

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ БЮДЖЕТНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ
ФЕДЕРАЦИИ»

Факультет информационных технологий и анализа больших данных
Департамент анализа данных и машинного обучения

Выпускная квалификационная работа

на тему: «Разработка алгоритма прогнозирования социально-экономического
процесса на основе когнитивного графа»


Направление подготовки: 01.03.02 Прикладная математика и информатика

Профиль: Анализ данных и принятие решений в экономике и финансах

Выполнил студент учебной группы
ПМ19-2


Гераськина Надежда Станиславовна 

Научный руководитель работы
д.э.н., профессор

Коровин Дмитрий Игоревич 

**ВКР соответствует предъявленным
требованиям**

Руководитель Департамента анализа данных
и машинного обучения, к.т.н., доцент

 Д.А. Петросов

«19» мале 2023 г.

Москва 2023

Содержание

Введение	2
Глава 1. Теоретические аспекты исследования социально-экономических систем.....	5
1.1. Социально-экономические системы.....	5
1.2. Методы прогнозирования социально-экономических процессов	7
1.3. Когнитивные карты и графы	12
Глава 2. Сценарии рискованных событий	17
Глава 3. Практическая реализация алгоритма предсказания социально-экономического процесса.....	38
3.1. Рекуррентные нейронные сети для предсказания временных рядов.....	38
3.2. Разработка алгоритма прогнозирования социально-экономического процесса на основе когнитивного графа	42
3.3. Принятия решений на основе алгоритма прогнозирования социально-экономического процесса	54
Заключение	56
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ИНТЕРНЕТ-РЕСУРСОВ	
58	
ПРИЛОЖЕНИЕ.....	63

Введение

Современный мир состоит из множества социально-экономических систем (СЭС), которые влияют на жизнь каждого человека ежедневно. Их целью является обеспечение устойчивого экономического и социального развития общества. Это достигается за счет регулирования деятельности политических, экономических и правовых структур, создания благоприятных условий для развития бизнеса и инноваций, а также за счет поддержки социальной справедливости и равенства возможностей.

Развитие социально-экономических систем в настоящее время активно обсуждается в свете изменений, происходящих в мировой экономике и политике. Важно учитывать интересы всех участников общества и стремиться к созданию устойчивых и эффективных моделей развития.

Однако, решение широкого круга задач, нацеленных на проблему устойчивого развития, повышения качества жизни, подготовки, принятия и обоснования важных стратегических решений, в настоящее время в Российской Федерации затруднено по ряду причин. Введение иностранных санкций в основных сферах развития российской экономики, невозможность быстрого перехода на отечественное ПО в ключевых отраслях и нестабильность текущей политической ситуации снижает инновационную активность бизнеса.

Также в современных условиях перестают корректно работать многие экономико-математические модели, носящие дискретный и статичный характер, что вызывает необходимость перехода на моделирование непрерывных и динамичных процессов, которое позволит осуществлять эффективное управление даже в кризисных и нестабильных ситуациях.

Прогнозирование социально-экономических процессов является важной задачей, которая позволяет принимать правильные решения в различных областях деятельности, таких как бизнес, экономика и политика. Для прогнозирования социально-экономических процессов используются различные методы, такие как статистические, эконометрические, методы машинного обучения и т. д. Однако, каждый из этих методов имеет свои ограничения и недостатки, что не позволяет

получить точный и надежный прогноз.

В данной работе предлагается формализовать определенную экспертную систему за счет использования когнитивных графов, а также построить и обучить модель машинного обучения, которая будет предсказывать значения на вершинах в когнитивном графе с учетом временного лага, что позволит эффективно учитывать цикличность социально-экономических процессов.

Таким образом, целью данной работы является повышение эффективности принятия решений при помощи алгоритма прогнозирования социально-экономического процесса на основе когнитивного графа. Для достижения этой цели были поставлены следующие задачи:

- изучение существующих методов прогнозирования социально-экономических процессов;
- построение когнитивного графа социально-экономического процесса;
- разработка алгоритма прогнозирования социально-экономического процесса на основе когнитивного графа;
- проверка эффективности разработанного алгоритма на примере синтетических данных;
- разработка методологии принятия решений на основе прогнозов и когнитивного графа.

Изучение существующих методов прогнозирования социально-экономических процессов позволит выявить недостатки и ограничения существующих методов и определить преимущества разработанного алгоритма.

Построение когнитивного графа будет подробно поэтапно описано на примере системы оценки эффективности мер государственной поддержки регионов и других объектов экономики. Когнитивный граф — это графическая модель, которая позволяет представить взаимосвязи между различными факторами, что помогает в прогнозировании различных событий и процессов. Когнитивный граф позволяет учитывать не только прямые взаимосвязи между факторами, но и косвенные зависимости. Это позволяет получить более точный и надежный прогноз.

Разработка алгоритма прогнозирования социально-экономического процесса

на основе когнитивного графа будет включать в себя генерацию синтетических данных на основе взаимосвязей, описанных на когнитивном графе, полученном на предыдущем этапе. Далее в алгоритм будет интегрирована рекуррентная нейронная сеть. Данные модели имеют несколько разновидностей, которые являются наиболее успешными в решении задач генерации текстовых данных, изображений, временных рядов, машинного перевода.

Алгоритм, разработанный в данной работе, может быть использован для прогнозирования различных социально-экономических процессов, таких как рост ВВП, инфляция, безработица и т. д. Полученные результаты могут быть использованы для разработки эффективных стратегий в различных областях деятельности.

Глава 1. Теоретические аспекты исследования социально-экономических систем

1.1. Социально-экономические системы

В современной литературе используют множество определений понятий социальная и экономическая система. Для полноценного понимания процессов, описываемых в данной работе необходимо ознакомиться с некоторыми из них.

Социальная система — это упорядоченное целое, включающее отдельных индивидов, группы, организации, институты, общности и сообщества, объединенные социальными связями и отношениями. Основными элементами социальной системы являются люди, а также их нормы, связи, отношения, взаимодействия. Общество как суперсистема предстает как совокупность систем (социальная группа, социальный институт, личность). [4]

Таким образом, в данной работе будут рассматриваться социальные системы, элементы которых связывают экономические отношения. Экономические отношения в первую очередь подразумевают некие, парные взаимодействия кого-то с кем-то в процессе производства, распределения, обмена и потребления благ и услуг [5]. Например, купля и продажа товаров, распределение бюджетных средств, доставка продукции в пункты продаж и т. д.

Социально-экономическая система — это целостная совокупность взаимосвязанных и взаимодействующих людей, социальных и экономических институтов (субъектов) и их отношений по поводу производства, распределения, обмена и потребления товаров, услуг, а также природных материальных и нематериальных ресурсов. [2] В социально-экономической системе возникают устойчивые связи и взаимоотношения между фирмами, социальными группами, людьми (экономическими субъектами), возникающие в процессе производства, распределения, обмена и потребления благ в отношении собственности.

Социально-экономические системы обладают свойством целостности, то есть

изменение одного компонента ведет к изменению системы в целом. Также им присуща иерархичность – одна система может являться подсистемой другой, более крупной системы. К тому же они интегративны, это означает, что системе присущи свойства, которых нет у компонентов по одиночке, и наоборот. [7] Развитие социально-экономических систем циклично, то есть для них характерна закономерность наступления и разрешения кризисов.

Социально-экономическая система имеет две основные тенденции в существовании: функционирование и развитие. Функционирование системы представляет собой процесс, направленный на поддержание ее жизнедеятельности, сохранение функций, которые определяют ее целостность и качественную определенность. В социально-экономической системе имеется потенциал (ресурс, капитал, власть), способствующий ее развитию. Развитием называют изменения, происходящие в системе, принимающие структурную, качественно-количественную форму, либо это изменения, отражающие характер функционирования [18].

В отличие от технических или физических систем, социально-экономическая система характеризуется не только некоторым равновесным состоянием и способностью возвращаться к нему при возникновении внешних возмущающих факторов, но и способностью эффективно использовать, адаптироваться и видоизменять ресурсы своего развития, непрерывно наращивая показатели своего положительного изменения.

При этом социально-экономическая система может использовать различные стратегии для достижения своих целей. Например, она может осуществлять диверсификацию своих ресурсов, чтобы увеличить свою устойчивость к внешним факторам, или сосредоточиться на развитии определенных секторов экономики для достижения максимальной эффективности.

В процессе нашего анализа мы выделяем пять больших групп факторов развития социально-экономических систем:

- политические: режимы, стили управления;
- экономические: налоги, система хозяйствования, инвестиции, рыночная конкуренция и др.;

- социальные: демографические, уровень образования, стабильность общественного климата, уровень развития социальной среды населенного пункта;
- технологические: развитие научного потенциала, наличие роста высокотехнологичных отраслей, инновации;
- информационно-психологические: воздействие средств массовой информации, пропаганда и информационные войны.

1.2. Методы прогнозирования социально-экономических процессов

Поскольку в данной работе будет построен алгоритм предсказания социально-экономического процесса, необходимо изучить существующие методы прогнозирования, чтобы выявить их недостатки и доработать их.

Для начала необходимо дать определение ключевым понятиям. Прогноз — это количественное, вероятностное, научно-обоснованное суждение о возможном будущем состоянии системы или явления и (или) о возможных альтернативах и сроках их реализации. [9] В свою очередь прогнозирование – это разработка прогноза. Прогнозирование осуществляется множеством методов: статистическими, экспертными и методами моделирования.

У прогнозирования социально-экономических процессов выделяют несколько основных функций:

- анализ процессов и тенденций;
- исследование связей социально-экономических явлений в развитии объекта прогнозирования в конкретных условиях в определенном периоде;
- выявление альтернативных путей развития;
- оценка последствий принимаемых решений.

Разработка прогнозов должна основываться на некоторых методологических принципах. Таковыми являются принцип альтернативности, наблюдательности, системности, непрерывности и согласованности. Они позволяют исследователю составить полноценное представление об анализируемом объекте и построить на основе полученных данных достоверный прогноз. Также немаловажным является

принцип сбалансированности и пропорциональности. Он подразумевает под собой соблюдение общеэкономических, межотраслевых, внутриотраслевых, территориальных и внешнеэкономических пропорций, а также согласованность частных прогнозов. [8]

Рассмотрим подробнее основные методы прогнозирования социально-экономических процессов.

