

# Algorithm for link prediction in self-regulating network with adaptive topology based on graph theory and machine learning

E. Y. Pavlenko<sup>1</sup>

DOI: [10.18255/1818-1015-2023-4-288-307](https://doi.org/10.18255/1818-1015-2023-4-288-307)

<sup>1</sup>Peter the Great St. Petersburg Polytechnic University, 29 Polytechnicheskaya str., St. Petersburg 195251, Russia.

MSC2020: 93B70; 68R10

Research article

Full text in Russian

Received August 7, 2023

After revision October 24, 2023

Accepted November 2, 2023

The paper presents a graph model of the functioning of a network with adaptive topology, where the network nodes represent the vertices of the graph, and data exchange between the nodes is represented as edges. The dynamic nature of network interaction complicates the solution of the task of monitoring and controlling the functioning of a network with adaptive topology, which must be performed to ensure guaranteed correct network interaction. The importance of solving such a problem is justified by the creation of modern information and cyber-physical systems, which are based on networks with adaptive topology. The dynamic nature of links between nodes, on the one hand, allows to provide self-regulation of the network, on the other hand, significantly complicates the control over the network operation due to the impossibility of identifying a single pattern of network interaction.

On the basis of the developed model of network functioning with adaptive topology, a graph algorithm for link prediction is proposed, which is extended to the case of peer-to-peer networks. The algorithm is based on significant parameters of network nodes, characterizing both their physical characteristics (signal level, battery charge) and their characteristics as objects of network interaction (characteristics of centrality of graph nodes). Correctness and adequacy of the developed algorithm is confirmed by experimental results on modeling of a peer-to-peer network with adaptive topology and its self-regulation at removal of various nodes.

**Keywords:** modeling; networks with adaptive topology; graph model; link prediction; centrality metrics

## INFORMATION ABOUT THE AUTHORS

Evgeny Y. Pavlenko | [orcid.org/0000-0003-1345-1874](https://orcid.org/0000-0003-1345-1874). E-mail: [pavlenko@ibks.spbstu.ru](mailto:pavlenko@ibks.spbstu.ru)  
corresponding author | Associate Professor, Ph.D. in Technical Sciences.

**Funding:** The research is funded by the Russian Science Foundation, project no. 22-21-20008. The research is funded by the grant of the St. Petersburg Science Foundation in accordance with the agreement of April 15, 2022 № 61/220.

**For citation:** E. Y. Pavlenko, "Algorithm for link prediction in self-regulating network with adaptive topology based on graph theory and machine learning", *Modeling and analysis of information systems*, vol. 30, no. 4, pp. 288-307, 2023.

# Алгоритм предсказания связей в саморегулирующейся сети с адаптивной топологией на базе теории графов и машинного обучения

Е. Ю. Павленко<sup>1</sup>

DOI: [10.18255/1818-1015-2023-4-288-307](https://doi.org/10.18255/1818-1015-2023-4-288-307)

<sup>1</sup>Санкт-Петербургский политехнический университет Петра Великого, ул. Политехническая, д. 29, г. Санкт-Петербург, 195251 Россия.

УДК 519.17

Получена 7 августа 2023 г.

Научная статья

После доработки 24 октября 2023 г.

Полный текст на русском языке

Принята к публикации 2 ноября 2023 г.

В статье представлена графовая модель функционирования сети с адаптивной топологией, где узлы сети представляют собой вершины графа, а обмен данными между узлами представлен в виде ребер. Динамический характер сетевого взаимодействия усложняет решение задачи мониторинга и контроля функционирования сети с адаптивной топологией, которую необходимо выполнять для обеспечения гарантированно корректного сетевого взаимодействия. Значимость решения такой задачи обосновывается созданием современных информационных и киберфизических систем, в основе которых лежат сети с адаптивной топологией. Динамический характер связей между узлами, с одной стороны, позволяет обеспечивать саморегуляцию сети, с другой стороны, существенно усложняет контроль за работой сети в связи с невозможностью выделения единого шаблона сетевого взаимодействия.

На базе разработанной модели функционирования сети с адаптивной топологией предложен графовый алгоритм предсказания связей, распространенный на случай с одноранговыми сетями. В основу алгоритма положены значимые параметры узлов сети, характеризующие как их физические характеристики (уровень сигнала, заряд батареи), так и их характеристики как объектов сетевого взаимодействия (характеристики центральности вершин графа). Корректность и адекватность разработанного алгоритма подтверждена экспериментальными результатами по моделированию одноранговой сети с адаптивной топологией и ее саморегуляции при удалении различных узлов.

**Ключевые слова:** моделирование; сети с адаптивной топологией; графовая модель; предсказание связей; метрики центральности

## ИНФОРМАЦИЯ ОБ АВТОРАХ

Евгений Юрьевич Павленко  
автор для корреспонденции

[orcid.org/0000-0003-1345-1874](https://orcid.org/0000-0003-1345-1874). E-mail: [pavlenko@ibks.spbstu.ru](mailto:pavlenko@ibks.spbstu.ru)  
доцент, кандидат технических наук.

**Финансирование:** Исследование выполнено за счет гранта Российского научного фонда № 22-21-20008. Исследование выполнено за счет гранта Санкт-Петербургского научного фонда в соответствии с соглашением от 15 апреля 2022 г. № 61/220.

**Для цитирования:** Е. Ю. Павленко, “Algorithm for link prediction in self-regulating network with adaptive topology based on graph theory and machine learning”, *Modeling and analysis of information systems*, vol. 30, no. 4, pp. 288-307, 2023.

## Введение

Развитие информационных и сетевых технологий, цифровизация различных отраслей деятельности человека привели к трансформации информационных систем, расширив их состав интеллектуальными устройствами физического мира и изменив принципы их сетевого взаимодействия за счет появления беспроводных самоорганизующихся сетей (ad hoc сети, сети MANET и VANET) [1, 2].

Цифровые информационные системы, в состав которых входит большое число интеллектуальных устройств, взаимодействующих друг с другом и окружающей средой посредством сенсорных и сетевых технологий, далее будем называть киберфизическими системами (КФС). Ключевым отличием КФС от традиционных информационных систем является соединение необратимых физических и обратимых информационных процессов в рамках единой системы. При этом ведущая роль в реализации процессов остается за интеллектуальными устройствами, а влияние человека на работу КФС сведено к минимуму.

Сетевая топология КФС может варьироваться от классической клиент-серверной до динамической, с переменным составом узлов и связей между ними, а также с возможностью перемещения узлов в пространстве. К преимуществам динамической сетевой инфраструктуры следует отнести возможность саморегуляции структуры таких сетей, возможную за счет адаптивного перестроения структуры сети в случае появления/исчезновения участников сетевого взаимодействия и разрыва сетевых соединений. Это весомый фактор, в частности, для КФС, предоставляющих различные сервисы населению — медицинскую помощь, навигацию людей в опасных регионах с ограниченным числом аварийных выходов, различные транспортные сервисы [3–7].

Беспроводные сети с адаптивной топологией можно разделить на следующие типы [1]:

1. Беспроводные децентрализованные самоорганизующиеся сети мобильных устройств (mobile ad hoc network, MANET).
2. Автомобильные самоорганизующиеся сети (vehicular ad hoc network, VANET). Используются для связи между транспортными средствами.
3. Беспроводные сети ad hoc, использующие смартфоны (smartphone ad hoc network, SPAN). После внедрения технологии сетей ad hoc группа смартфонов, находящихся в непосредственной близости друг от друга, может совместно создать сеть ad hoc.
4. Беспроводные ячеистые сети (wireless mesh network, mesh-сеть) — коммуникационная сеть, состоящая из радиоузлов, организованных в ячеистой топологии. Каждый узел в mesh-сети работает как маршрутизатор и как хост.
5. Военные тактические сети (army tactical MENT) — используется военными для связи «на ходу». Такая беспроводная тактическая сеть ad hoc полагается на дальность действия и мгновенную работу для организации сетевого взаимодействия в случае необходимости.
6. Беспроводные сенсорные сети (wireless sensor network, WSN). Датчики в беспроводных сенсорных сетях обычно представляют собой небольшие сетевые узлы с очень ограниченной вычислительной мощностью, ограниченной коммуникационной способностью и ограниченным источником питания. Таким образом, датчик может выполнять только простые вычисления и общаться с датчиками и другими узлами на небольшом расстоянии.
7. Сети ad hoc для спасения при стихийных бедствиях (disaster rescue ad hoc network). Такие ad hoc сети используются, когда случается катастрофа и установленное коммуникационное оборудование не функционирует должным образом.

Для эффективной саморегуляции сетевой инфраструктуры, при которой обеспечивается доставка данных от всех узлов сети к центру обработки данных и стабильная работа системы, построенной на базе динамической сети, необходимо выявить принципы функционирования самой сети

и объединить их со спецификой, накладываемой видом системы. Это позволит адаптировать уже заложенные архитектурно возможные возможности саморегуляции системы к ее целевой функции, которая будет значительно варьироваться в зависимости от типа системы.

