УДК 004.891

**Анализ влияния характеристик графа в задаче предсказания новых зависимостей**

**Мосалов Олег Петрович,** кандидат физико – математических наук, заведующий лабораторией интеллектуальных систем управления,

**Ефимова Анастасия Олеговна,** студент 3 курса кафедры информационно – коммуникационных систем и технологий,

**Соколова Дарья Александровна,** студент 3 курса кафедры информационно – коммуникационных систем и технологий,

Государственное бюджетное образовательное учреждение высшего образования Московской области «Технологический университет имени дважды Героя Советского Союза, лётчика – космонавта А.А. Леонова», г. Королёв, Московская область.

*В работе рассматривается задача предсказания новых зависимостей между сущностями предметной области на основе информации, извлечённой из плохо структурированных данных. Предложены структура хранения таких данных в виде графа и алгоритм поиска новых сущностей на основе значений ряда характеристик такого графа. Приведены описание вычислительных экспериментов и обсуждение их результатов. Сформулированы требования к единой метрике оценки эффективности таких алгоритмов, приведены результаты анализа существующих метрик.*

Граф онтологии, поиск новых связей в графе, метрика эффективности алгоритма, анализ данных.

**Analysis of influence of graph characteristics in the task of link prediction**

**Mosalov Oleg Petrovich,** PhD*.*, Head of the Laboratory of Intelligent Control Systems,

**Efimova Anastasya Olegovna,** 3rd year student of the Department of Information and Communication Systems and Technologies,

**Sokolova Daria Aleksandrovna,** 3rd year student of the Department of Information and Communication Systems and Technologies,

State educational institution of higher education Moscow region “Leonov’s University of technology”, Korolev, Moscow region.

*In the article a task of link prediction in a subject area, that is based on the information extracted from improperly structured data, is considered. A structure to store such data in a form of a graph and an algorithm for link prediction based on the characteristics of the graph are suggested. The computational experiments are described and their results are discussed. Requirements for a unified metric for such algorithms are defined, and the results of existing metrics analysis are given.*

Ontology graph, link prediction, algorithm performance metric, data mining.

Анализ данных является одним из приоритетных направлений компаний для их успешного развития и получения дополнительных средств путем привлечения новых потребителей, но не все имеющиеся данные возможно представить в виде, который подходит для машинного анализа. Анализ данных – это своего рода процесс извлечения знаний из экспериментальных данных или же процесс фильтрации, исследования. Большинство компаний сталкиваются с такой проблемой, как отсутствие готовых инструментов и экспертизы для работы с плохо структурированными данными [1].

В данной статье предлагается подход для решения данной проблемы, который позволяет улучшить качество работы при обработке плохо структурированных данных на промышленных предприятиях.

**Формат хранения обработанных данных.** Существует огромное количество способов хранения и обработки информации, подходящих для различных задачи и имеющих свои плюсы и минусы.

В данной работе рассматривается графовая структура данных [2, 3], так как она реализует иерархию, что соответствует экспертным представлениям о рассматриваемых данных. Эта структура данных позволяет быстро и эффективно получать доступ к атрибутам сущностей и связям между ними, по сравнению с иными структурами данных, а также позволяет эффективно визуализировать полученные данные [8, 9, 18].

В данной работе в качестве источника данных был использован массив документов, предоставленных компанией-партнёром. Документы имели различный формат: doc, docx, pdf, ppt и т.п., а также различный тип: презентация с конференции, письмо, запрос на коммерческое предложение, благодарность, чертёж и другие.

Иерархия рассматриваемой структуры данных состоит из пяти слоёв, а именно: «способности», «компании», «интересы», «технологии» и «материалы». Информация сохраняется в данной структуре в виде точек, соответствующих отдельным сущностям, и в виде зависимостей между ними. Ключевой слой – это «компании», именно от него идут зависимости к другим слоям и сущностям. При этом считается, что между двумя сущностями есть зависимость, если обе эти сущности упомянуты в одном и том же документе.