Методы экспертных оценок основаны на выявлении и обобщении мнений экспертов. Данный метод не исключает субъективизм, что является его недостатком, однако он может оказаться эффективнее других методов в условиях ограниченных временных и финансовых ресурсов, а также он позволяет принимать решения в процессах, которые сложно формализовать или описать количественными метриками.

Наиболее известным примером метода экспертных оценок является метод Дельфи. Его суть заключается в том, чтобы с помощью серии последовательных коммуникаций с экспертами — опросов, интервью, мозговых штурмов — добиться максимального консенсуса при определении правильного решения. Анализ с помощью дельфийского метода проводится в несколько этапов, результаты обрабатываются статистическими методами. [10] Данный метод не подходит для оперативного планирования, но бывает эффективен для стратегического. Однако, метод Дельфи на протяжении полувека подвергается критике из-за сильных искажений результатов ввиду психологических факторов, влияющих на экспертов.

Методы аналогий также имеют широкое распространение при прогнозировании социально-экономических процессов. В данном случае высока вероятность когнитивной ошибки, когда аналогии на самом деле случайны и не являются действительными. Также стоит помнить, что данные методы работают только если выполнится условие неизменности прогнозного фона. Ярким примером метода математических аналогий является модель роста числа изобретений по аналогии с процессом биологического размножения.

Наиболее развивающимися в современном мире являются методы математического моделирования. Математическая модель — математическое

представление реальности, набор логических взаимосвязей между различными элементами системы. Выделяют несколько основных методов математического моделирования:

- методы математического программирования;
- эконометрические и статистические методы;
- методы принятия решений;
- методы исследования операций;
- методы имитационного моделирования ;
- методы нейросетевого моделирования;
- системно-динамические методы;
- методы оптимального управления;
- методы сетевого моделирования;
- методы матричного моделирования.

Наиболее популярными являются эконометрические и статистические методы. Они предполагают построение моделей временных рядов, их сравнение при помощи статистических показателей качества и отбор лучшей модели. Примерами являются регрессионные методы, методы скользящих средних, ARIMA, адаптивные методы сезонных явлений, модели авторегрессий и многие другие.

Недостатком эконометрических методов является невозможность предсказать неожиданные всплески, учесть цикличность в сложноорганизованных процессах и большой набор математических проверок, который требует специфичной квалификации человека, проводящего моделирование и валидацию, и время. Также одно из условий, на которое опирается эконометрическое моделирование, состоит в том, что функциональное соотношение не меняется в течение рассматриваемого периода, что в некоторых моделях социально-экономических явлений недопустимо, например, в период кризисов. Более того, часто для корректного эконометрического и статистического моделирования необходим большой набор данных, что практически невозможно при изучении новообразующихся систем.

Также популярными проблемами эконометрического моделирования являются

мультиколлинеарность, то есть высокая корреляция между зависимыми переменными, и ложная регрессия, то есть регрессия, которая дает вводящие в заблуждение статистические доказательства линейной взаимосвязи между независимыми нестационарными переменными. Они могут привести к искажению точности прогнозов и осложнить интерпретацию результатов.

В современном мире получили огромную популярность и развитие нейросетевые методы и методы машинного обучения во всех областях жизни человека, в том числе в прогнозировании социально-экономических процессов. Машинное обучение основано на изучении статистических моделей, которые компьютерные системы используют для выполнения конкретной задачи без использования явных инструкций, вместо этого полагаясь на шаблоны и умозаключения. Оно рассматривается как часть более широкого понятия - искусственного интеллекта. Алгоритмы машинного обучения строят математическую модель на основе обучающего набора данных для того, чтобы делать прогнозы или принимать решения без явного программирования для выполнения задачи. В данном случае решается несколько проблем эконометрических методов: нейросетевые модели и алгоритмы машинного обучения построены на основе математических моделей, которые самостоятельно подстраиваются под специфику подаваемых данных, выявляют скрытые взаимосвязи. Эффективность моделей оценивается эмпирически на огромных количествах примеров и тестов.

Эконометрические модели – параметрические, они зависят от определённого набора параметров, который чаще всего ограничен вычислительными мощностями. В свою очередь, в машинном обучении часто создаются непараметрические модели, основанные исключительно на данных (не делается никаких базовых предположений о распределении), а используемые гиперпараметры оптимизируются с помощью многочисленных быстродействующих алгоритмов оптимизации. Также модели машинного обучения быстрее адаптируются к изменениям в данных, чем эконометрические модели.

Недостатком нейросетевых моделей и моделей машинного обучения является «черный ящик» работы алгоритмов. Для выявления, как был получен тот или иной

результат, необходимо глубокое понимание работы алгоритмов, что сокращает круг лиц, которые могут принимать решения и снижает интерпретируемость модели, что критично в моделировании социально-экономических процессов. Также необходимо проводить тщательную предобработку данных, убирать выбросы, проверять достоверность данных, так как из злого умысла они могут быть испорчены. К тому же, не всегда реально решить проблему лжекорреляции даже в машинном обучении.

Итак, мы видим, что существует множество методов прогнозирования социально-экономических процессов, каждый из которых имеет свои достоинства и недостатки. Логичным решением будет взять положительные черты некоторых методов и объединить их для предотвращения пагубных последствий каждого из них по отдельности.

В данной работе будет использоваться один из методов имитационного моделирования на основе когнитивных графов и рекуррентных нейронных сетей. Разберем подробнее, в чем особенность данного вида моделирования. Имитационное моделирование – это частный случай математического моделирования, возникающий в ситуациях, когда невозможно или неэффективно разрабатывать аналитические модели. Такие ситуации возникают, когда дорого или невозможно экспериментировать на реальном объекте, необходимо имитировать поведение системы во времени (появляются временные лаги), а также когда невозможно построить аналитическую модель по причине того, что в системе учитываются:

- причинные связи;
- нелинейности;
- стохастические (случайные) переменные[19].

Таким образом, выбранные методы позволят учесть цикличность, которая влечет за собой появление временных лагов, и сбалансированность социально-экономических процессов и предсказать взаимосвязи между ними при помощи быстрых вычислений нейросетевого моделирования. Также данная модель хорошо интерпретируема, так как когнитивный граф (карта) является одним из самых эффективных методов описания и изображения какого-либо процесса в системе.

1.3. Когнитивные карты и графы

Познакомимся с основными понятиями теории графов.

Граф G определяют как совокупность двух множеств: непустого множества вершин $V(G)$ и множества ребер $E(G)$. При этом вершиной (узлом, точкой) v называется некоторая фундаментальная единица, образующая графы, а ребром (линией) e называется пара вершин $e = (v, u)$, называемых связанными этим ребром или концевыми вершинами ребра. При этом говорят, что вершин инциденты этому ребру. [11]

Ребра, имеющие направления, называются **дугами** и обозначаются $a = (v, u)$. Направление дуги на рисунках обозначаются стрелками (от начала к концу соответственно). Граф, в котором все ребра являются ориентированными, называется **ориентированным графом (орграфом)**. Если в графе присутствуют и ориентированные, и неориентированные ребра, то его называют **смешанным**. [11]

Путем между двумя вершинами u и v в графе называют конечную последовательность вершин $u = e_0, e_1, \dots, e_n = v$, в которой каждые два соседних элемента соединены существующим ребром (a_i, a_{i+1}) , где $i \in (0, n)$. Количество составляющих путь ребер, равное $n + 1$ называется **длиной** этого пути. В свою очередь, **цикл** – это путь на графе, начальная и конечная вершины которого совпадают. [11]

Определим матрицу смежности $R = [r_{ij}]_n^n$ графа $G = (V, E)$ как квадратную матрицу порядка $n = |V|$, где:

$$r_{ij} = \begin{cases} 1, & \text{если существует дуга } a \text{ из } v_i \text{ в } v_j \\ 0, & \text{если такой дуги не существует} \end{cases}$$

Матрица смежности полностью определяет структуру графа [11]. Для неориентированного графа матрица смежности всегда симметрична.

Матрица инциденции $S = [s_{ij}]_n^m$, описывающая ориентированные графы – это прямоугольная матрица размерности $|V| \times |E|$, в которой

$$s_{ij} = \begin{cases} 1, & \text{если дуга } d_j \text{ исходит из вершины } v_i \\ -1, & \text{если дуга } d_j \text{ заходит в } v_i \\ 0, & \text{если дуга } d_j \text{ не инцидентна } v_i \end{cases}$$

Одним из подвидов орграфов являются знаковые взвешенные орграфов. Помимо орграфа $G(V, E)$ в модель включаются множество параметров вершин $X = \{x_i, i \leq N = |V|\}$ и функционал преобразования дуг $F(X, E)$, ставящих в соответствие каждой дуге либо знак, либо вес, либо функцию. [12]

Если функционал имеет вид:

$$F(x_i, x_j, e_{ij}) = \begin{cases} +1, & \text{когда рост (падение)} x_i \text{ влечет за собой рост (падение)} x_j \\ -1, & \text{когда рост (падение)} x_i \text{ влечет за собой падение (рост)} x_j \end{cases}$$

то такая модель называется **знаковым орграфом**.

Оставаясь знаковым, орграф может быть в то же время взвешенным в том смысле, что его вершинам может быть приписан вес, то есть если функционал имеет вид:

$$F(x_i, x_j, e_{ij}) = \begin{cases} +w_{ij}, & \text{когда рост (падение)} x_i \text{ влечет за собой рост (падение)} x_j \\ -w_{ij}, & \text{когда рост (падение)} x_i \text{ влечет за собой падение (рост)} x_j \end{cases}$$

то такая модель называется **взвешенным знаковым орграфом**. [12]

Данные графы удобно использовать в моделях, в которых делаются предположения о влиянии изменений значения параметра одной вершины на параметры других вершин. На таких графах можно качественно и количественно оценить изменения значений параметров вершин $X(t)$, происходящие в системе в дискретные моменты времени под влиянием поступающего импульса (изменения) $P(t)$. Стоит учитывать, что этот импульс может быть поступать из «внешнего мира» за пределом модели и отражать изменение одного из параметров.

Для изучения работы сложных систем получил широкое распространение метод формализации в виде сетей Петри.