Так, для КФС мониторинга труднодоступных регионов (например, лесов, гор, мест землетрясений) целевая функция будет заключаться в качественном получении и доставке данных от всех (или максимально возможного числа) сенсоров, то есть, при разрыве связи между парой узлов необходимо, чтобы данные от узлов не потерялись [8]. В этом случае первостепенное значение имеет связность сети, и саморегуляция должна быть направлена на ее обеспечение.

Для КФС другого типа, например, промышленных объектов, ключевое значение может иметь непрерывающееся выполнение какого-либо процесса, пусть даже с потерей качества в связи с выходом из строя части узлов-участников сетевого взаимодействия. В таком случае саморегуляция должна быть направлена на выполнение этого процесса, что может быть выражено как последовательный сетевой обмен между узлами определенного типа в заданном порядке.

Целью настоящей статьи является создание алгоритма предсказания связей, работающего на базе разработанной ранее автором графовой модели функционирования сетей с адаптивной топологией [9]. В рамках модели изменчивая топология таких сетей рассматривается в динамике за счет представления состояний сети в виде «снимков» динамического графа. Значимым преимуществом данной модели среди подобных является ее способность описывать различные типы сетей с адаптивной топологией и ее ориентированность на сети, являющиеся базой для КФС, за счет единовременного учета физических и информационных характеристик узлов сети.

Разработанный алгоритм предсказания связей использует параметры, которыми оперирует модель, и выполняет предсказание появления или исчезновения ребер в «снимке» динамического графа в следующий момент времени. Область применения данного алгоритма уже, чем у модели — он ориентирован только на работу одноранговых сенсорных сетей как наиболее простого типа сетей с адаптивной топологией. Исходя из результатов экспериментальных исследований, подтвердивших его эффективность и соответствие поведению сетей с адаптивной топологией на практике, в рамках дальнейших исследований целесообразно расширить данный алгоритм, распространив его на случаи с гетерогенными сенсорными и промышленными сетями.

Статья организована следующим образом:

- в главе 1 представлен обзор связанных работ, цель которого — сравнить разработанный алгоритм с аналогами и описать в целом область применения алгоритмов предсказания связей;
- в главе 2 приводится краткое описание разработанной ранее автором графовой модели, подкрепленное практическим примером для случая с гетерогенной сенсорной сетью;
- глава 3 содержит формальную постановку задачи разработки алгоритма предсказания связей в одноранговых сенсорных сетях, включающая определение целевой функции для формирования условий к запуску алгоритма, требования к алгоритму и используемые им параметры;
- в главе 4 представлено описание работы алгоритма в виде псевдокода;
- глава 5 содержит описание проведенных экспериментальных исследований.

## 1. Обзор связанных работ в части алгоритмов предсказания связей

Алгоритмы предсказания связей в зарубежной литературе носят название link prediction. Области применения таких алгоритмов включают предсказание неизвестных белковых взаимодействий, прогнозирование реакции на лекарственные препараты [10], рекомендации продуктов пользователям [11], заполнение графов знаний [12]. Также они нашли широкое применение в графовых методах анализа взаимосвязей между пользователями в социальных сетях [13, 14].

Работа таких алгоритмов весьма разнообразна, однако наиболее распространенный подход, реализуемый ими, состоит в оценке схожести вершин графа в соответствии с некоторым множеством

заданных характеристик. Так, для социальных сетей к таким метрикам могут относиться: число общих друзей, одно и то же место учебы или работы в сочетании с похожим возрастом, схожие интересы и т. п.

Однако следует отметить, что подходы, основанные на сходстве, имеют некоторые проблемы, связанные с потерей информации об узлах и обобщающей способностью индексов сходства.

Работы, посвященные предсказанию связей в различных сетях, как правило, используют такие графовые характеристики как метрики центральности, вес пути между узлами и распределение степеней графа [15–18]. Однако часто вычисление метрик центральности используется для определения наиболее «влиятельных» узлов в сети, что не требуется для предсказания связей в одноранговых сенсорных сетях, где все узлы равноправны и одинаковы по характеристикам.

Многие алгоритмы link prediction реализуются с использованием искусственных нейронных сетей или алгоритмов машинного обучения, и следует отметить, что это направление перспективно за счет возможности более эффективной обработки больших объемов данных и выявления неявных зависимостей. Однако для всех методов и алгоритмов, использующих техники искусственного интеллекта, важны два аспекта:

- сбалансированная и репрезентативная обучающая выборка, сформированная таким образом, чтобы в полной мере отражать специфику данных исследуемой предметной области;
- грамотный выбор признаков объектов, используемых для обучения.

Как правило, наборы обучающих данных для КФС и сетей с адаптивной топологией (сенсорные сети, MANET и VANET) отражают либо физические аспекты функционирования системы (показатели мощности сигнала, заряда устройств, координаты их перемещения), либо сетевые (число отправленных и полученных сетевых пакетов, используемый протокол маршрутизации и т. п.). Крайне малое число наборов данных включает информацию о состоянии сетей с адаптивной топологией, вследствие чего исследования, посвященные решению задач мониторинга и анализа безопасности таких сетей, используют достаточно давно опубликованные наборы данных, такие как KDD-99 [19, 20].

Относительно грамотного выбора признаков объектов, в ряде публикаций наблюдается стремление авторов переложить задачу выявления значимых признаков объектов на искусственный интеллект. Так, авторы работы [21] утверждают, что в силу отсутствия знаний о некоторых эвристиках, используемых для предсказания связей, целесообразнее получить эвристики, характерные для каждой отдельно взятой сети, с использованием графовых нейронных сетей. Такой подход не подходит для предсказания связей в сетях, на базе которых функционируют КФС, поскольку для таких систем особенную значимость имеют как физические характеристики узлов сети, так и информационные характеристики сетевого обмена.

Однако часть публикаций посвящена абстрактным исследованиям сложных сетевых структур, в которых признаки графа не описываются явно, и оставляют возможность для других исследователей по включению в модель признаков различной природы. К таким исследованиям относится работа [22], авторы которой предлагают использовать графовые нейронные сети для предсказания связей, уделяя при этом особое внимание подграфам, моделирующим отдельные части сети. Такое направление является перспективным для промышленных сетей, для которых с точки зрения графовой модели характерно наличие нескольких компонент связности.

Применительно к КФС и сенсорным сетям, алгоритмы предсказания связей часто используются для прогнозирования маршрутов в сетях, предсказания сбоев и решения подобных проблем, связанных с надежностью. Так, авторы работы [23] также используют link prediction для прогнозирования качества каналов связи как одной из важнейших задач маршрутизации в сложных сетях.

В работе [24] авторы прогнозируют состояния каналов для сетей подводных акустических датчиков. Они отмечают, что акустический канал значительно меняется с течением времени, однако есть ряд физических характеристик, по которым можно оценить состояние канала. В рамках алгоритма предсказания связей авторы используют такие параметры как интенсивность сигнала, интенсивность шума и расстояние между узлами. Таким образом, графовые метрики здесь не учитываются, набор характеристик ограничен физическими параметрами. В основе алгоритма лежит получение краткосрочного прогноза с использованием фильтра Калмана, что говорит также о значительной сложности настройки параметров фильтра для конкретной исследуемой сети.

Исследование авторов работы [25] посвящено прогнозированию связей в мобильных ad hoc сетях (MANET), решение этой задачи существенно осложняется подвижностью узлов в пространстве. Для предсказания связей авторы предлагают алгоритм, учитывающий длительность связей между узлами, топологию сети и корреляции между узлами. По их мнению, это позволяет получить прогноз с учетом пространственно-временных характеристик узлов. Однако авторы не учитывают физические параметры мобильных узлов, что может значительно повлиять на результат прогноза. Так, атака физического подавления сигнала в определенном радиусе исключит из сети сразу несколько узлов, а устройство с разряженной батареей в любой момент может отключиться и перестать принимать и передавать данные.

Отличительной чертой предложенного в данной работе алгоритма является одновременный учет им как физических параметров устройств, составляющих сеть, так и тех параметров, которые оказывают большое влияние на информационные процессы. Эти параметры выражаются посредством графовых метрик центральности.

Значимость данного алгоритма с практической точки зрения обоснована его соответствием поведению сетей с адаптивной топологией, которые способны к саморегуляции. Кроме того, условия начала работы алгоритма тесно связаны с целевой функцией той системы, основой для работы которой является исследуемая сеть. Следует отметить, что разработанный алгоритм предназначен для одновременного учета физических, информационных и функциональных свойств сети, что отличает его от аналогичных алгоритмов предсказания связей.

## 2. Графовая модель функционирования сетей с адаптивной топологией

Модель функционирования сетей с адаптивной топологией базируется на динамической теории графов [26]. На взгляд автора, для решения задач мониторинга и контроля состояния сети с адаптивной топологией в первую очередь необходимо иметь возможность наблюдения за сетью — ее структурой и характеристиками узлов и ребер — в динамике, что позволит сравнивать состояния сети в различные моменты времени. Динамический граф удобно рассматривать как дискретную последовательность статических графов, изучая свойства каждого «снимка» графа с использованием методов, ориентированных на статические графы.