Для извлечения информации из массива документов и представления её в виде описанной выше структуры данных было использовано два дополняющих друг друга подхода:

1. Анализ документов и извлечение данных вручную.
2. Получение данных с помощью алгоритмов извлечения именованных сущностей.

Первый подход даёт очень хорошие точность и качество извлечения данных, но требует большого количества ресурсов как на подготовительном этапе – когда определяются общие правила извлечения данных, так и на основном этапе – когда данные непосредственно извлекаются.

Второй подход гораздо эффективнее и быстрее в части процесса извлечения данных, однако может обеспечить меньшую точность, так как часть информации существующими алгоритмами извлечь не удалось.

Исходя их описанных достоинств и недостатков рассматриваемых подходов, они были применены совместно: первый подход (ручное извлечение данных) применялся на первом этапе, чтобы корректно сформулировать алгоритм для второго подхода (автоматического извлечения данных), при этом результат ручного извлечения также использовался для оценки работы автоматического извлечения.

Так, по итогам ручного анализа 15% имеющегося массива документов было выделено 429 сущностей следующих типов:

* «способности» – 23 сущности;
* «компании» – 183 сущности;
* «интересы» – 50 сущностей;
* «технологии» – 78 сущностей;
* «материалы» – 95 сущностей.

Примеры выделенных сущностей:

* «способности»: «сертификация готовых изделий», «изготовление форсунок газовых турбин»;
* «компании»: «ПАО Электромеханика», «МИСиС»;
* «интересы»: «струйная 3D-печать», «производство стального порошка»;
* «технологии»: «выборочное лазерное плавление», «прямое лазерное спекание»;
* «материалы»: «тугоплавкие металлы», «порошок никелевый».

Примеры выделенных зависимостей:

* «компания – компания»: «Роскосмос» – «Sciaky, Inc.»;
* «способности – технологии»: «порошок из титанового сплава вт6» – «аддитивные технологии»;
* «компания – интерес»: «СПбПУ» – «развитие технологий».

На рисунке 1 приведён пример визуализации данных, где граф построен на реальных данных.



Рисунок 1 – пример визуализации на текстовых данных.

Нужно отметить, что для выделенных 429 сущностей потенциально возможно существование $\frac{1}{2}429\left(429-1\right)=91806$ зависимостей, однако извлечено было только 693 зависимости, что составляет 0,75% от потенциально возможного количества.

В связи с этим сделано предположение, что фактически существуют и другие зависимости между известными сущностями, информация о которых не содержится в явном виде в имеющихся данных, но может быть определена с помощью алгоритма предсказания новых зависимостей. Такие предсказанные зависимости могут связывать точки на разных слоях структуры данных и быть весьма неочевидными.

Существует большое количество семейств алгоритмов, применимых к данной задаче, в данной статье рассматривается одно из таких семейств.

**Описание используемого алгоритма.** В рамках данной работы предлагается к рассмотрению алгоритм, основанный на вычислении ряда характеристик графа, образованного сущностями и связями между ними. Алгоритм основан на использовании следующих десяти характеристик графа, реализованных в библиотеке NetworkX [14]:

* Сообщества, основанные на модулярности.
* Сообщества, полученные методом распространения меток.
* Связность.
* K – компонентная структура.
* Клики.
* Кластеризация.
* Доминирующие множества.
* Независимое множество.
* Паросочетание.
* Вершинное покрытие.

Ниже каждая из этих характеристик рассмотрена более подробно и для каждой из них выделены предлагаемые параметры алгоритма:

1. **Сообщества, основанные на модулярности.**

Сообщество – это подмножество точек графа и связи между ними.

Модулярность – величина из отрезка [-1,1], оптимизируя которую можно получить разбиение графа на сообщества.

Модуляность сообщества – это разность между количеством связей, входящих в него, и ожидаемым количеством связей в сообществе, если бы все связи были заданы случайно.

