью является построение набора данных, которая наиболее эффективно сможет динамично моделировать карту сайта на примере русскоязычных статей из Википедии и отвечать требованиям необходимых запросов для анализа.

## Создание набора данных

Процесс сбора данных из Википедии с использованием библиотеки `wikipediaapi` [11] и `NetworkX` [12] для создания графа осуществляется следующим образом:

1. Начиная с заданной статьи, он переходит по ссылкам на другие статьи в заданной глубине.
2. Извлекаются статьи и их ссылки для построения графа, представляющего связи между статьями.
3. Обработка ошибок, таких как отсутствие статьи или возникновение исключений при получении данных.
4. Далее графовые данные импортируются в графовую СУБД `Neo4j` [13] для хранения. Так как процесс сбора данных занимает большое количество времени, это позволяет хранить определенные контрольные точки. Далее при непосредственном создании тестового и обучающего набора данных данные берутся непосредственно из `Neo4j`.

## Разработка нейронной сети

Для разработки необходимой нейронной сети используется специальная библиотека Torch Geometric [14]. Torch Geometric это библиотека для графовых нейронных сетей, разработанная на основе библиотеки глубокого обучения PyTorch. Она предоставляет инструменты для работы с графами, обработки данных и реализации различных моделей графовых нейронных сетей. Вот несколько ключевых особенностей torch\_geometric, таких как поддержка графовых структур, библиотека предоставляет инструменты для работы с различными типами графов и структур, такими как направленные и ненаправленные графы, мультиграфы и гиперграфы, есть поддержка графовых сверточных сетей, предоставляются различные типы графовых сверточных слоев (Graph Convolutional Layers) для анализа графовых данных. Включает в себя GATConv (Graph Attention Network), GCNConv (Graph Convolutional Network), SAGEConv (GraphSAGE), GINConv (Graph Isomorphism Network), и другие. Также есть интеграция с PyTorch: поскольку torch\_geometric построен поверх PyTorch, его модули и функции легко интегрируются и работают с синтаксисом PyTorch. torch\_geometric активно используется в исследованиях и разработке графовых моделей машинного обучения и глубокого обучения.

В разработанной нейросети используется слой GATConv [15] с гиперпараметром количества “голов”, принцип работы которого был описан выше. Для регуляризации обучения и повышения обобщающей способности итоговой GNN используется dropout. Выходным слоем является вероятность принадлежности определенной статьи к одному из четырех возможных классов - научная, техническая, историческая или художественная статья. Итоговая точность модели на тестовом наборе данных составляет 80,02%

## *Заключение*

Концепция графовых нейронных сетей (GNN) представляет собой фундаментальный подход в сфере обработки данных, ориентированных на графы. Эти нейронные сети предоставляют эффективные инструменты для анализа и извлечения информации из различных видов данных, представленных в виде графовой структуры.

В данной работе мы более подробно рассмотрели принципы работы графовых нейронных сетей, а также выявили их уникальные особенности в контексте обработки графовых данных. Мы изучили различные способы представления данных в виде графов и разобрали разнообразные задачи, которые можно успешно решать с помощью GNN.

Особое внимание было уделено сверточным графовым сетям (GCN) и графовым сетям, построенным на основе механизмов внимания (GAT), позволяющим эффективно учесть связи между узлами графа. Мы исследовали принципы работы этих моделей и их применимость к различным задачам анализа графов.

В качестве практической демонстрации применения GNN был собран датасет, основанный на данных русскоязычной Википедии. Это позволило нам провести эксперименты и тестирование графовой нейронной сети на реальных данных с целью анализа, классификации или предсказания свойств графа, что имеет большое значение во многих областях, от социальных сетей до биоинформатики.

Графовые нейронные сети являются важным инструментом в машинном обучении, поскольку они позволяют эффективно учитывать структуру данных, организованных в виде графов, и использовать их для решения различных задач анализа и предсказания. В данной работе мы попытались охватить ключевые аспекты и потенциал графовых нейронных сетей в обработке графовых данных.

## *Список литературы*

1. MIT News. "Artificial intelligence identifies new antibiotic." [Электронный ресурс]. URL: <https://news.mit.edu/2020/artificial-intelligence-identifies-new-antibiotic-0220> (дата обращения: 11.11.2023)
2. Distill. "Graph Neural Networks: A Review of Methods and Applications." [Электронный ресурс]. URL: <https://distill.pub/2021/gnn-intro/> (дата обращения: 11.11.2023)
3. Weights & Biases. "An Introduction to Message Passing Graph Neural Networks." [Электронный ресурс]. URL: <https://wandb.ai/graph-neural-networks/spatial/reports/An-Introduction-to-Message-Passing-Graph-Neural-Networks--VmlldzoYMDI2NTg2> (дата обращения: 11.11.2023)
4. ResearchGate. "Data Preprocessing and Graph Neural Networks (GNNs) Design." [Электронный ресурс]. URL: [https://www.researchgate.net/figure/Data-Preprocessing-and-Graph-Neural-Networks-GNNs-Design\\_fig1\\_370213107](https://www.researchgate.net/figure/Data-Preprocessing-and-Graph-Neural-Networks-GNNs-Design_fig1_370213107) (дата обращения: 11.11.2023)
5. arXiv. "A Review of Graph Neural Networks: Models and Applications." [Электронный ресурс]. URL: <https://arxiv.org/abs/2106.05081> (дата обращения: 11.11.2023)
6. arXiv. "Graph Neural Networks: A Review of Methods and Applications." [Электронный ресурс]. URL: <https://arxiv.org/abs/2104.13478> (дата обращения: 11.11.2023)
7. arXiv. "Graph Attention Transformers." [Электронный ресурс]. URL: <https://arxiv.org/abs/2202.11097> (дата обращения: 10.12.2023)
8. arXiv. "Edge Prediction and Visualization in Graphs with Graph Attention Transformers." [Электронный ресурс]. URL: <https://arxiv.org/abs/2204.07321> (дата обращения: 10.12.2023)

9. arXiv. "Semi-Supervised Classification with Graph Convolutional Networks." [Электронный ресурс]. URL: <https://arxiv.org/abs/1710.10903> (дата обращения: 10.12.2023)
10. Medium. "Understanding Graph Attention Networks: A Practical Exploration." [Электронный ресурс]. URL: <https://medium.com/@farzad.karami/understanding-graph-attention-networks-a-practical-exploration> (дата обращения: 11.12.2023)
11. Wikipedia-API. "Python Wikipedia API." [Электронный ресурс]. URL: <https://pypi.org/project/Wikipedia-API/> (дата обращения: 11.12.2023)
12. NetworkX. "NetworkX Documentation." [Электронный ресурс]. URL: <https://networkx.org/>
13. Neo4j. "Neo4j Graph Database." [Электронный ресурс]. URL: <https://neo4j.com/> (дата обращения: 11.12.2023)
14. PyTorch Geometric. "PyTorch Geometric Documentation." [Электронный ресурс]. URL: <https://pytorch-geometric.readthedocs.io/en/latest/> (дата обращения: 11.12.2023)
15. PyTorch Geometric. "GATConv Documentation." [Электронный ресурс]. URL: [https://pytorch-geometric.readthedocs.io/en/latest/generated/torch\\_geometric.nn.conv.GATConv.html](https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.GATConv.html) (дата обращения: 11.12.2023)