Подробное описание процесса разработки модели представлено в более ранней статье автора [9]. К значимым параметрам, которые следует учесть в модели, относятся как характеристики вершин и ребер графа, так и характеристики графа в целом (например, число компонент связности).

### 2.1. Описание модели

Сеть с адаптивной топологией представляется в виде ориентированного графа  $G = (V, E)$ , где  $V = v_1, v_2, \dots, v_n$  — множество узлов сети,  $E = e_1, e_2, \dots, e_m$  — множество дуг,  $E \subseteq (VV)$ .

Начальное состояние сети обозначается  $G^0$ , а некоторое состояние сети в момент времени  $p$  обозначается  $G^p$ .

$DG$  — динамический граф, представляющий собой набор (серию) «снимков» графа  $G$ ,  $DG = (S^0, S^1, \dots)$ , где  $S^{step}$  — «снимок» графа. Любой «снимок» графа  $S^{step}$  включает атрибуты:

- предыдущее состояние сети, выраженное в виде графа  $G^{t_{step}}$ . На первом шаге  $t_{step}$  принимает значение  $t_0$ , «снимок» графа на нулевом шаге представляет собой изначальное состояние исследуемой сети;
- набор выполненных преобразований  $Op^{step}$ ;
- временная метка  $t_n$ .

Таким образом,  $S^{step} = (G^{t_n}, Op^{step}, t_n)$ .

1.  $Op^{step}$  – кортеж, характеризующий изменения, совершенные при преобразовании предыдущего «снимка» к текущему;  $Op = (action, V, E)$ , где  $action$  – набор булевых признаков, демонстрирующий, какие унарные операции над графом применяются на данном шаге,  $V$  – вершины, к которым были применены операции  $action$ ,  $E$  – дуги, к которым были применены операции  $action$ ;  $action = (VA, VD, EA, ED)$ , где:
  - $VA$  – операция добавления вершины,  $VA = 0$ , если вершина не была добавлена,  $VA = 1$ , если вершина была добавлена).
  - $VD$  – операция удаления вершины,  $VD = 0$ , если вершина не была удалена,  $VD = 1$ , если вершина была удалена).
  - $EA$  – операция добавления дуги,  $EA = 0$ , если дуга не была добавлена,  $EA = 1$ , если дуга была добавлена).
  - $ED$  – операция удаления дуги,  $ED = 0$ , если дуга не была удалена,  $ED = 1$ , если дуга была удалена).

Результатом применения каждой из операций к вершине или паре вершин является бинарный ответ: 0 или 1, в зависимости от того, было выполнено преобразование или нет.

При преобразовании предыдущего «снимка» графа каждая унарная операция ассоциируется с временной меткой. В ходе практической реализации модели предполагается использование базы данных, в которую будут сохраняться сведения о конфигурациях графа и внесенных изменениях. В базе данных также предполагается хранение пар совершенных унарных операций и соответствующих каждой из них временных меток. Такая упорядоченность во времени обеспечивает согласованность операций и не позволяет, например, добавить ребро к удаленной вершине.

В описании модели предлагается сосредоточиться именно на характере внесенных изменений, в связи с чем на первый план выходит множество типов совершенных преобразований при переходе к новому «снимку».

2. «Снимок» графа  $G^{step}$  описывается как общими структурными характеристиками графа  $Str$ , так и характеристиками отдельных вершин ( $Vrtx$ ) и дуг ( $Ed$ ),  $G^{step} = (Str, Vrtx, Ed)$ .

2.1 Структурные характеристики:  $Str = (Cn, D, |V|, |E|)$ , где:

- $Cn$  – связность графа, определяет число компонент сильной связности  $num$ ;
- $D$  – наибольший диаметр компонент связности,  $D = d(v_a, v_b)$ , где  $v_a$  и  $v_b$  – наиболее удаленные друг от друга вершины графа;
- $|V|$  – число вершин графа;
- $|E|$  – число дуг графа.

2.2 Характеристики отдельных вершин  $Vrtx$ . На практике множество вершин, характеристики которых будут храниться в памяти системы и анализироваться, может варьироваться от всего множества вершин графа, моделирующего сеть, до множества некоторых критических вершин, характеризующих наиболее значимые узлы сети. Зададим множество вершин  $V^c \subseteq V$ , данные о которых будут сохраняться и отслеживаться при практической реализации модели. Для этого множества вершин  $V^c$  в модели определены следующие характеристики:  $Vrtx = (Cent, Pow)$ , где:

- $Cent$  – значения метрик центральности для множества вершин  $V^c$ . Таким образом, для каждой вершины  $v_i \in V^c$  множество  $Cent$  представляет собой вектор значений  $Cent^i = (Cent_1, Cent_2, Cent_3)$ , где  $Cent_1$  – степень вершины,  $Cent_2$  – степень посредничества,  $Cent_3$  – степень близости;
- $Pow$  – мощность вершин из множества  $V^c$ , характеризующих узлы моделируемой сети с точки зрения их энергоемкости.

2.3 Характеристики отдельных дуг  $Ed$ . По аналогии с вершинами, выделим некоторое множество дуг  $E^c$ , характеристики которых будут постоянно контролироваться при дальнейшем анализе функционирования системы. Для них в модели определены следующие характеристики:  $Ed = (W, TS)$ , где:

- $W$  – вес дуги, характеризующий качество сетевого соединения. Как было отмечено ранее, практическая интерпретация данной характеристики может варьироваться в зависимости от типа системы;
- $TS$  – множество временных рядов для этой дуги за время, прошедшее с момента предыдущего «снимка» сети.

## 2.2. Пример описания реальной сети в терминах модели

Рассмотрим применение модели к реальной сети с адаптивной топологией на примере беспроводной гетерогенной сети из источника [27]. Зафиксируем ее структуру в момент времени  $t_0$  (рисунок 1).

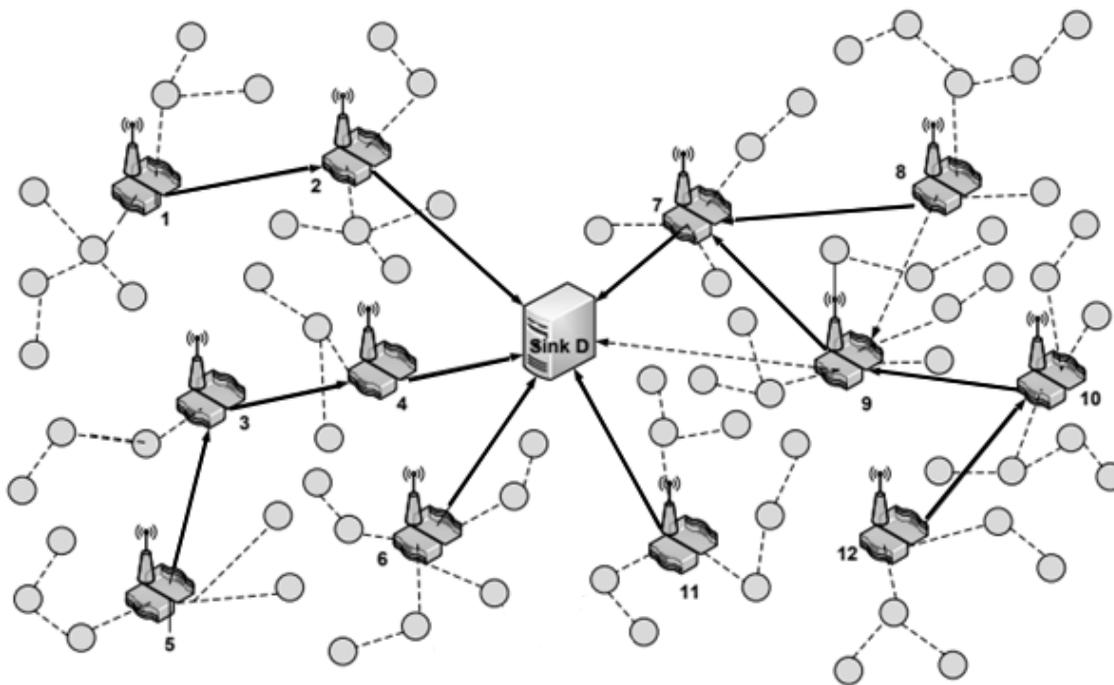


Fig. 1. Sensor network structure at time  $t_0$

Рис. 1. Структура сенсорной сети в момент времени  $t_0$

Узлы с номерами от 1 до 12 представляют собой так называемые «супер-узлы», их мощность сигнала выше, чем у обычных сенсоров, обозначенных на рисунке серыми кругами. За счет этого они могут агрегировать показания от множества сенсоров и передавать их на узел  $D$  ( $sink D$  на ри-