Сети Петри — это метод моделирования динамических дискретных систем, в которых процессы происходят асинхронно параллельно. Сеть Петри представляет собой двудольный ориентированный граф, который состоит из вершин, которые могут являться позициями или переходами (Р и Т), и дуг, которые соединяют их (I направлены от позиций к переходам, О — от переходов к позициям).

В сетях Петри вводятся объекты двух типов: динамические, которые изображаются метками (маркерами), являющиеся объектами (заказы, заявки, ресурсы

и др.) моделируемых процессов внутри позиций, и статические, которым соответствуют вершины сети Петри. Перемещение маркеров отображает динамику процессов. Число маркеров в позиции выражается целым неотрицательным числом и определяет состояние позиции, а набор состояний всех позиций – состояние сети Петри. Движение маркеров происходит в результате выполнения (срабатывания) перехода под воздействием внешних и внутренних условий осуществления процесса. [12]

В данной работе будут использованы подвид орграфов – когнитивные графы (карты). Когнитивные карты – это представление связей, которые существуют между объектами в какой-либо предметной области. Основной особенностью когнитивных графов является то, что они могут качественно описать динамические ситуации. Исследования когнитивных карт и методов их анализа дали теоретическую основу для создания систем, которые могут помочь принимать решения в сложных и неопределенных ситуациях. Качество таких систем зависит не только от удобства и развитости интерфейсов программ, но и от строения самой когнитивной карты — это субъективная модель процессов реальности. Это означает, что при создании когнитивной карты мы опираемся на свои представления о процессах, которые происходят в реальности. Таким образом, чем более точной и полной будет когнитивная карта, тем более обоснованными будут решения, принятые на ее основе. Качественная и гибкая система поддержки принятия решений помогает решать задачи в условиях неопределенности, увеличивая рациональность принимаемых решений. [17]

Отношения в когнитивных графах часто выражаются в численном эквиваленте, и тогда они могут иметь положительное, отрицательное и нейтральное влияние друг на друга. В данной работе будет продемонстрировано, как когнитивные графы используются для прогноза и оценки взаимовлияния вершин друг на друга. [12]

Для определения наличия циклов в орграфе будет использоваться алгоритм Джонсона по поиску циклов в графе. Данный алгоритм позволяет найти кратчайшие пути между всеми парами вершин взвешенного ориентированного графа в течение времени $O(V^2 \log(V) + VE)$, где V -количество вершин, E -количество ребер. Его

особенностью является то, что он работает, если в графе содержатся рёбра с положительным или отрицательным весом, но отсутствуют циклы с отрицательным весом. [13]

В этом алгоритме используется метод изменения веса. Суть его заключается в том, что для заданного графа G строится новая весовая функция ω_φ неотрицательная для всех ребер графа G и сохраняющая кратчайшие пути. Такая весовая функция строится с помощью так называемой потенциальной функции.

Пусть $\varphi: V \rightarrow R$ — произвольное отображение из множества вершин в вещественные числа. Тогда новой весовой функцией будет:

$$\omega_\varphi(u, v) = \omega(u, v) + \varphi(u) - \varphi(v).$$

Такая потенциальная функция строится добавлением фиктивной вершины s в G , из которой проведены ориентированные ребра нулевого веса во все остальные вершины графа, и запуском алгоритма Форда-Беллмана из нее ($\varphi(v)$ будет равно длине кратчайшего пути из s в v). На этом же этапе мы сможем обнаружить наличие отрицательного цикла в графе.

Теперь, когда мы знаем, что веса всех ребер неотрицательны, и кратчайшие пути сохраняются, можно запустить алгоритм Дейкстры из каждой вершины и таким образом найти кратчайшие расстояния между всеми парами вершин. [14]

В данной работе предлагается рассмотреть метод описания функционирования социально-экономической системы с помощью теории знаковых и взвешенных орграфов на основе свойства их **сбалансированности**.

Цикл называется сбалансированным, если произведение весов на ребрах цикла равно положительно, и не сбалансированным в противном случае. [12] Знаковый граф называется сбалансированным, если каждый его простой цикл положительный, где знак цикла определяется как произведение знаков входящих в него ребер. [15]

Необходимое и достаточное условие сбалансированности устанавливает теорема Харари [16]:

«Знаковый граф $\Sigma = (G, \sigma)$ сбалансирован тогда и только тогда, когда множество его вершин V можно разбить на две доли A и B , одна из которых может быть пустой, таким образом, что $A \cup B = V, A \cap B = \emptyset$ и любое ребро,

соединяющее вершины из одной доли, имеет знак « + », а ребра, соединяющие вершины из разных долей – знак « – ».

Опасность несбалансированных циклов заключается в том, что они приводят к дестабилизации всех элементов цикла, влияют на все связи в данном цикле.

Глава 2. Сценарии рискованных событий

Мы рассматриваем потенциально возможные варианты развития событий, которые при изменении внешних факторов могут изменять свою значимость. Нами выбраны сценарии рискованных событий, которые, с одной стороны, демонстрируют необходимость применения принципов, которые диктует доказательная государственная политика (ДГП), а с другой стороны, мы указываем на спектр новых рисков, которые будут ожидать лица, принимающие решения (ЛПР) в случае применения новых принципов в текущей экономической и политической обстановке.

1. Рассмотрим систему, при которой показатели эффективности мер государственной поддержки учитывают интересы только одной группы стейкхолдеров, интересы второй группы же игнорируются.
2. Сравним два подхода к построению мер аудиторской проверки эффективности мер государственной поддержки. Первый подход предполагает для повышения эффективности проверки разрабатывать более полную систему ключевых показателей, покрывающих все экономические, финансовые и технологические процессы на предприятии. Второй подход основывается на положении доказательной государственной политики, определяющей проведение риск-ориентированного метода оценок. Этот метод определяет необходимость проверки потенциально ущербных процедур.
3. Проведем анализ рисков, возникающих при реализации таких подходов доказательной государственной политики как
 - а) использование для проведения анализа внешних негосударственных банков данных
 - б) использование для получения доказательного анализа мнений и методов независимых экспертов.

Модель 1 и Модель 2. Учет рисков несбалансированного учета интересов стейкхолдеров

Одним из положений ДГП становится то, что в рамках проводимых процедур необходимо учитывать интересы не только бенефициаров, государственных

институтов, но и большого числа стейкхолдеров. Построим модель, описывающую развитие отношений между указанными группами в случае, если интересы одной группы стейкхолдеров введением государственных мер поддержки нарушаются, а аудиторская проверка это нарушение не учитывает. На наш взгляд два исхода аудиторской проверки существенно влияют на развитие ситуации, поэтому предусмотрены два варианта: результат проверки критичен в оценке эффективности мер поддержки (Модель 1) и результат проверки лоялен к оценке мер поддержки (Модель 2). Визуализация графов была осуществлена на электронном ресурсе app.diagrams.net (дата обращения 28.11.2022).

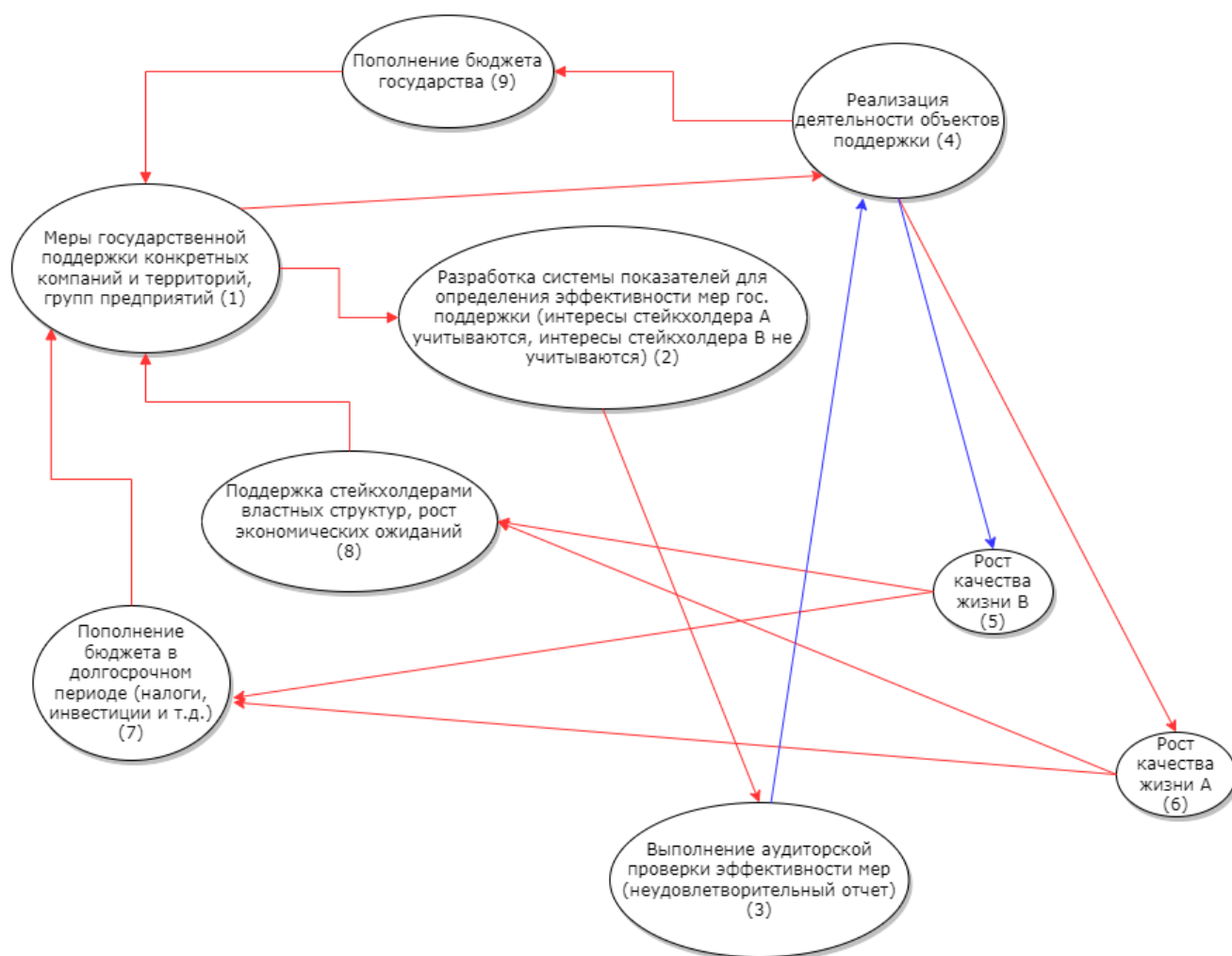


Рисунок 1. Граф Модели 1.