Рассматриваемый алгоритм изначально считает каждую точку отдельным сообществом, а затем объединяет пары сообществ так, чтобы модулярность возрастала, пока это не станет невозможным.

Результат работы методы – набор сообществ.

Предлагаемый параметр для анализа – количество сообществ.

1. **Сообщества, полученные методом распространения меток.**

Алгоритм присваивает каждой точке графа уникальную метку, затем для каждой точки берутся все метки в ее окрестности (от точек, соединенных с данной) и метка точки заменятся на самую популярную метку из окрестности. Если несколько меток одинаково популярны – берется одна из них случайным образом. Когда метка каждой точки совпадает с самой популярной точкой ее окрестности – алгоритм останавливается.

Результат работы методы – набор сообществ.

Предлагаемый параметр для анализа – количество сообществ.

1. **Связность.**

Связность – это минимальное количество точек, которые нужно удалить из графа, чтобы он стал несвязным.

Результат работы метода – величина связности графа.

Предлагаемый параметр для анализа – величина связности графа.

1. **K – компонентная структура.**

Алгоритм возвращает пары «число – подграф», где подграф – максимален и имеет связность не ниже указанного числа. *K*‑связность означает, что из графа надо удалить минимум *k* связей, чтобы он стал несвязным.

Предлагаемые параметры для анализа – размер подграфа для значения связности 1, 2, 3.

1. **Клики.**

Клика – это набор точек, каждая пара из которых соединена связью. Алгоритм возвращает клику максимального размера.

Предлагаемый параметр для анализа – размер максимальной клики.

1. **Кластеризация.**

Величина кластеризации для точки – это отношение количества треугольников, которые существуют в окрестности точки к общему количеству возможных треугольников в окрестности точки.

Средняя величина кластеризации графа – среднее значение кластеризации точек по всем точкам графа.

Алгоритм берет случайную точку в графе, затем берет две случайных точки в ее окрестности и проверяет, соединены ли они связями (есть ли треугольник). Это повторяется *N* раз, после чего оценкой кластеризации считается отношение количества найденных треугольников к числу повторов *N*. Алгоритм возвращает оценку средней величины кластеризации графа.

Предлагаемый параметр для анализа – средняя величина кластеризации для *N* больше числа точек в графе (10000) при фиксированном значении инициализации генератора случайных чисел.

1. **Доминирующие множества.**

Для графа можно определить доминирующее множество точек и доминирующее множество связей.

Доминирующее множество точек – такое подмножество точек графа, что любая точка, не входящая в это подмножество, соединена связью с одной из точек подмножества.

Доминирующее множество связей – такое подмножество связей графа, что любая связь, не входящая в это подмножество, содержит точку одной из связей подмножества.

Алгоритмы возвращают минимальные доминирующей множества точек и связей графа.

Предлагаемые параметры для анализа – размеры минимальных доминирующих множеств.

1. **Независимое множество.**

Независимое множество – это подмножество точек графа, никакие две из которых не связаны между собой.

Алгоритм возвращает максимальное независимое множество.

Предлагаемый параметр для анализа – размер максимального независимого множества.

1. **Паросочетание.**

Паросочетание – это множество попарно несмежных связей в графе.

Максимальное паросочетание – такое паросочетание, которое не содержится в другом паросочетании.

Алгоритм возвращает наименьшее максимальное паросочетание.

Предлагаемый параметр для анализа – размер наименьшего максимального паросочетания.

1. **Вершинное покрытие.**

Вершинное покрытие графа – такое множество точек графа, что у любой связи хотя бы одна точка входит в данное множество.

Алгоритм возвращает минимальное вершинное покрытие графа.

Предлагаемый параметр для анализа – размер минимального вершинного покрытия.