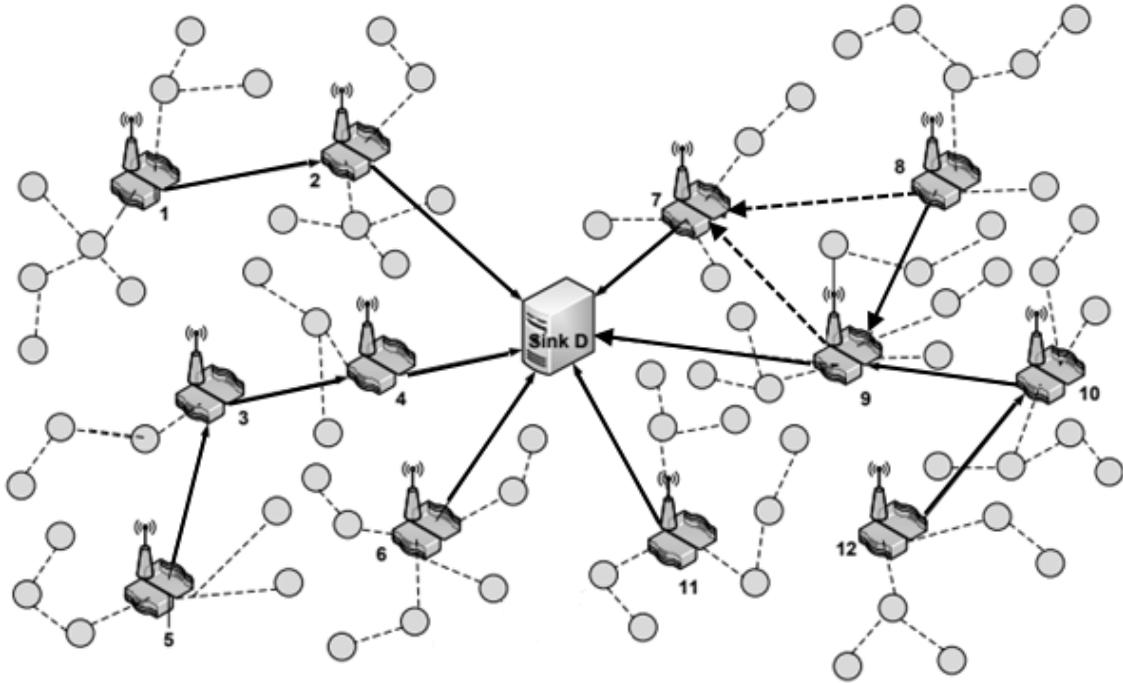
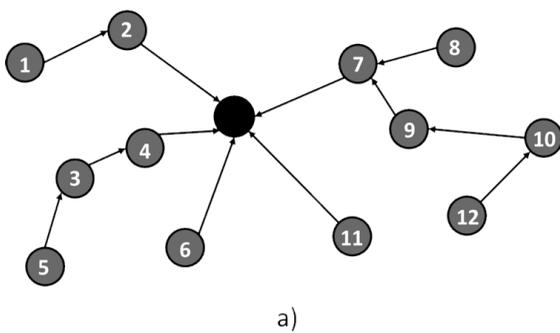
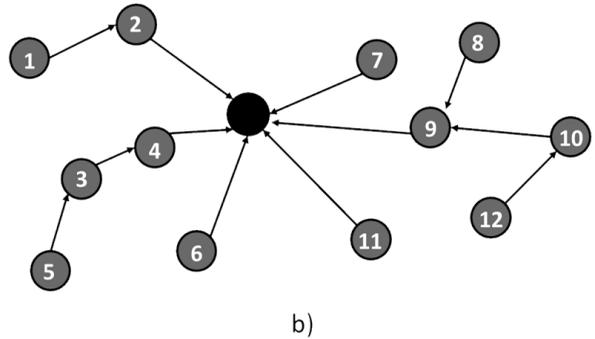


Fig. 2. Sensor network structure at time  $t_1$

Рис. 2. Структура сенсорной сети в момент времени  $t_1$



a)



b)

Fig. 3. Snapshots of dynamic graph

Рис. 3. «Снимки» динамического графа

сунке). Этот узел с точки зрения графа является стоком, физически он представляет собой сервер или точку доступа.

Связи между супер-узлами и сервером могут быть двух типов: прямой линией обозначена активная связь, в соответствии с которой идет обмен данными, пунктирной линией обозначены альтернативные пути обмена данными. Исследуем эту сеть в динамике, активируя альтернативные пути и получая «снимок» графа  $G^1$  в момент времени  $t_1$ . Полученный результат представлен на рисунке 2.

Представим эти два состояния сети в виде двух «снимков» динамического графа, для удобства не визуализируя узлы-сенсоры (рисунок 3). Тогда в терминах модели,  $DG = (S^0, S^1)$ . Рассмотрим «снимок»  $S^1 : S^1 = (G^1, Op^0, t_1)$ . Здесь из четырех возможных унарных операций над статическим графом  $G^0$  выполнены только две: добавление и удаление дуг. Распишем  $Op^0$ :

$$Op^0 = ((VA, VD, EA, ED), \{\}, \{\{(v_8, v_9), (v_9, v_D)\}, \{(v_9, v_7), (v_8, v_7)\}\});$$

$$Op^0 = ((0, 0, 1, 1), \{\}, \{\{(v_8, v_9), (v_9, v_D)\}, \{(v_9, v_7), (v_8, v_7)\}\}).$$

Характеристики графов:  $Str(G^0) = ((1, 84), 12, 84, 81)$ ,  $Str(G^1) = ((1, 84), 11, 84, 81)$ , видим, что изменился диаметр графа, несмотря на то, что число ребер в графе осталось прежним.

Значимыми вершинами в данном случае будут все 12 супер-узлов, и для них будут меняться только показатели центральности, поскольку мощность  $Pow$  в данном случае одинакова и постоянна. Так, изменились степени входа вершин 7 и 9: для «снимка» графа  $G^0$ :  $deg(v_7) = 5$ ,  $deg(v_9) = 4$ , для «снимка» графа  $G^1$  степени входа:  $deg(v_7) = 3$ ,  $deg(v_9) = 5$ , степени выхода остались прежними.

Следует отметить, что альтернативные пути обмена данными между супер-узлами (пунктирная линия) не учитывались при вычислении степеней, учитывались только связи с сенсорами и активные связи (прямая линия) между супер-узлами.

Вес дуг в данном примере предполагаются одинаковыми, а анализ временных рядов, характеризующих обмен данными между узлами сети, выходит за рамки данной статьи и здесь не рассматривается.

Таким образом, представленная модель, базирующаяся на математическом аппарате динамической теории графов, описывает функционирование адаптивной сетевой топологии и предоставляет основу для разработки методов и алгоритмов мониторинга и контроля состояния системы. К таким алгоритмам и относится разработанный алгоритм предсказания связей, детально описанный в следующей главе.

### 3. Постановка задачи разработки алгоритма предсказания связей в одноранговых сенсорных сетях

Для создания алгоритма предсказания связей между узлами сети с учетом специфики, накладываемой способностью сети к саморегуляции, необходимо:

1. Задать условия, при возникновении которых в сети должен быть запущен процесс саморегуляции. Эти условия будут исходить из нарушений в целевой функции системы, определение которой дано далее.
2. Выбрать параметры, которые на практике будут наиболее значимыми при выполнении сетью саморегуляции.
3. Разработать алгоритм предсказания связей, работа которого инициируется при возникновении хотя бы одного условия из п. 1, и базирующийся на выбранных в п. 2 параметрах.

#### 3.1. Целевая функция системы

Поскольку рассматриваемые сети за счет своей адаптивной топологии способны к саморегуляции, этот процесс будет направлен на реализацию целевой функции системы. Определим понятие целевой функции (ЦФ) системы, исходя из чего сформулируем условия для выполнения саморегуляции.

Сеть с адаптивной топологией рассматривается как инструмент реализации ЦФ той КФС, которая построена на ее основе. Предполагается, что любая КФС обладает ЦФ, выражаемой множеством процессов, которые обязательно реализует рассматриваемая система. При этом каждый такой процесс характеризуется определенными диапазонами значений для характеристик узлов сети и процессов передачи данных.

Для различных типов сетей с адаптивной топологией ЦФ будет варьироваться. Если рассматривать только сети, узлы которых не меняют своего положения в пространстве, их можно разделить на 3 группы:

1. Одноранговые сенсорные сети — включают множество однотипных маломощных датчиков (сенсоров), выполняющих функции измерения, приема и передачи данных.
2. Гетерогенные сенсорные сети — помимо маломощных датчиков, включают так называемые «суперузлы», обладающие мощным сигналом и способностью к агрегации и обработке дан-

ных, поступающих от множества сенсоров. Рассмотренная в п. 1.2 сеть относится к данному типу.

3. Промышленные сети — обладают выраженной иерархической структурой, включают несколько типов устройств, варьирующихся по своему функционалу и сложности. Возможно перераспределение нагрузки между более сложными узлами (концентраторами, хабами и т. д.).