Описываемый ситуацию когнитивный граф состоит из 9 вершин:

- 1) Меры государственной поддержки конкретных компаний и территорий, групп предприятий;

- 2) Разработка системы показателей для определения эффективности мер гос. поддержки (интересы стейкхолдера А учитываются, интересы стейкхолдера В не учитываются);
- 3) Выполнение аудиторской проверки эффективности мер;
 - а. Результат: неудовлетворительный отчет;
 - б. Результат: удовлетворительный отчет;
- 4) Реализация деятельности объектов поддержки;
- 5) Рост качества жизни В;
- 6) Рост качества жизни А;
- 7) Пополнение бюджета в долгосрочном периоде (налоги, инвестиции и т. д.);
- 8) Поддержка стейкхолдерами властных структур, рост экономических ожиданий;
- 9) Пополнения бюджета государства, создание мультиплицирующего эффекта, развивающего смежные отрасли.

Во взвешенный орграф включены наиболее важные связи, определяющие направленные влияния (всего 14 ребер). Поскольку наша задача найти сбалансированные циклы, необходимо объяснить знаки каждого ребра. Опишем причинно-следственные связи (ребра когнитивного графа):

1-4. При увеличении мер государственной поддержки очевидно должна увеличиваться эффективность деятельности субъекта, поэтому вес данного ребра (связи) +1;

4-9. Успешная реализация деятельности поддерживаемого субъекта влечет за собой увеличение выделяемых средств в бюджет государства, поэтому данная связь положительна (+1);

9-1. Чем больше количество денежных средств, которые есть в бюджете, и чем выше эффективность мер поддержки, тем больше мер поддержки оказывается со стороны государства, вес данного ребра +1;

1-2. По мере увеличения мер поддержки необходимо развивать систему показателей для определения эффективности мер, поэтому данное ребро положительное (+1);

2-3. Разработка системы показателей определения эффективности мер позволяет осуществлять аудиторскую проверку, связь данных вершин имеет вес +1;

3-4. Поскольку в данной модели мы рассматриваем случай, когда после аудиторской проверки был получен неудовлетворительный отчет, то не была реализована эффективная деятельности объекта поддержки, поэтому вес данного ребра -1;

4-5. Реализация деятельности объекта поддержки негативно скажется на росте качества жизни стейкхолдеров В, так как их интересы не были учтены при разработке системы показателей определения эффективности мер, поэтому данная связь отрицательна (-1);

4-6. Поскольку интересы стейкхолдеров А были учтены при разработке системы показателей определения эффективности мер, то реализация деятельности объекта поддержки положительно скажется на их качестве жизни, поэтому вес данного ребра +1;

5-7. Стейкхолдеры В являются налогоплательщиками, то есть от их качества жизни напрямую зависит, сколько налогов и дополнительные инвестиции в государство будут готовы отдать стейкхолдеры В, эта связь положительная (+1);

6-7. Аналогично 5-7 со стейкхолдерами А, связь имеет вес +1;

5-8. Если качество жизнь стейкхолдеров В растет, то и поддержка властных структур также будет увеличиваться и наоборот, это ребро имеет положительный вес (+1);

6-8. Аналогично 5-8 со стейкхолдерами А, связь имеет вес +1;

7-1. Пополнение бюджета в долгосрочной перспективе позволяет государству увеличивать количество мер поддержки, поэтому это ребро положительное (+1);

8-1. Аналогично связи 7-1 поддержка власти является мотиватором государства наращивать меры поддержки, эта связь имеет вес +1.

Применив алгоритм Джонсона («Код 01. Построение и определение циклов в графах») к графу Модели 1, получаем следующие циклы:

сбалансированные: [[1, 4, 6, 7, 1], [1, 4, 6, 8, 1], [1, 4, 9, 1],
[1, 2, 3, 4, 5, 7, 1], [1, 2, 3, 4, 5, 8, 1]]

несбалансированные: $[[1, 4, 5, 7, 1], [1, 4, 5, 8, 1], [1, 2, 3, 4, 6, 7, 1],$
 $[1, 2, 3, 4, 6, 8, 1], [1, 2, 3, 4, 9, 1]]$

Опишем основные из несбалансированных циклов:

$1-(+)\rightarrow 2-(+)\rightarrow 3-(-)\rightarrow 4-(+)\rightarrow 9-(+)\rightarrow 1$

Этот цикл несбалансирован, так как неудовлетворительный отчет свидетельствует о неудачной реализации деятельности объекта поддержки, которая в свою очередь может сократить объем выделяемых средств на меры поддержки государства.

$4-(-)\rightarrow 5-(+)\rightarrow 8-(+)\rightarrow 1-(+)\rightarrow 4$

В данном случае не будет расти качество жизни стейкхолдеров В из-за реализации мер поддержки без учета интереса группы стейкхолдеров В, от чего будет снижаться их благосклонность к властным структурам и их экономические ожидания, что повлечет за собой поступление средств в бюджет государства и, как следствие, сокращение мер поддержки на реализацию деятельности объекта поддержки.

$4-(-)\rightarrow 5-(+)\rightarrow 7-(+)\rightarrow 1-(+)\rightarrow 4$

Данный цикл похож на предыдущий, однако от снижения качества жизни стейкхолдеров В будут сокращаться поступления в бюджет на долгосрочной перспективе, что уменьшит количество мер государственной поддержки и сократит количество поддерживаемых объектов.

Циклы с участием вершины 6 ($1-(+)\rightarrow 2-(+)\rightarrow 3-(-)\rightarrow 4-(+)\rightarrow 6-(+)\rightarrow 8-(+)\rightarrow 1$ и $1-(+)\rightarrow 2-(+)\rightarrow 3-(-)\rightarrow 4-(+)\rightarrow 6-(+)\rightarrow 7-(+)\rightarrow 1$) не столь критичны, так как интересы стейкхолдера А были учтены при любом результате аудиторской проверки.

Примитивной рекомендацией можно считать ограничение мер государственной поддержки, так негативное влияние на качество жизни стейкхолдеров В и на реализацию деятельности объекта поддержки будет минимальным.

Рассмотрим теперь ситуацию, когда в результате аудиторской проверки был получен удовлетворительный отчет.

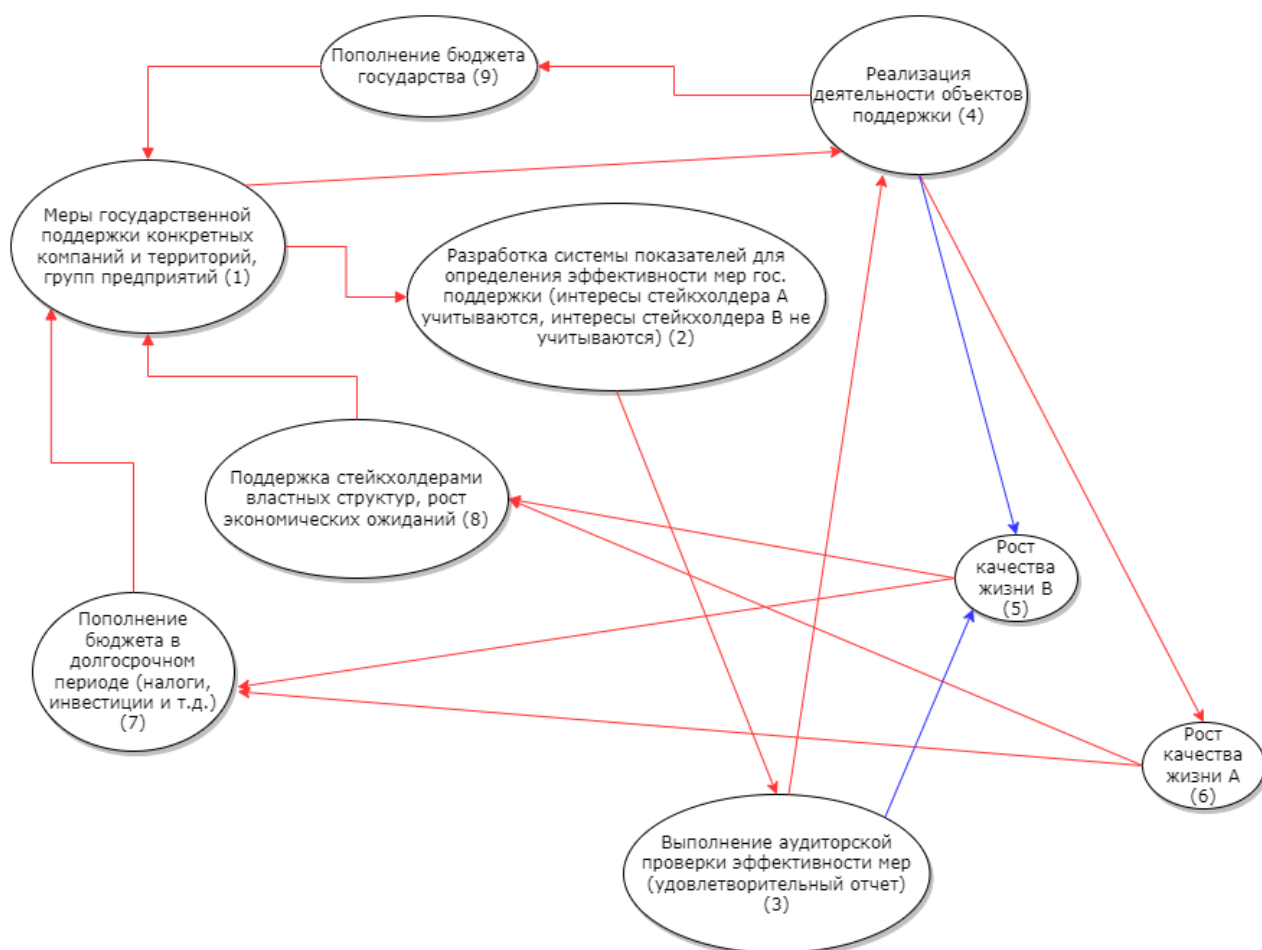


Рисунок 2. Граф Модели 2.