Итого, после рассмотрения приведённых выше характеристик, был сформулирован следующий список параметров для алгоритма:

* Количество сообществ, основанных на модулярности;
* Количество сообществ, найденных алгоритмом распространения меток;
* Величина связности графа;
* Размер подграфа для значения связности 1;
* Размер подграфа для значения связности 2;
* Размер подграфа для значения связности 3;
* Размер максимальной клики;
* Оценка средней величины кластеризации графа для 10000 повторов;
* Размер минимального доминирующего множества точек;
* Размер минимального доминирующего множества связей;
* Размер максимального независимого множества;
* Размер наименьшего максимального паросочетания;
* Размер минимального вершинного покрытия.

Алгоритм построения новых связей в графе имеет следующий вид:

1. Фиксируется список параметров для анализа.
2. Вычисляются и сохраняются значения параметров для исходного графа.
3. Формируется список потенциальных новых связей – всех возможных связей, отсутствующих в графе.
4. Для каждой связи из списка потенциальных новых связей вычисляются и сохраняются значения параметров для анализа.
5. Вычисляются евклидовы расстояния от значений параметров для потенциальных связей до значений параметров исходного графа.
6. Отбираются десять самых близких и десять самых дальних конфигураций графа по отношению к исходному графу.
7. Полученные данные анализируются и интерпретируются.

**Вычислительные эксперименты.** Для проведения вычислительных экспериментов, в том числе с применением мощностей СберКлауд [4], была создана программа на языке программирования Python, которая ищет новые связи.

В рамках серии экспериментов была проведена трудоемкая работа по анализу параметров. Все десять параметров были проверены и классифицированы на два типа: максимально влияющие на результат работы алгоритма и вносящие минимальные изменения.

План вычислительных экспериментов выглядел следующим образом: вычисляем самые похожие и самые непохожие десять потенциальных рёбер для следующих наборов параметров:

1. Первые два – «Количество сообществ, основанные на модулярности», «количество сообществ, полученные методом распространения меток»;

2. Первые три – «Количество сообществ, основанные на модулярности», «количество сообществ, полученные методом распространения меток», «величина связности графа»;

3. Первые четыре – «Количество сообществ, основанные на модулярности», «количество сообществ, полученные методом распространения меток», «величина связности графа», «размер подграфа для значения связности 1, 2, 3» (было показано, что имеет смысл рассматривать только один из этих трёх параметров, так как влияние на результат идентично);

4. Первые пять – «Количество сообществ, основанные на модулярности», «количество сообществ, полученные методом распространения меток», «величина связности графа», «размер подграфа для значения связности 1, 2, 3», «размер максимальной клики».

В результате оказалось, что для всех этих экспериментов полученные новые зависимости совпадают, причём вычисляемые в процессе работы расстояния между новыми вариантами графа и исходным графам также совпадают.

Далее аналогично были проанализированы различные наборы параметров с шестого по десятый: «Оценка средней величины кластеризации графа», «размеры минимальных доминирующих множеств», «размер максимального независимого множества», «размер наименьшего максимального паросочетания», «размер минимального вершинного покрытия».

В данном случае полученные отношения снова совпали, однако различались вычисленные расстояния.

Далее были проведены эксперимент по комбинации различных вариантов параметров и наблюдалось, как меняются новые отношения и вычисляемые расстояния. Целью данного эксперимента заключалось в том, чтобы увидеть изменения ребер и влияние данных параметров, понять их нужность и актуальность. В таблице 1 кратко изложены результаты теста.

|  |  |
| --- | --- |
| Параметр | Количество сообществ, основанные на модулярности, количество сообществ, полученные методом распространения меток |
| Максимальная клика | Ближние ребра – одинаковыеДальние ребра – разные |
| Оценка средней величины | Ближние ребра – отчасти схожи Дальние ребра – разные |
| Максимально независимое множество | Ближние ребра – разные Дальние ребра – разные |
| Мин. доминирующие множества | Ближние ребра – одинаковыеДальние ребра – разные |

Таблица 1 – результат тестирования.