Представленное исследование касается саморегуляции только одноранговой сетевой топологии, выполняющей простейшую ЦФ — сбор и передачу данных от всех сенсоров от одного конца сети до другого, подключенного к системе хранения данных. Далее будем считать, что число вершин  $n$  в графе не меняется. Также будем рассматривать граф как неориентированный, в связи с тем, что каждый узел выполняет одинаковые функции и способен как принимать, так и отправлять данные, возможность этого определяется только нахождением в заданном радиусе с соседними узлами.

Ребро  $e_{ij}$  между парой вершин  $v_i, v_j$  существует только в том случае, если они удалены друг от друга на расстояние  $d(v_i, v_j)$ , не большее  $r$ ,  $d(v_i, v_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \forall v_i, v_j \in V, (x_i, y_i), i = (1, N)$ , где  $(x_i; y_i)$  и  $(x_j; y_j)$  — координаты вершин  $v_i$  и  $v_j$  в двумерном евклидовом пространстве, соответственно.

Такая ЦФ характерна для сенсорных сетей, расположенных в труднодоступных местах, где ключевая задача — собрать информацию и передать ее в хранилище данных для дальнейшего анализа [28, 29].

Выразим ЦФ в виде набора следующих условий, при возникновении которых в сети должен быть запущен процесс саморегуляции:

1. Граф  $G$  является связным:  $\nexists v_i : deg(v_i) = 0$ .
2. Все вершины графа функционально эквивалентны и выполняют одинаковые функции  $F = measure, send, receive$  — измерение, отправку (измеренного значения или значений, полученных от соседних узлов) и прием измеренных значений от соседних узлов, находящихся на расстоянии  $r : \forall i : f(v_i) = (1, 1, 1)$ . Здесь  $f$  — оператор, сопоставляющий каждой вершине графа двоичный вектор, соответствующий операциям множества  $F$ : 1 — если вершина выполняет данную функцию, 0 — в противном случае.
3. Распределение степеней графа  $G$ , моделирующего сетевую структуру рассматриваемой КФС, близко к равномерному. Такое предположение выдвинуто из-за предполагаемого равномерного покрытия сенсорами всей заданной области и во избежание перегрузки некоторых узлов, что характерно в случае с реализацией сетевых атак. Например, атаки отказа в обслуживании и атаки типа «черная дыра» (sinkhole) характеризуются увеличением степени вершин графа. Для этого зададим максимальную степень вершин графа  $deg_{max}(v_i), i = (1, n)$ .

Триггером для выполнения саморегуляции сети является нарушение хотя бы одного из вышеописанных условий.

Важно отметить, что вышеописанные условия являются обязательными для саморегуляции, однако в зависимости от типа КФС, саморегуляция может быть выполнена и при наличии дополнительных факторов, таких как снижение заряда батареи устройства или уровня его сигнала. Представленный алгоритм описан только для трех ключевых условий.

### 3.2. Формальная постановка задачи

Задача состоит в разработке алгоритма *Pred* предсказания связей между узлами неориентированного графа, узлы которого функционально эквивалентны и выполняют одинаковый набор функций.

Определим условия для запуска алгоритма *Pred*:

1.  $\exists v_i : deg(v_i) = 0$ .

2.  $\exists v_i : f(v_i) \neq (1, 1, 1)$ .
3.  $\nexists v_k : deg(v_k) = deg_{max}(v_i)$ .

На вход алгоритм *Pred* принимает следующие параметры:

- текущий «снимок» динамического графа  $S^{step}$ ;
- множество *Vrtx* значимых параметров вершин графа, моделирующего сеть;
- максимальную степень вершин графа  $deg_{max}(v_i)$ ;
- радиус установления связи между узлами  $r$ ;
- пороговое значение *threshold* для принятия решения о добавлении/удалении ребра.

Разрабатываемый алгоритм должен быть таким, что спрогнозированный «снимок» динамического графа на шаге  $step + 1$  максимально соответствует реальному «снимку» графа на шаге  $step + 1$ . Отклонение от предсказания вычисляется на основе симметрической разности множеств ребер для спрогнозированного и реального «снимков», результатом которой является множество, состоящее только из тех ребер, которые входят ровно в одно из множеств. Мощность множества и представляет собой отклонение от предсказания.

$$Pred : Pred(S^{step}, Vrtx, deg_{max}(v_i), r) \rightarrow S_{pred}^{step+1}, S_{pred}^{step+1} : |S_{pred}^{step+1} - S^{step+1}| = 0.$$

Учитывая, что рассматривается случай с постоянным числом узлов в сенсорной сети, разрабатываемый алгоритм *Pred* должен предсказывать только появление или исчезновение ребер в графе. Целью настоящего исследования является разработка такого алгоритма *Pred*, чтобы он выполнял эту функцию на основе вычисления вероятностей наличия ребра между заданной парой вершин.

Требования, накладываемые на алгоритм:

1. Алгоритм работает с неориентированным графом.
2. Для повышения скорости работы алгоритма, он не реализует пересчет вероятностей для ребер, инцидентных вершинам, которые не подпадают под условия саморегуляции.
  - 2.1 В случае с появлением изолированной вершины, алгоритм вычисляет вероятности появления ребер между этой вершиной и ее ближайшими соседями, и добавляет в порождаемый им граф  $S_{pred}^{step+1}$  ребро, для которого вероятность была наибольшей.
  - 2.2 В случае, когда вершина  $v_i$  графа не выполняет одну или несколько из своих функций, алгоритм *Pred* работает для множества вершин, смежных с  $v_i$ . Так, при проблемах с приемом и передачей данных от соседних узлов, саморегуляция сети будет заключаться в построении альтернативных маршрутов, обеспечивающих передачу данных в нужном направлении.
  - 2.3 В случае с возникновением в сети «перегруженных» узлов, саморегуляция сети будет заключаться в попытке приблизить распределение степеней графа к равномерному за счет удаления ребер, инцидентных «загруженной» вершине  $v_i$  и созданию новых ребер для некоторого подмножества вершин, смежных с  $v_i$ . Поэтому алгоритм *Pred* будет работать для некоторого случайно выбранного подмножества вершин, инцидентных  $v_i$ .
3. Для работы алгоритма требуется обученная модель логистической регрессии. Задача предсказания связей трактуется как задача бинарной классификации, где в одном классе собраны ребра, «рекомендованные» алгоритмом к построению в будущем графе, а во втором классе — ребра, которых, по прогнозу алгоритма, не будет в будущем графе. Выбор логистической регрессии обоснован ее эффективностью и удобством с точки зрения генерации значений вероятности принадлежности объекта к классу.

Выбор параметров, которые на практике будут наиболее значимыми при выполнении сетью саморегуляции, зависит от типа сети. В случае с сенсорной сетью, узлы которой неподвижны в пространстве, значимыми предполагается следующее множество характеристик вершин *Vrtx*:

- мощность сигнала устройства сети,  $Pow_{v_i}$ ;
- заряд батареи узла  $Bat_{v_i}$ ;
- возможности узла по обмену данными, выражаемые значением метрики центральности по степени (степень вершины)  $Cent_1^{v_i}$ ;
- критичность узла, выражаемая метрикой центральности по посредничеству  $Cent_2^{v_i}$ ;
- пороговое значение  $threshold$  для принятия решения о внесении изменений в матрицу смежности графа.

Следует учитывать, что адаптивная сетевая топология является основой для функционирования некоторой КФС, а такие системы отличаются от информационных и компьютерных наличием взаимосвязанных информационной и физической составляющих. Поэтому, на взгляд автора, отнести к значимым параметрам вершин как физические характеристики узлов (мощность сигнала и заряд батареи), так и характеристики, важные для информационного обмена (метрики центральности).

## 4. Описание алгоритма в виде псевдокода

### 4.1. Появление изолированной вершины

Алгоритм вычисляет всех соседей изолированной вершины  $v_i$ , с использованием логистической регрессии определяет вероятность появления ребра между  $v_i$  и каждым ее соседом, выбирает ребро, вероятность появления которого максимальна, добавляет его к текущему графу и присваивает получившийся граф графу для следующего шага  $S^{step+1}$ .

START

READ входные параметры ( $S^{step}$ ,  $V_{rtx}=\{Pow_{v_i}, Bat_{v_i}, Cent_1^{v_i}, Cent_2^{v_i}\}$ ,  $deg_{max}(v_i)$ ,  $r$ ,  $threshold$ )

COMPUTE множество  $V^x$  вершин, находящихся на расстоянии, не большем чем  $r$ , от  $v_i$

FOR каждой вершины из множества  $V^x$

    SAVE параметры  $Pow$ ,  $Bat$ ,  $Cent_1$ ,  $Cent_2$

    COMPUTE вероятность  $P(e_{ij})$  появления ребра между вершиной  $v_i$  и текущей вершиной

    COMPUTE ребро с максимальным значением вероятности

    ADD ребро с максимальным значением вероятности к «снимку» графа  $S^{step}$

    SET  $S^{step+1} \leftarrow S^{step}$

END

### 4.2. Нарушение в работе вершины

Для вершины  $v_i$ , функционирование которой нарушено, алгоритм определяет все вершины, смежные с ней, на основе матрицы смежности графа  $A$ . Для каждой такой вершины он определяет множество соседних вершин, вычисляет вероятности появления ребра между текущей вершиной и каждой из соседних. На основе выбранных максимальных значений изменяет матрицу смежности, для выбранных узлов меняя значения 0 и 1 на соответствующие вероятности. Затем алгоритм вычисляет разность между изначальной и измененной матрицами смежности, и если абсолютное значение разности превышает некоторый заданный пользователем порог  $threshold$ , меняет в матрице смежности значение на противоположное ( 0 вместо 1 и наоборот). На основе обновленной матрицы смежности строится граф, соответствующий состоянию сети на следующем шаге  $S^{step+1}$ .