Данный граф отличается от предыдущего следующими связями:

3-4. Теперь это ребро положительно (+1), так как был получен удовлетворительный отчет, что означает успешную реализацию деятельности объекта поддержки.

3-5. Данная связь возникает, так как удовлетворительный отчет негативно скажется на росте качества жизни стейкхолдеров В, ведь их интересы не были учтены и данное положение дел признается государством, поэтому вес ребра -1.

В данной модели наблюдаем следующие циклы:

сбалансированные: [[1, 4, 6, 7, 1], [1, 4, 6, 8, 1], [1, 4, 9, 1], [1, 2, 3, 4, 6, 7, 1],
[1, 2, 3, 4, 6, 8, 1], [1, 2, 3, 4, 9, 1]]

несбалансированные: [[1, 4, 5, 7, 1], [1, 4, 5, 8, 1], [1, 2, 3, 5, 7, 1], [1, 2, 3, 5, 8, 1],
[1, 2, 3, 4, 5, 7, 1], [1, 2, 3, 4, 5, 8, 1]]

1-(+)->2-(+)->3-(-)->5-(+)->8-(+)->1

1-(+)->2-(+)->3-(-)->5-(+)->7-(+)->1

Данные циклы несбалансированны, так как от отрицательного влияния удовлетворительного аудиторского отчета на рост качества жизни стейкхолдеров В будут снижаться и их поддержка властных структур (8) и долгосрочные поступления в бюджет (7), что повлечет за собой сокращение мер государственной поддержки.

1-(+)->4-(-)->5-(+)->8-(+)->1

1-(+)->4-(-)->5-(+)->7-(+)->1

Данные циклы аналогичны соответственным циклам в Модели 1.

Решением проблемы несбалансированных циклов в Модели 2 видится ограничение мер государственной поддержки. Значимость подобной рекомендации ничтожна, следовательно решением данной проблемы является создание процедур ДГП, которые, во-первых, должны определить список стейкхолдеров с определением характера взаимодействия с оцениваемым процессом, во-вторых, разрабатывать процедуры оценки, учитывающие возможные негативные эффекты реализации мер поддержки для определенных стейкхолдеров.

Модель 3 и Модель 4. Обоснование необходимости использования риск-ориентированного подхода при реализации процедур ДГП

Методы ДГП предполагают использование алгоритмов риск-ориентированного подхода. Логичным является то, что для улучшения качества анализа необходимо проверять выполнение качества исполнения каждого из актуальных процессов, что требует создания более широкой системы ключевых показателей функционирования субъекта. Процедура определения такой системы весьма затратна и требует значительного времени. Аналогичное возрастание затрат возникает и при аудите деятельности такой системы, при этом значимость доказательных выводов по отдельным показателям может быть ничтожной.

Для достижения более качественного аудита рекомендуется использовать методы риск-ориентированного подхода. Суть этого подхода в том, что рассмотрению подлежат лишь качество тех процессов, ущерб от нарушения которых существенен или является критическим для всего субъекта. В этом случае процесс, когда

организация аудита несет издержки, смещается в процесс определения риска. Рассмотрим две схемы организации проверки: Модель 3 - результат расширения количества ключевых показателей для аудита качества процессов, Модель 4 – результат применения риск-ориентированного подхода.

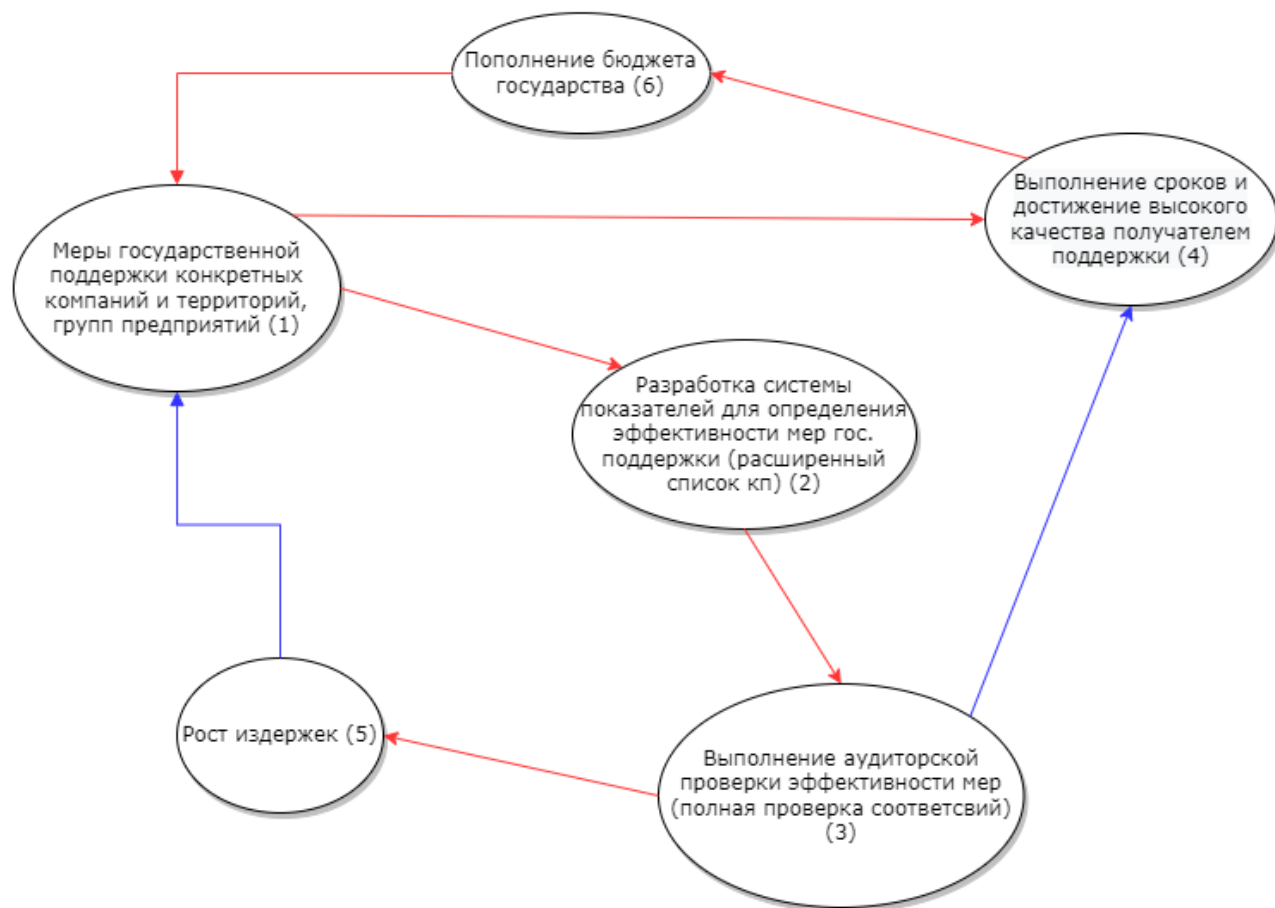


Рисунок 3. Граф Модели 3.

В модели 3 был сформирован орграф, содержащий следующие вершины:

- 1) Меры государственной поддержки конкретных компаний и территорий, групп предприятий;
- 2) Разработка системы показателей для определения эффективности мер гос. поддержки (расширенный список кп);
- 3) Выполнение аудиторской проверки эффективности мер (полная проверка соответствий);
- 4) Выполнение сроков и достижение высокого качества;
- 5) Рост издержек;

6) Пополнения бюджета государства, создание мультиплицирующего эффекта, развивающего смежные отрасли.

Вершины «Поддержка стейкхолдерами властных структур, рост экономических ожиданий» и «Пополнение бюджета в долгосрочной перспективе» были упразднены. Также вершина (4) теперь учитывает факторы времени и качества выполненных мер поддержки.

Опишем новые связи в данном графе.

1-4. При увеличении мер государственной поддержки увеличивается качество работы объекта поддержки, поэтому вес данного ребра (связи) +1;

4-6. Своевременное выполнение работы при достижении высокого качества влечет за собой увеличение отчисляемых средств в бюджет государства, поэтому данная связь положительна (+1);

6-1. Чем больше эффект мер поддержки (величина отчислений в бюджет), тем более интенсивней можно реализовать меры поддержки со стороны государства, вес данного ребра +1;

1-2. По мере увеличения мер поддержки будет вынужденно разрабатываться система показателей для определения эффективности мер, поэтому данное ребро положительное (+1);

2-3. Разработка системы показателей определения эффективности мер позволяет реализовывать аудиторскую проверку, в рамках которой будет проверяться полный список соответствий деятельности установленным ключевым показателям, связь имеет вес +1;

3-4. Проверка полного соответствия аудиторами крайне трудозатратна, что негативно влияет на своевременное выполнение сроков мер поддержки и отражается на качестве реализации деятельности объектов поддержки, поэтому данное ребро имеет вес -1;

3-5. Полная аудиторская проверка при расширенном списке ключевых показателей помимо времени требует дополнительных издержек, поэтому данная связь положительна (+1);

5-1. Рост издержек негативно влияет на увеличение количества мер

государственной поддержки, так как влечет за собой дополнительные расходы бюджета, эта связь имеет вес -1.

После применения алгоритма Джонсона были выявлены следующие циклы:
сбалансированные: [[1, 4, 6, 1]]

несбалансированные: [[1, 2, 3, 5, 1], [1, 2, 3, 4, 6, 1]]

Далее рассмотрим возникшие несбалансированные циклы:

1-(+)->2-(+)->3-(+)->5-(-)->1

Возникновение отрицательного ребра 5-1 может в будущем негативно сказаться на все вершины данного цикла, так как, во-первых, мы наблюдаем рост издержек на проведение процедур аудита, во-вторых, производство большого количества ничтожных для анализа процедур снижает эффективность выводов и качества аудита. При этом негативные эффекты вряд ли могут быть признаны критическими для динамики системы.

1-(+)->2-(+)->3-(-)->4-(+)->6-(+)->1

В данном цикле ребро 3-4 не позволяет выполнять проверку в срок, что негативно сказывается на реализации деятельности поддерживаемого объекта, ведь она может оказаться крайне неэффективной. Это, в свою очередь, влечет уменьшение выделяемых денежных средств на поддержку субъектов и сокращает количество таких инициатив со стороны государства. Таким образом существенно возрастают риски несвоевременного исполнения обязательств субъекта, что является критичным. Управление этим риском в данной ситуации является первоочередной задачей.