В результате начальный список из десяти параметров был переработан:

* часть параметров (четыре) были отброшены по одной из двух причин: либо минимальное влияние на результат работы алгоритма, либо существенное увеличение времени работы алгоритма;
* часть параметров (четыре) были выбраны основными, как имеющие наибольшее влияние на результаты: «размер максимальной клики», «оценка средней величины кластеризации графа», «размер максимального независимого множества», «размер минимальных доминирующих множеств»;
* часть параметров (два) были оставлены как опциональные, в том числе для дальнейшего изучения: «количество сообществ, основанные на модулярности» и «количество сообществ, полученные методом распространения меток».

**Работа по созданию единой метрики.** Так как задача данной работы заключается в поиске новых зависимостей между объектами на основании имеющейся информации об объектах и их известных связях, для решения данной задачи необходимо иметь возможность сравнивать различные алгоритмы или версии одного алгоритма. Добиться необходимого результата возможно путём создания единой метрики как критерия для сравнения.

В рамках данной работе рассмотрено одиннадцать различных метрик эффективности алгоритмов поиска новых зависимостей, как общеупотребительных, так и специфичных для генеративно‑состязательных сетей [15]:

* Frechet Inception Distance (FID) [11] – по своей сути FID сравнивает статистику сгенерированных образцов с реальными выборками. Чем меньше расстояние между распределениями, тем выше схожесть между реальным и сгенерированным изображением.
* Annealed Importance Sampling (AIS) [10] – возвращает набор взвешенных выборок, и, в пределе бесконечного множества промежуточных распределений, дисперсия весов приближается к нулю. Чаще всего используется для оценки функций распределения, а также для оценки статистической суммы вероятностной модели.
* Geometry Score [13] – данная метрика может помочь корректировать и «настраивать» гиперпараметры в ходе работы алгоритма. В этом методе проблема оценки качества и разнообразия сгенерированных изображений решается путем изучения топологии базового множества сгенерированных выборок, которые могут отличаться от топологии исходного множества данных, что дает представление о свойствах GAN и может использоваться для настройки гиперпараметров.
* Tournament Based Method [12] – соревновательный подход, в котором одна модель оценивается, играя против её прошлых и будущих версий. Процент побед в турнире обозначает среднюю скорость, с которой сеть генератора «обманывает» сеть дискриминатора. Система оценки навыков, как следует из названия, даёт рейтинг навыков для каждого генератора, то есть данная метрика помогает контролировать процесс обучения GAN. И эти измерения подразделяются на два рейтинга: процент побед и рейтинг навыков.
* Discriminator Rejection Sampling [6] – идея этого метода заключается в том, чтобы улучшить качество обученных генераторов путем последующей обработки их выборок с использованием информации от обученного дискриминатора. При довольно строгих предположениях это позволяет точно восстановить распределение данных. Затем проводится исследование, где эти строгие допущения нарушаются, и разрабатывается практический алгоритм, называемый выборкой отбраковки дискриминатора, который можно использовать с реальными наборами данных.
* Precision And Recall [16] – с помощью этой метрики измеряется точность. Ключевая идея данной метрики заключается в измерении точности представленных данных с «заложенным нами идеалом», что практически невозможно в нашей задаче. Точность определяется как доля соответствующих экземпляров среди всех извлеченных экземпляров.
* Birthday Paradox Test [17] – идея этой метрики заключается в том, что алгоритм пытается найти «клона» исходных данных среди всех данных. Чаще всего это не работает, но если выбрать *n* ближайших пар в соответствии, то получим потенциальный пул «почти дубликатов».
* Normalized Relative Discriminative Score (NRDS) [19] – с помощью данной метрики можем сравнивать полученные результаты и выделять из них «наиболее ценные».
* Hits\_at\_n\_score [5, 7] – суть метрики в формировании некоторого списка по «важности» среди выстроенных связей, что отчасти подходит к данной задаче.
* Hit and K [5, 7] – данная метрика частично подойдет к данной задаче, она используется при поиске информации по критериям в документах.
* Mrr\_score [5, 7] – метрика направлена на выявление наиболее соответствующих полученных критериев к заданным. Подойдет совместно с другой метрикой.