START

READ входные параметры ( $S^{step}$ ,  $V_{rtx}=\{Pow_{v_i}, Bat_{v_i}, Cent_1^{v_i}, Cent_2^{v_i}\}$ ,  $deg_{max}(v_i)$ ,  $r$ ,  $threshold$ )

INIT матрицу вероятностей  $PA$  появления ребер в графе, размером  $n \times n$

COMPUTE множество  $V^y$  вершин, смежных с  $v_i$

SET  $PA \leftarrow A$ , где  $A$  — матрица смежности графа

FOR каждой вершины из множества  $V^y$

COMPUTE множество  $V^z$  вершин, находящихся на расстоянии, не большем чем  $r$ , от текущей вершины  
 COMPUTE вероятность  $P(e_{yz})$  появления ребра между текущей вершиной и вершинами из множества  $V^z$   
 COMPUTE ребро с максимальным значением вероятности для текущей вершины  
 SET  $P(e_{yz}) \leftarrow$  значение вероятности, полученное на предыдущем шаге  
 COMPUTE  $\Delta \leftarrow A - PA$   
 FOR каждого элемента матрицы смежности  $a_{kj}$   
 IF  $|\Delta_{kj}| > threshold$  THEN  
 SET  $a_{kj} \leftarrow 1 - a_{kj}$   
 ENDIF  
 COMPUTE «снимок» графа  $S^{step+1}$  на основе полученной матрицы смежности  $A$   
 END

#### 4.3. Возникновение «перегруженной» вершины

Алгоритм работает аналогично случаю 2, с тем лишь отличием, что для «разгрузки» вершины  $v_i$  графа выбирается некоторое подмножество смежных с ней вершин, для которых в дальнейшем и идет перестроение.

START  
 READ входные параметры ( $S^{step}$ ,  $Vrtx = \{Pow_{v_i}, Bat_{v_i}, Cent_1^{v_i}, Cent_2^{v_i}\}$ ,  $deg_{max}(v_i)$ ,  $r$ ,  $threshold$ )  
 COMPUTE множество  $V^y$  вершин, смежных с  $v_i$   
 COMPUTE подмножество  $V^{y_1}$  множества  $V^y$   
 INIT матрицу вероятностей  $PA$  появления ребер в графе, размером  $n \times n$   
 SET  $PA \leftarrow A$ , где  $A$  – матрица смежности графа  
 FOR каждой вершины из множества  $V^{y_1}$   
 COMPUTE множество  $V^z$  вершин, находящихся на расстоянии, не меньшем чем  $r$ , от текущей вершины  
 COMPUTE вероятность  $P(e_{yz})$  появления ребра между текущей вершиной и вершинами из множества  $V^z$   
 COMPUTE ребро с максимальным значением вероятности для текущей вершины  
 SET  $P(e_{yz}) \leftarrow$  значение вероятности, полученное на предыдущем шаге  
 COMPUTE  $\Delta \leftarrow A - PA$   
 FOR каждого элемента матрицы смежности  $a_{kj}$   
 IF  $|\Delta_{kj}| > threshold$  THEN  
 SET  $a_{kj} \leftarrow 1 - a_{kj}$   
 ENDIF  
 COMPUTE «снимок» графа  $S^{step+1}$  на основе полученной матрицы смежности  $A$   
 END

Алгоритм выбора подмножества  $V^{y_1}$  базируется на том, что после определения соседних вершин для каждой загруженной вершины выполняется ранжирование соседних вершин по двум параметрам: степень и уровень заряда батареи. В приоритете для разрыва соединения оказываются вершины с невысокой степенью и высоким уровнем заряда батареи. Такой выбор основан на том, что для таких вершин процесс переподключения пройдет с меньшими потерями с точки зрения качества работы сети, чем для вершин, которые имеют большое число связей и при этом не обладают запасом заряда батареи.

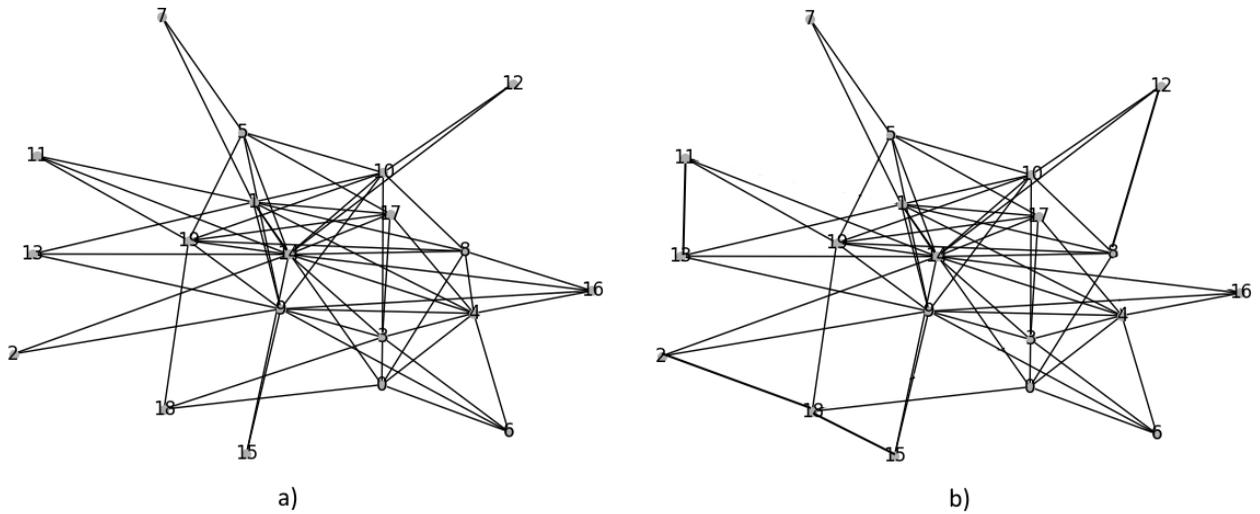


Fig. 4. 2 snapshots of a dynamic graph

Рис. 4. 2 «снимка» динамического графа

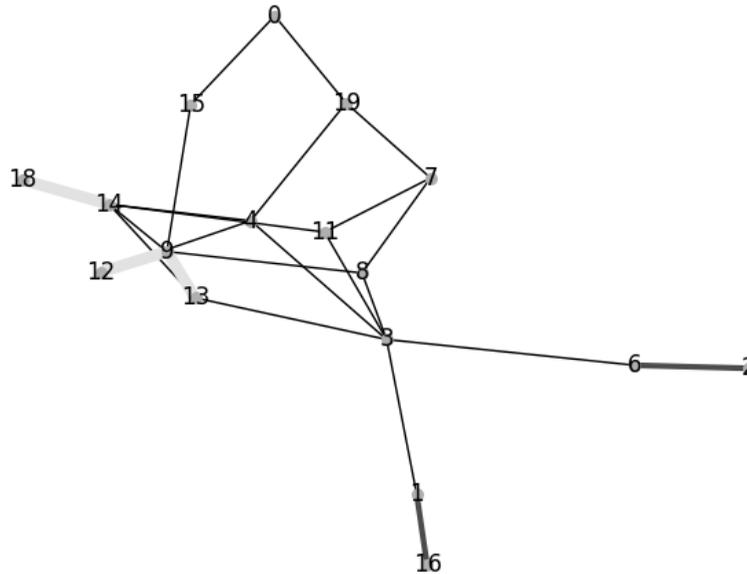


Fig. 5. Links prediction

Рис. 5. Предсказание связей в графе

## 5. Эксперименты по предсказанию связей в графе

Для проведения экспериментальных исследований в части оценки качества разработанного алгоритма на языке Python была смоделирована одноранговая сеть в виде неориентированного графа. Для визуального удобства размер сети составлял 20 узлов, пронумерованных от 0 до 19.

Смоделированная одноранговая сеть построена на базе разработанной модели функционирования сетей с адаптивной топологией. На рисунке 4 показаны 2 последовательных «снимка» а) и б) динамического графа, моделирующего одноранговую сеть.

Визуализация предсказанного разработанным алгоритмом графа представлена на рисунке 5. Жирные светлые линии обозначают появление связей в графе на следующем шаге, а жирные темные линии – исчезновение связей между вершинами на следующем шаге.