В Модели 4 рассмотрим использование положений риск-ориентированного подхода.

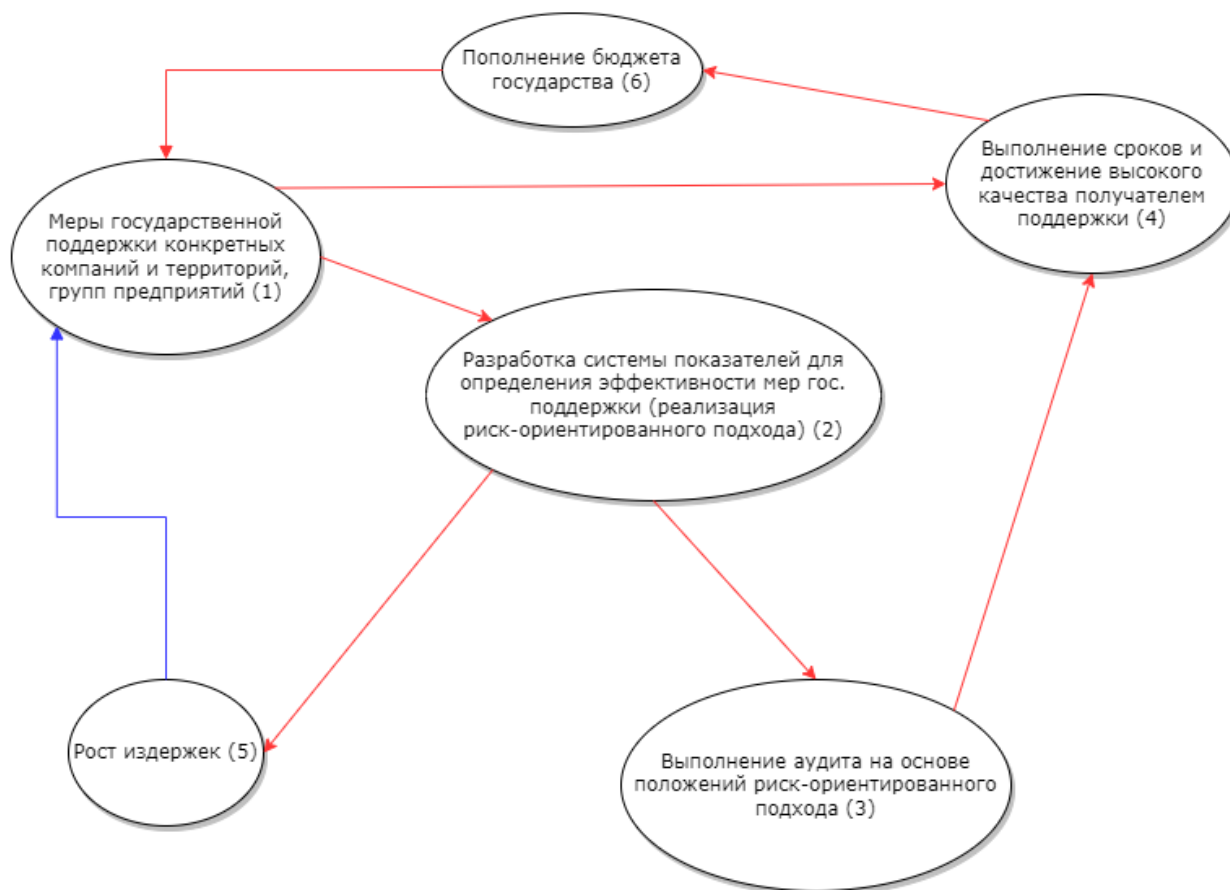


Рисунок 4. Граф Модели 4.

Опишем изменения в связях. Во-первых, мы изъяли ребро 3-5, построили связь 2-5, так как теперь издержки растут на этапе разработки системы показателей с реализацией риск-ориентированного подхода, поэтому ребро 2-5 имеет вес +1. Ребро 3-4 теперь имеет вес +1, так как риск-ориентированный подход позволил убрать излишнюю трудозатратность аудиторской проверки.

Аналогично прошлым моделям, проверим наличие циклов в данном графе:

сбалансированные: $[[1, 4, 6, 1], [1, 2, 3, 4, 6, 1]]$

несбалансированные: $[[1, 2, 5, 1]]$

В данном графе возникает только один несбалансированный цикл:

$1-(+)\rightarrow 2-(+)\rightarrow 5-(-)\rightarrow 1$.

Действительно, определение риска в работе субъекта высокотехнологическая процедура, требующая привлечение экспертов. (Методы ДГП будут требовать привлечения экспертов даже в случае присутствия надлежащей системы риск-

ориентированной системы на предприятии).

В случае применения такого подхода мы избавляемся от несбалансированного цикла, проходящего через вершину 4, что существенно снижает риски решения основной задачи. Последнее указывает на актуальность применения элементов ДГП при проведении аудита эффективности мер государственной поддержки.

Модель 5. Идеальная модель использования разных обоснований в ДГП

Основные обоснования, применяемые в качестве доказательств, приводимых в рамках внедрения идеологии ДГП можно разделить на три группы:

- а) объективные научные данные;
- б) результаты изучения общественного мнения (коллективный интеллект);
- в) экспертные данные (экспертное мнение).

Под объективными научными данными мы подразумеваем классические методы сбора и анализа данных для проведения оценок в рамках аудиторских проверок, результатами изучения общественного мнения можно признать обобщение практик функционирования производственных и общественных субъектов, которые можно интерпретировать как банки больших данных, собираемых в первую очередь негосударственными институтами, реализующих их в целях оперативного принятия решений бизнес структурами. Как показывает практика, государственные системы сбора обобщенной информации (в Российской Федерации) не обладают надлежащей гибкостью и оперативностью. Под экспертными данными можно полагать мнения независимых экспертов, которые не аффилированы с государственными институтами, но признаны специалистами в своей профессиональной среде. Эти мнения в нашем случае предполагаются разработанными альтернативными методиками анализа, имеющими научное обоснование.

В модели 5 изложим идеальные связи, возникающие при реализации дополнительных ((б) и (в)) способов доказательств.

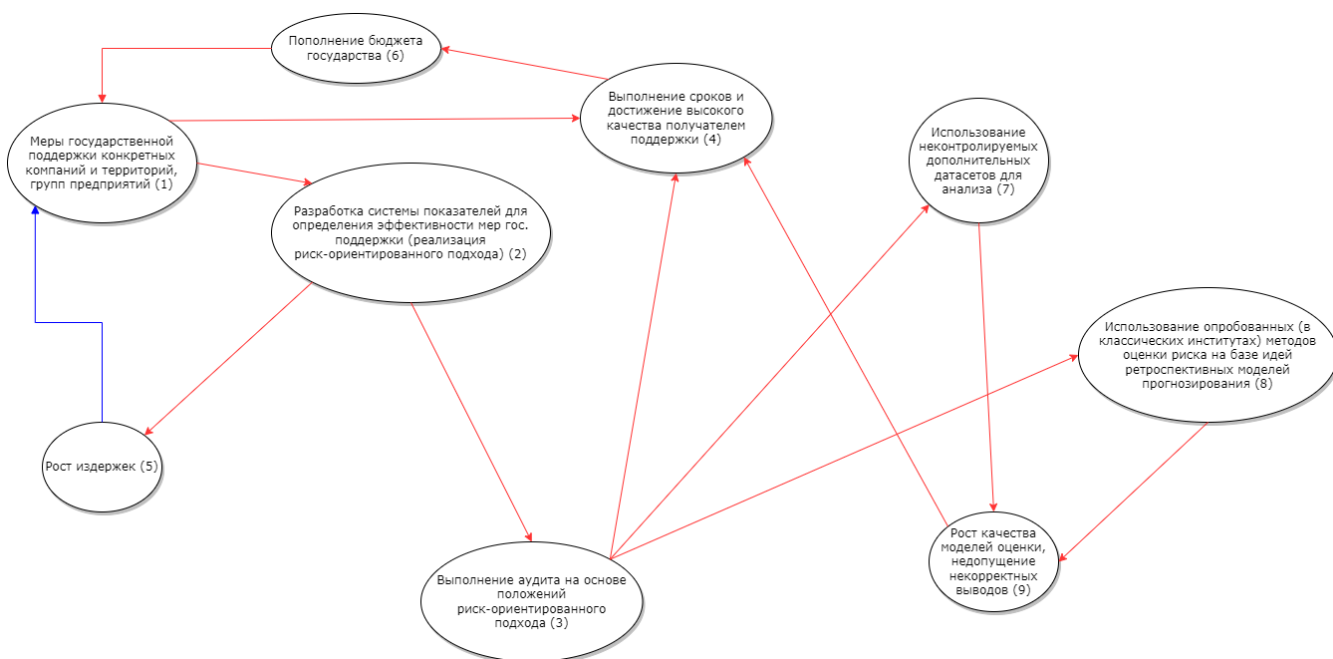


Рисунок 5. Граф Модели 5.

Данный граф содержит следующие вершины:

- 1) Меры государственной поддержки конкретных компаний и территорий, групп предприятий;
- 2) Разработка системы показателей для определения эффективности мер государственной поддержки (реализация риск-ориентированного подхода);
- 3) Выполнение аудита на основе положений риск-ориентированного подхода;
- 4) Выполнение сроков и достижение высокого качества;
- 5) Рост издержек;
- 6) Пополнения бюджета государства, создание мультиплицирующего эффекта, развивающего смежные отрасли;
- 7) Использование неконтролируемых государством дополнительных датасетов для анализа;
- 8) Использование опробованных (в классических институтах) методов оценки риска на базе идей ретроспективных моделей прогнозирования, предложенных внешними экспертами;
- 9) Рост качества моделей оценки, недопущение некорректных выводов.