Ниже представлена оценочная табл. 2, в которой каждая метрика рассматривается на критерии соответствия.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Соответствует | Частично соответствует | Не соответствует |
| Frechet Inception Distance (FID) |  | + |  |
| Annealed Importance Sampling (AIS) |  |  | + |
| Geometry Score |  | + |  |
| Tournament Based Method |  |  | + |
| Discriminator Rejection Sampling |  | + |  |
| Precision And Recall |  |  | + |
| Birthday Paradox Test |  |  | + |
| Normalized Relative Discriminative Score (NRDS) | + |  |  |
| Hits\_at\_n\_score |  | + |  |
| Hit and K  |  | + |  |
| Mrr\_score | + |  |  |

Таблица 2 – Оценка метрик.

Для оценки того, насколько каждая рассматриваемая метрика подходит к решаемой задачи, был сформирован набор требований:

1. Универсальность – метрика не должна быть слишком специфичной, т.к. она будет применяться к различным алгоритмам и не должна вносить слишком жёсткие ограничения на их устройство.
2. Простота – числовое значение метрики должно вычисляться быстро, как минимум – оценка алгоритма не должна занимать больше времени, чем работа алгоритма.
3. Интерпретируемость – результаты применения метрики не должны быть просто числами для сравнения друг с другом, а должны соответствовать каким – то действительно существующим величинам.
4. Устойчивость – результаты вычисления метрики для алгоритма не должны меняться сильно при небольших изменениях параметров самого алгоритма.

Представим данный вывод с помощью табл. 3.

|  |  |  |
| --- | --- | --- |
|  | «Самостоятельная» метрика | «Дополнительная» метрика |
| Frechet Inception Distance (FID) |  | + |
| Geometry Score |  | + |
| Discriminator Rejection Sampling |  | + |
| Normalized Relative Discriminative Score (NRDS) | + |  |
| Hits\_at\_n\_score |  | + |
| Hit and K |  | + |
| Mrr\_score | + |  |

Таблица 3 – Результат по метрикам.

На основании данных требований из проанализированных метрик были выделены две подгруппы: основные метрики, которые могут применяться самостоятельно, и дополнительные метрики, которые могут применяться вместо с основными для уточнения оценки.

В подгруппу основных метрик вошли Normalized Relative Discriminative Score (NRDS) и Mrr\_score.

В подгруппу дополнительных метрик вошли Frechet Inception Distance (FID), Geometry Score, Discriminator Rejection Sampling, Hits\_at\_n\_score и Hit and K.

**Выводы и планы дальнейших исследований.** В данной работе разработан формат хранения информации, извлечённой из плохо структурированных данных; предложен алгоритм поиска новых, ранее неизвестных зависимостей в графе; проведен ряд экспериментов, исходя из которых были отобраны параметры, которые не дают ошибочных результатов; проанализировано одиннадцать различных метрик, для дальнейшего анализа отобрано пять из них.

В дальнейшем необходимо разработать и рассмотреть другие, более сложные и более точные алгоритмы формирования векторных представлений рёбер графов, а также продолжить работу над созданием единой метрики, которая поможет упростить работу оценки эффективности алгоритмов.

Также, для снятия ряда ограничений, имеющихся в рассматриваемом подходе к решению задачи, планируется рассматривать данные не как статические, а в динамике. Это усложнит процесс сбора и обработки данных, однако позволит более интерпретируемо применять алгоритмы поиска новых зависимостей.