Стартом к запуску алгоритма предсказания ребер, помимо трех ключевых условий саморегуляции, являлись также:

- снижение уровня заряда батареи узла более чем на 5 %;
- ухудшение мощности сигнала больше чем на 10 %.

В условиях сети с маломощными сенсорами, на взгляд автора, показатель заряда батареи является более значимым для сохранения работоспособности сети, чем мощность сигнала, на которую могут влиять посторонние факторы. Исходя из этого, были заданы соответствующие пороговые значения для изменений: 5 % и 10 %, соответственно.

Модель логистической регрессии, прогнозирующая вероятность появления или исчезновения ребра на следующем шаге, была обучена на 10 000 «снимках» графа, показатель  $AUC$  для логистической регрессии составил 95,8 %. Подобранные моделью коэффициенты показали, что наиболее значимым признаком из всех является уровень заряда батареи устройства.

Для оценки качества работы алгоритма были смоделированы случаи появления изолированных и «загруженных» вершин. Пороговое значение  $threshold$  было выбрано эмпирически,  $threshold = 0.7$ .

### 5.1. Появление изолированных вершин

На графе было смоделировано появление изолированных вершин под номерами 2 и 8, при одновременном снижении заряда батареи у вершины 19 на 4 %.

Результаты работы алгоритма представлены на рисунке 6.

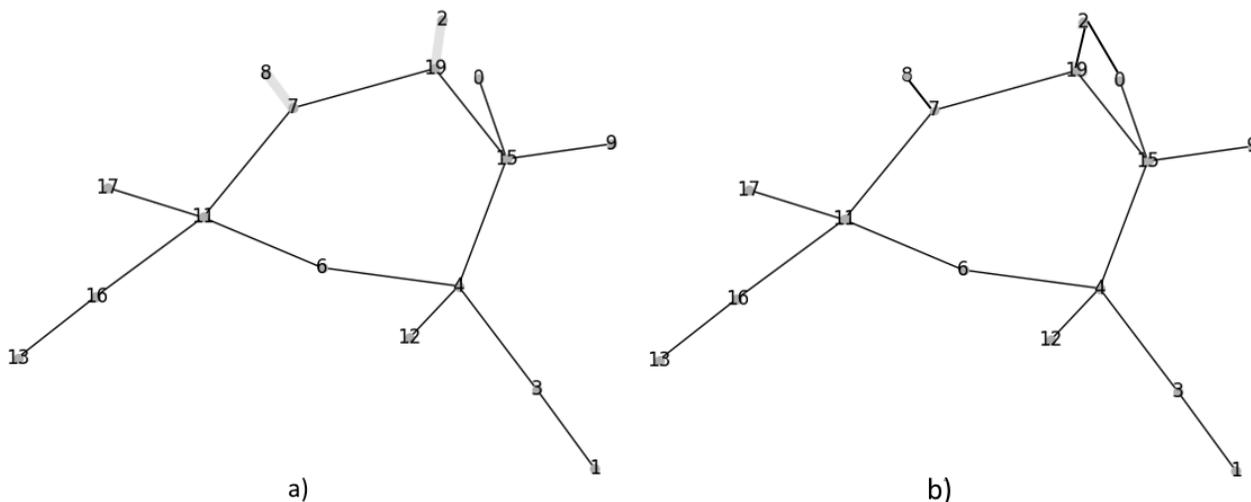


Fig. 6. Links prediction

Рис. 6. Предсказание связей в графе

Алгоритм работал следующим образом:

1. Определил ближайших соседей изолированных вершин:
  - 1.1 Для вершины 8: вершина 7.
  - 1.2 Для вершины 2: вершины 0 и 19, оценка расстояния  $r$  показала, что вершина 19 расположена чуть ближе к вершине 2.
2. Использовал обученную модель логистической регрессии для вычисления вероятностей появления ребер (0, 2) и (0, 19) — они составили 0,98 и 0,87, соответственно.
3. Выбрал ребро с максимальным значением вероятности — между вершинами (0, 2).
4. Добавил к текущему «снимку» графа ребра (7, 8) и (0, 2).
5. Присвоил полученный на шаге 4 граф графу для следующего шага.

Из визуализации работы смоделированной сети, действительно, видно что оба спрогнозированных ребра появились на следующем шаге.

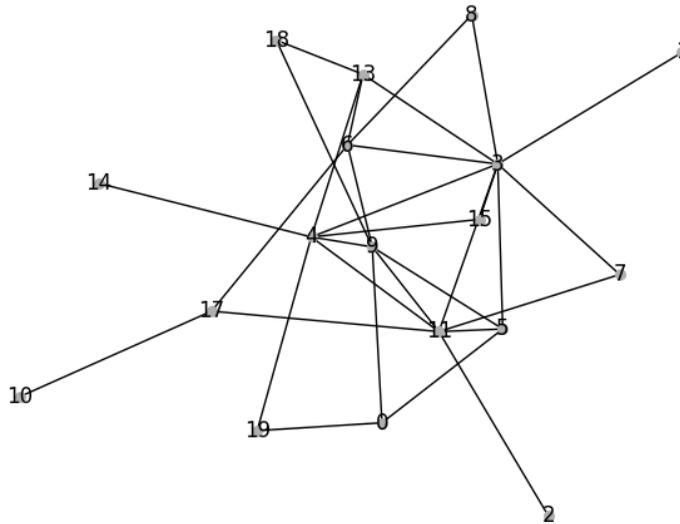


Fig. 7. Network with "overloaded" vertices

Рис. 7. Сеть с «перегруженными» вершинами

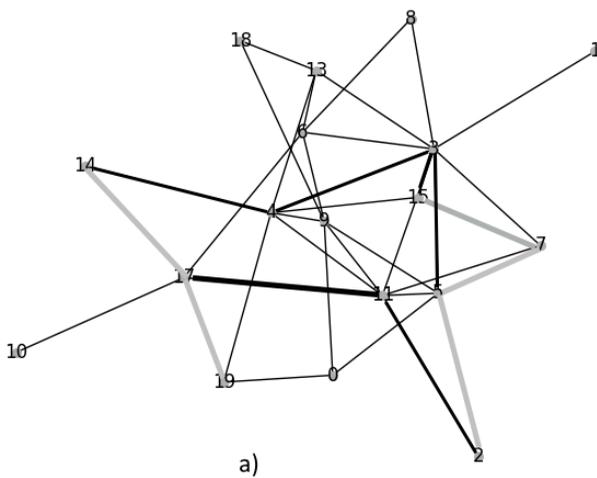


Fig. 8. Links prediction

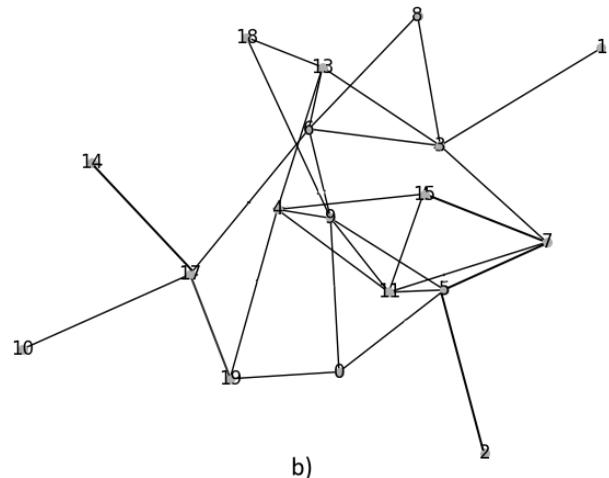


Рис. 8. Предсказание связей в графе

## 5.2. Возникновение «загруженных» вершин

Было смоделировано появление вершин с более высокой степенью, чем у остальных. Это были вершины под номерами 3, 4 и 11. Изначально,  $\text{deg}(v_3) = 8$ ,  $\text{deg}(v_4) = 7$ ,  $\text{deg}(v_{11}) = 7$ . «Снимок» сети представлен на рисунке 7.

Результат работы алгоритма представлен на рисунке 8.

Алгоритм работал следующим образом:

1. Определил «загруженные вершины»: {3, 4, 11}.
2. Определил множество соседних вершин:
  - 2.1 Для вершины 3: {1, 5, 4, 6, 7, 8, 13, 15}.
  - 2.2 Для вершины 4: {3, 9, 11, 13, 14, 15, 19}.
  - 2.3 Для вершины 11: {2, 4, 5, 7, 9, 15, 17}.
3. Для каждой вершины из п.1 выбрал по 2 вершины, которые будут «переключены» на другие:
  - 3.1 Для вершины 3: {5, 15}.