Опишем причинно-следственные связи в данной системе:

1-4. При увеличении мер государственной поддержки увеличивается качество работы объекта поддержки, поэтому вес данного ребра (связи) +1;

4-6. Своевременное выполнение работы при достижении высокого качества влечет за собой увеличение выделяемых средств государством в виде налоговых отчислений, реализации мультиплицирующего эффекта в смежных отраслях, поэтому данная связь положительна (+1);

6-1. Чем больше эффект мер поддержки (величина отчислений в бюджет), тем более интенсивней можно реализовать меры поддержки со стороны государства, вес данного ребра +1;

1-2. По мере увеличения мер поддержки будет вынужденно разрабатываться система показателей для определения эффективности мер, поэтому данное ребро положительное (+1);

2-3. Разработка системы показателей определения эффективности мер влечет за собой аудиторскую проверку, связь данных вершин имеет вес +1;

3-4. Риск-ориентированный подход позволяет убрать излишнюю трудозатратность аудиторской проверки и достичь высокого качества реализации деятельности объекта поддержки, поэтому ребро имеет вес +1;

2-5. Издержки растут на этапе разработки системы показателей с реализацией риск-ориентированного подхода, поэтому данная связь положительна (+1);

5-1. Рост издержек негативно влияет на увеличение количества мер государственной поддержки, так как влечет за собой дополнительный расход бюджета, эта связь имеет вес -1;

3-7. При выполнении аудиторской проверки в соответствии с идеями ДГП должны будут использоваться неконтролируемые государству дополнительные датасеты для анализа, это пополняет объем информации для принятия решений, поэтому данная связь положительна (+1);

3-8. Использование мнений внешних экспертов является элементом внедрения идей ДГП, ребро имеет вес +1;

7-9. Использование независимых датасетов повлечет за собой рост качества моделей оценки и недопущение ошибок, поэтому данная связь положительна (+1);

8-9. Аналогично ребру 7-9 внедрение опробованных методов оценки риска на базе идей ретроспективных моделей прогнозирования оказывает положительное влияние на качество моделей оценки, данная связь имеет вес +1;

9-4. Рост качества моделей оценки и недопущение некорректных выводов напрямую влияет на выполнение мер государственной поддержки в срок при достижении высокого качества, поэтому данное ребро положительно (+1);

В данном графе возникает всего один несбалансированный цикл, описанный в Модели 4, однако, как и ранее, риски дополнительных издержек компенсированы управляемостью более значимых процессов.

Модель 5 представляет собой идеальный, но, к сожалению, недостижимый вариант использования ДГП. Ввиду современных геополитических и экономических изменений она не может быть реализована без дополнительных вершин и связей

Чтобы описать реальную текущую ситуацию была разработана Модель 6.

Модель 6. Реальная модель использования разных обоснований в ДГП

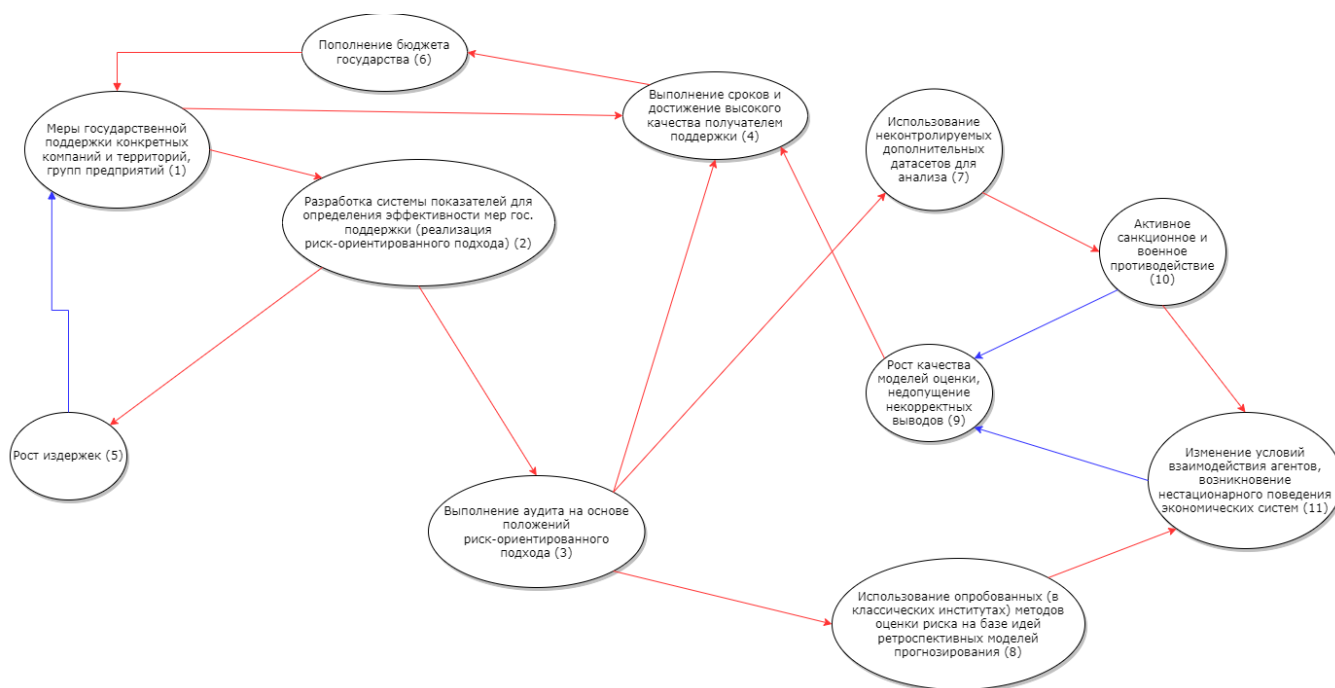


Рисунок 6. Граф Модели 6.

В данном графе появились вершины:

- 10) Активное санкционное и военное противодействие;

11) Изменение условий взаимодействия агентов, возникновение нестационарного поведения экономических систем.

В ребрах графа произошли следующие изменения:

8-11. Ребро 8-9 модели 5 изменяется на 8-11, так как использование опробованных методов оценки риска начинают работать в условиях нестационарного поведения экономических систем, поэтому связь положительна (+1);

7-10. Связь 7-9 модели 5 преобразуется в ребро 7-10, так как использование неконтролируемых датасетов происходит в условиях активного санкционного и военного противодействия, поэтому это ребро имеет вес (+1);

10-11. Данное ребро объясняется тем, что активное санкционное и военное противодействие порождает изменение условий взаимодействия агентов, связь положительна (+1);

10-9. Поскольку при активном санкционном и военном противодействии иностранным агентам выгодно ухудшать качество моделей оценки, возникает эта связь с весом -1;

11-9. При возникновении нестационарного поведения экономических систем ухудшается качество моделей оценки эффективности мер, так как опробованные методы оценки риска в таких условиях не работают, данное ребро отрицательно (-1).

Применим алгоритм Джонсона, проверим наличие циклов в данной системе:

сбалансированные: $[[1, 4, 6, 1], [1, 2, 3, 4, 6, 1]]$

несбалансированные: $[[1, 2, 5, 1], [1, 2, 3, 8, 11, 9, 4, 6, 1], [1, 2, 3, 7, 10, 9, 4, 6, 1], [1, 2, 3, 7, 10, 11, 9, 4, 6, 1]]$

В данной системе наблюдаются следующие несбалансированные циклы:

1-(+)->2-(+)->5-(-)->1 описан в Модели 4.

1-(+)->2-(+)->3-(+)->8-(+)->11-(-)->9-(+)->4-(+)->6-(+)->1

Причиной несбалансированности этого цикла является кризисное поведение экономических показателей, которое искажает характер связей между факторами системы и изменяет силу и направления воздействий, что не позволяет корректно применять стационарные рекуррентные модели прогнозирования.

1-(+)->2-(+)->3-(+)->7-(+)->10-(-)->9-(+)->4-(+)->6-(+)->1

Несбалансированность этого цикла определена не только санкционными ограничениями по отношению к агентам Российской Федерации, но и диверсиями в информационной сфере. Примером таких диверсий можно признать появление после 24 февраля 2022 закладок в свободном софте, используемом разработчиками ПО, которые идентифицируя агента как находящегося на территории РФ или Республики Беларусь вызвали некорректное поведение ПО этого агента (сбой 14 марта 2022 в интернет магазине Wildberries).

1-(+)->2-(+)->3-(+)->7-(+)->10-(+)->11-(-)->9-(+)->4-(+)->6-(+)->1

Цикл имеет ту же природу что и

1-(+)->2-(+)->3-(+)->7-(+)->10-(-)->9-(+)->4-(+)->6-(+)->1, но причиной нестабильности выступает злонамеренные действия оппонентов для противодействия РФ.

Таким образом достигается цель: при неэффективности мер поддержки будут сокращаться объемы выделяемых средств государства на поддержку предприятий и, соответственно, будет сокращаться количество мер поддержки.

Для избежания этих несбалансированных циклов (управления рисками) была предложена Модель 7.

Модель 7. Управление рисками в модели использования разных обоснований в ДГП

В качестве мер управления рисками системы мы предлагаем рассмотреть следующие мероприятия:

- Использование проверенных поставщиков информации (новая вершина 13);
- Образование и поддержка отечественных банков данных, отечественных средств программного обеспечения, российских операционных систем (новая вершина 12);
- Разработка алгоритмов прогнозирования и моделирования на основе новых методов, актуальных в новых условиях высокой волатильности (новая вершина 14).

Использование проверенных поставщиков информации в нашем случае мера временная, хотя очевидно, требующая временных и финансовых издержек на передачу аккредитации таких центров профессиональным государственным правоохранительным подразделениям. Однако, на начальном этапе проблема несбалансированного цикла $1 \rightarrow 2 \rightarrow 3 \rightarrow 7 \rightarrow 10 \rightarrow 9 \rightarrow 4 \rightarrow 6 \rightarrow 1$ будет частично решена.

Полное решение таких проблем реализует образование и поддержка отечественных банков данных, отечественных средств программного обеспечения, российских операционных систем. Данная задача видится долгосрочной и основные бенефициары таких элементов само государство, предприятия ВПК и крупные концерны с государственным участием. Алгоритмы ДГП весьма удачно вписываются в использование таких элементов, которые решают проблему успешного применения доказательной политики. Издержки при образовании этих объектов велики, но без их возникновения методы доказательства ДКП окажутся неполными.

Под разработкой алгоритмов прогнозирования и моделирования на основе новых методов, актуальных в новых условиях высокой волатильности мы подразумеваем использование новых математических методов анализа данных (прежде всего машинное обучение, имитационное моделирование), которые, к нашему сожалению, на сегодняшний день часто используют западные датасеты, программные средства. Таким образом, необходимо подчеркнуть важность поддержки отечественных банков данных, отечественных средств программного обеспечения, российских операционных систем.