 *Литература*

1. Ефимова, А.О., Соколова, Д.А. Представление и поиск информации с помощью графа онтологии // Сборник материалов XXI Ежегодной научной конференции студентов Технологического университета «Ресурсам области – эффективное использование». 2021.
2. Муромцев Д., Волчек Д., Романов А. Индустриальные графы знаний – интеллектуальное ядро цифровой экономики // Control Engineering Россия. 2019. № 5 (83). С. 32 – 39.
3. Орлов Ю.Н., Парфенова Ю.А. Анализ структуры онтологического графа толкового словаря // Препринты ИПМ им. М.В.Келдыша. 2017. № 31. 25 с.
4. СберКлауд [Электронный ресурс]. Режим доступа: https://aicloud.sbercloud.ru (дата обращения: 07.06.2021).
5. Accenture. Ampligraph [Электронный ресурс]. Режим доступа: https://github.com/Accenture/AmpliGraph (дата обращения: 21.03.2021).
6. Azadi S., Olsson C., Darrell T., Goodfellow I., Augustus O. Discriminator rejection sampling [Электронный ресурс]. Режим доступа: https://deepai.org/publication/discriminator-rejection-sampling (дата обращения: 21.03.2021).
7. Costabello L., Pai S., Van C.L., McGrath R., McCarthy N., Tabacof P. AmpliGraph: a Library for Representation Learning on Knowledge Graphs [Электронный ресурс]. Режим доступа: https://zenodo.org/record/4792436 (дата обращения: 21.03.2021).
8. De Domenico M. Multilayer Networks: Analysis and Visualization. Introduction to muxViz with R // Принято к публикации в Springer-Verlag. 2021.
9. De Domenico M., Porter M.A., Arenas A. Multilayer Analysis and Visualization of Networks // Journal of Complex Networks 3. 2015. PP. 159 –176.
10. Grosse R., Reed C. Annealed importance sampling [Электронный ресурс]. Режим доступа: https://metacademy.org/graphs/concepts/

annealed\_importance\_sampling (дата обращения: 21.03.2021).

1. Jean N. Fréchet Inception Distance [Электронный ресурс]. Режим доступа: https://nealjean.com/ml/frechet-inception-distance (дата обращения: 21.03.2021).
2. Jiang X., Gripon V., Berrou C., Rabbat M. Storing sequences in binary tournament-based neural networks [Электронный ресурс]. Режим доступа: https://arxiv.org/pdf/1409.0334v1.pdf (дата обращения: 21.03.2021).
3. Khrulkov V., Oseledets I. Geometry Score: A Method For Comparing Generative Adversarial Networks [Электронный ресурс]. Режим доступа: https://arxiv.org/pdf/1802.02664.pdf (дата обращения: 21.03.2021).
4. Network Analysis in Python [Электронный ресурс]. Режим доступа: [https://networkx.github.io](https://networkx.github.io/) (дата обращения: 06.03.2021).
5. Sagar R. Top 6 Metrics To Monitor The Performance Of GANs [Электронный ресурс]. Режим доступа: https://analyticsindiamag.com/top-6-metrics-to-monitor-the-performance-of-gans (дата обращения: 07.04.2021).
6. Sajjadi M.S.M., Bachem O., Lucic M., Bousquet O., Gelly S. Assessing Generative Models via Precision and Recall [Электронный ресурс]. Режим доступа: https://arxiv.org/pdf/1806.00035.pdf (дата обращения: 21.03.2021).
7. Sanjeev A., Zhang Y. Do GANs actually learn the distribution? An empirical study [Электронный ресурс]. Режим доступа: https://arxiv.org/pdf/1706.08224.pdf (дата обращения: 21.03.2021).
8. The Open Graph Viz Platform [Электронный ресурс]. Режим доступа: https://gephi.org/ (дата обращения:06.03.2021).
9. Zhang Zh., Song Y., Qi H. Decoupled Learning for Conditional Adversarial Networks [Электронный ресурс]. Режим доступа: https://arxiv.org/pdf/1801.06790.pdf (дата обращения: 21.03.2021).