- 3.2 Для вершины 4: {3, 14}.
  - 3.3 Для вершины 11: {2, 17}.
  4. Удалил ребра (3, 4), (3, 5), (3, 15), (4, 14), (2, 11), (11, 17).
  5. Для каждой из множества вершин {2, 5, 14, 15}:
    - 5.1 Определил ближайших соседей.
    - 5.2 Использовал обученную модель логистической регрессии для вычисления вероятностей появлений ребер.
    - 5.3 Выбрал ребро с максимальным значением вероятности, инцидентное текущей вершине.
  6. Сохранил множество ребер, которые будут добавлены: (5, 7), (7, 15), (14, 17), (2, 5) и (17, 19).
  7. Создал матрицу вероятностей путем копирования значений из матрицы смежности и обновления значений в ячейках, соответствующих вершинам, инцидентным ребрам, полученным на шаге 5.
  8. Для множества ребер, полученных на шаге 5, вычислил разность матриц смежности и вероятностей.
  9. Для ячеек матрицы, полученной на шаге 7, в которых абсолютное значение было меньше либо равно *threshold*:
    - 9.1 Установил значения, противоположные значениям матрицы смежности.
    - 9.2 Остальные значения установил в соответствии с матрицей смежности.
  10. Сгенерировал граф состояния сети на следующем шаге по матрице, полученной на шаге 8.2.
- Видно, что результаты работы алгоритма соответствуют поведению смоделированной сети. В результате саморегуляции, максимальная степень вершин составила 6 (у одной вершины под номером 9), и в распределении степеней вершин графа исчезли «всплески».

Таким образом, можно сделать вывод об адекватности разработанного алгоритма предсказания связей и о его соответствии реальному поведению узлов в самоорганизующейся сети.

Следует отметить, что данный алгоритм может быть незначительно доработан для предсказания связей в гетерогенных сенсорных и промышленных сетях. Тогда необходимо будет учитывать:

- типы узлов сети — они повлияют на возможность создания связей (например, типы узлов могут быть такими, что связь между узлами данных типов невозможна) и будут иметь разный приоритет при предсказании связей;
- контекст функций, выполняемых узлами — так, одно и то же ребро между узлами разных типов может иметь разное значение (отправка команды или передача данных);
- специфику, накладываемую ЦФ — это характерно для промышленных сетей, для них ЦФ может быть выражена как упорядоченная последовательность ребер, и сохранение упорядоченности также окажет влияние на результат прогноза.

## Заключение

В работе представлена обобщенная графовая модель, описывающая функционирование беспроводных сетей с адаптивной топологией. Выбранный для моделирования математический аппарат динамической теории графов позволяет не только адекватно описать все изменения, происходящие в таких сетях, но и получить нужные сведения для контроля и мониторинга состояния сети.

На базе разработанной модели предложен алгоритм предсказания связей между узлами сети с адаптивной топологией, ограниченный случаем с одноранговыми сетями. Проведенные исследования связанных работ продемонстрировали, что современные исследования в основном не учитывают одновременно информационную и физическую природу киберфизических систем, и, следовательно, динамических сетей, лежащих в их основе.

Предложенный алгоритм устраняет этот недостаток, поскольку он в полной мере базируется на разработанной графовой модели и использует как физические (уровень сигнала и заряда ба-

тарей), так и информационно-структурные характеристики вершин одноранговой сети (метрики центральности по степени и посредничеству). В основе алгоритма лежит обученная автором модель логистической регрессии, показатель AUC которой составил 95,8 %, что говорит о довольно высоком качестве классификации. Визуализация работы алгоритма и последующее сравнение с полученным на следующем шаге состоянием сети продемонстрировали его соответствие принципам работы сенсорных сетей.

## References

- [1] V. Student and R. Dhir, “A study of ad hoc network: A review”, *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 3, pp. 135–138, 2013.
- [2] V. Amulya, “Cyber physical systems by using wireless sensor networks”, *International Journal of Science and Research*, vol. 7, no. 4, pp. 1380–1396, 2018.
- [3] V. L. Narasimhan, A. A. Arvind, and K. Bever, “Greenhouse asset management using wireless sensor-actor networks”, in *International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM’07)*, IEEE, 2007, pp. 9–14.
- [4] T. Taleb, D. Bottazzi, M. Guizani, and H. Nait-Charif, “Angelah: A framework for assisting elders at home”, *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 4, pp. 480–494, 2009.
- [5] P. Mohan, V. N. Padmanabhan, and R. Ramjee, “Nericell: Rich monitoring of road and traffic conditions using mobile smartphones”, in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, 2008, pp. 323–336.
- [6] A. Thiagarajan *et al.*, “Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones”, in *Proceedings of the 7th ACM conference on embedded networked sensor systems*, 2009, pp. 85–98.
- [7] S. Mathur *et al.*, “Parknet: Drive-by sensing of road-side parking statistics”, in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, 2010, pp. 123–136.
- [8] J. Wang, Z. Li, M. Li, Y. Liu, and Z. Yang, “Sensor network navigation without locations”, *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1436–1446, 2012.
- [9] E. Y. Pavlenko, “Functional model of adaptive network topology of large-scale systems based on dynamical graph theory”, *Automatic Control and Computer Sciences*, vol. 56, no. 8, pp. 1016–1024, 2022.
- [10] Z. Stanfield, M. Coşkun, and M. Koyutürk, “Drug response prediction as a link prediction problem”, *Scientific reports*, vol. 7, no. 1, p. 40 321, 2017.
- [11] T. J. Lakshmi and S. D. Bhavani, “Link prediction approach to recommender systems”, *Computing*, 2023. DOI: [10.1007/s00607-023-01227-0](https://doi.org/10.1007/s00607-023-01227-0).
- [12] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, “A review of relational machine learning for knowledge graphs”, *Proceedings of the IEEE*, vol. 104, no. 1, pp. 11–33, 2015.
- [13] N. N. Daud, S. H. Ab Hamid, M. Saadoon, F. Sahran, and N. B. Anuar, “Applications of link prediction in social networks: A review”, *Journal of Network and Computer Applications*, vol. 166, p. 102 716, 2020.
- [14] Z. Qiu, J. Wu, W. Hu, B. Du, G. Yuan, and P. Yu, “Temporal link prediction with motifs for social networks”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 3, pp. 3145–3158, 2023.
- [15] H. Nassar, A. R. Benson, and D. F. Gleich, “Neighborhood and pagerank methods for pairwise link prediction”, *Social Network Analysis and Mining*, vol. 10, no. 1, p. 63, 2020.
- [16] X.-H. Yang, X. Yang, F. Ling, H.-F. Zhang, D. Zhang, and J. Xiao, “Link prediction based on local major path degree”, *Modern Physics Letters B*, vol. 32, no. 29, pp. 1 850 348–306, 2018.

- [17] S. Kumar, A. Mallik, and B. Panda, "Link prediction in complex networks using node centrality and light gradient boosting machine", *World Wide Web*, vol. 25, no. 6, pp. 2487–2513, 2022.
- [18] J. Cheriyan and G. Sajeev, "m-PageRank: A novel centrality measure for multilayer networks", *Advances in Complex Systems*, vol. 23, no. 5, p. 2 050 012, 2020.
- [19] F. Feng, X. Liu, B. Yong, R. Zhou, and Q. Zhou, "Anomaly detection in ad-hoc networks based on deep learning model: A plug and play device", *Ad Hoc Networks*, vol. 84, pp. 82–89, 2019.
- [20] R. Meddeb, F. Jemili, B. Triki, and O. Korbaa, "Anomaly-based behavioral detection in mobile ad-hoc networks", *Procedia Computer Science*, vol. 159, pp. 77–86, 2019.
- [21] M. Zhang and Y. Chen, "Link prediction based on graph neural networks", in *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, 2018, pp. 5165–5175.
- [22] B. P. Chamberlain *et al.*, *Graph neural networks for link prediction with subgraph sketching*, 2023. arXiv: [2209.15486](https://arxiv.org/abs/2209.15486) [cs.LG].
- [23] M. Niu, L. Liu, and J. Shu, "Link quality prediction for wireless networks: Current status and future directions", in *Proceedings of the 2023 8th International Conference on Intelligent Information Technology*, 2023, pp. 52–56.
- [24] J. Chen, Y. Han, D. Li, and J. Nie, "Link prediction and route selection based on channel state detection in UASNs", *International Journal of Distributed Sensor Networks*, vol. 7, no. 1, p. 939 864, 2011.
- [25] H. Shao, L. Wang, H. Liu, and R. Zhu, "A link prediction method for MANETs based on fast spatio-temporal feature extraction and LSGANs", *Scientific Reports*, vol. 12, no. 1, p. 16 896, 2022.
- [26] F. Harary and G. Gupta, "Dynamic graph models", *Mathematical and Computer Modelling*, vol. 25, no. 7, pp. 79–87, 1997.
- [27] L. K. Ketshabetswe, A. M. Zungeru, M. Mangwala, J. M. Chuma, and B. Sigweni, "Communication protocols for wireless sensor networks: A survey and comparison", *Heliyon*, vol. 5, no. 5, E01591, 2019.
- [28] D. Kandris, C. Nakas, D. Vomvas, and G. Koulouras, "Applications of wireless sensor networks: An up-to-date survey", *Applied system innovation*, vol. 3, no. 1, p. 14, 2020.
- [29] M. Kim, S. Park, and W. Lee, "Energy and distance-aware hopping sensor relocation for wireless sensor networks", *Sensors*, vol. 19, no. 7, p. 1567, 2019.