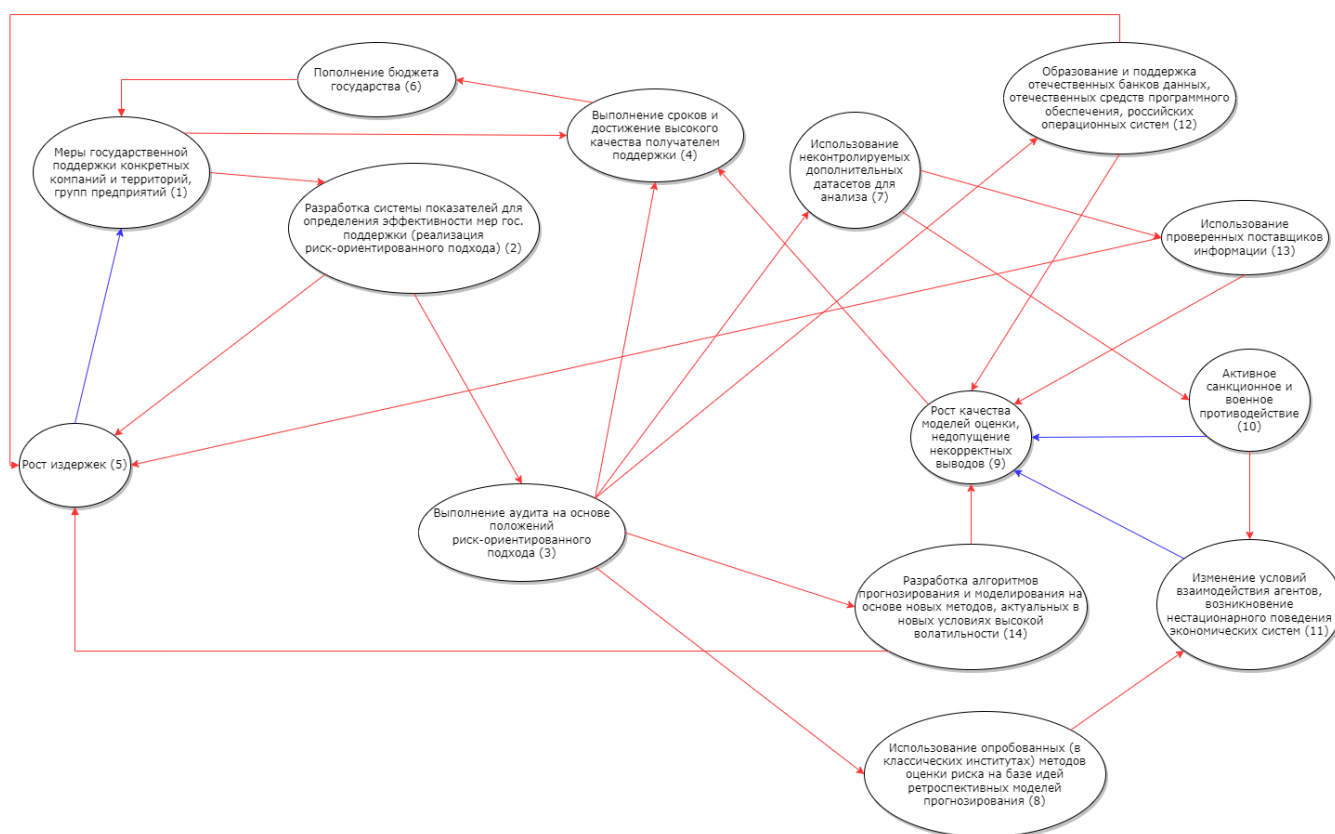


Рисунок 7. Граф Модели 7.

Таким образом, граф модели содержит следующие вершины:

- 1) Меры государственной поддержки конкретных компаний и территорий, групп предприятий;
- 2) Разработка системы показателей для определения эффективности мер гос. поддержки (реализация риск-ориентированного подхода);
- 3) Выполнение аудита на основе положений риск-ориентированного подхода;
- 4) Выполнение сроков и достижение высокого качества;
- 5) Рост издержек;
- 6) Пополнения бюджета государства, создание мультиплицирующего эффекта, развивающего смежные отрасли;
- 7) Использование неконтролируемых дополнительных датасетов для анализа;
- 8) Использование опробованных (в классических институтах) методов оценки риска на базе идей ретроспективных моделей прогнозирования;
- 9) Рост качества моделей оценки, недопущение некорректных выводов;

- 10) Активное санкционное и военное противодействие;
- 11) Изменение условий взаимодействия агентов, возникновение нестационарного поведения экономических систем;
- 12) Образование и поддержка отечественных банков данных, отечественных средств программного обеспечения, российских операционных систем;
- 13) Использование проверенных поставщиков информации;
- 14) Разработка алгоритмов прогнозирования и моделирования на основе новых методов, актуальных в новых условиях высокой волатильности.

Вершина (14) введена для возможности обхода несбалансированного цикла, проходящего через вершину (11). Именно разработка алгоритмов прогнозирования и моделирования на основе методов, актуальных в условиях высокой волатильности позволяет увеличивать качество моделей оценки при возникновении нестационарного поведения экономических систем. Аналогично вершина (13) позволяет избежать несбалансированного цикла, содержащего вершины (10). Вершина (12) добавлена для того, чтобы продемонстрировать инструменты, на которых должна проводиться аудиторская проверка в условиях активного санкционного и военного противодействия.

Раскроем суть новых связей в данной системе.

7-13. Использование неконтролируемых государством дополнительных датасетов в условиях противодействия будет инициировать использование проверенных поставщиков информации, поэтому данная связь имеет вес +1;

13-9. Использование проверенных поставщиков информации позволяет качеству модели оценки расти и предотвращает некорректные выводы, данное ребро положительно (+1);

13-5. Поиск и аккредитация проверенных поставщиков информации, и дальнейшее сотрудничество с ними требует бюджетных затрат, поэтому связь имеет вес (+1);

3-12. Деятельность во враждебной обстановке вынуждено требует использование отечественных банков данных и отечественного ПО при организации и проведении

аудиторской проверки, отсюда возникает положительная связь (+1);

12-9. Образование и поддержка отечественных банков данных, ПО и ОС поспособствует росту качества моделей оценки и недопущение некорректных выводов, данная связь имеет вес +1;

12-5. Образование и поддержка отечественных банков данных, программного обеспечения и операционной системы ресурсозатратно, что влечет за собой рост издержек, поэтому это ребро имеет вес -1;

3-14. При выполнении аудиторской проверки будут использоваться алгоритмы прогнозирования и моделирования на основе новых методов, актуальных в условиях высокой волатильности, поэтому возникает положительная связь (+1);

14-9. Разработка алгоритмов прогнозирования и моделирования на основе методов, актуальных в условиях высокой волатильности позволят вырасти качеству моделей оценки и недопущение некорректных выводов, данное ребро имеет вес +1;

14-5. Разработка алгоритмов прогнозирования и моделирования на основе методов, актуальных в условиях высокой волатильности требует дополнительных издержек, поэтому возникает данная положительная связь (+1).

Аналогично предыдущим моделям найдем циклы в данном графе:

сбалансированные: [[1, 4, 6, 1], [1, 2, 3, 14, 9, 4, 6, 1], [1, 2, 3, 12, 9, 4, 6, 1],

[1, 2, 3, 7, 13, 9, 4, 6, 1], [1, 2, 3, 4, 6, 1]]

несбалансированные: [[1, 2, 5, 1], [1, 2, 3, 8, 11, 9, 4, 6, 1], [1, 2, 3, 14, 5, 1],

[1, 2, 3, 12, 5, 1], [1, 2, 3, 7, 10, 11, 9, 4, 6, 1],

[1, 2, 3, 7, 10, 9, 4, 6, 1], [1, 2, 3, 7, 13, 5, 1]]

В данном графе наблюдаются следующие основные несбалансированные циклы:

1-(+)->2-(+)->3-(+)->8-(+)->11-(-)->9-(+)->4-(+)->6-(+)->1

1-(+)->2-(+)->3-(+)->7-(+)->10-(-)->9-(+)->4-(+)->6-(+)->1

1-(+)->2-(+)->3-(+)->7-(+)->10-(+)->11-(-)->9-(+)->4-(+)->6-(+)->1

Однако значимость этих осциллирующих циклов управляема представленными предложениями, позволяющими применять основные идеи ДГП в современных условиях.

Глава 3. Практическая реализация алгоритма предсказания социально-экономического процесса

3.1. Рекуррентные нейронные сети для предсказания временных рядов

Рекуррентные нейронные сети (RNN) — это вид нейронных сетей, который способен обрабатывать последовательности данных разной длины и сохранять информацию о предыдущих состояниях. Они достигают этого путем добавления в сеть обратной связи на каждом временном шаге, позволяя ей использовать информацию о предыдущих входах для принятия более точных предсказаний.

Рекуррентные нейронные сети имеют широкий спектр применений в различных областях, например, RNN используются в обработке временных рядов в финансовой сфере, где они могут быть использованы для прогнозирования цен на акции, изменений валютных курсов и других финансовых показателей. Они также могут быть использованы в медицине для анализа медицинских данных, таких как ЭКГ и других временных серийных данных. Они могут использоваться для прогнозирования временных рядов, обработки изображений и генерации текстов.

Одним из наиболее распространенных применений рекуррентных нейронных сетей является обработка естественного языка (NLP), где последовательность представляет собой набор слов или символов в тексте. Таким образом, они могут использоваться для таких задач как машинный перевод, распознавание речи и генерация текста.

В последние годы было изобретено много различных архитектур рекуррентных нейронных сетей, таких как LSTM (long-short term memory) и GRU (gates recurrent units). Эти архитектуры были разработаны для решения проблем, связанных с затуханием или взрывом градиентов, которые могут возникнуть при обучении RNN.

На практике для предсказания будущих значений временных рядов эффективнее всего работает архитектура LSTM. Ее особенность заключается в том, что она не использует функцию активации внутри своих рекуррентных компонентов. Таким образом, предотвращается затухание градиента при использовании метода обратного распространения ошибки во времени и хранимые значения в долгосрочной памяти не

размываются [1]. Далее приведу схему работы LSTM модуля.

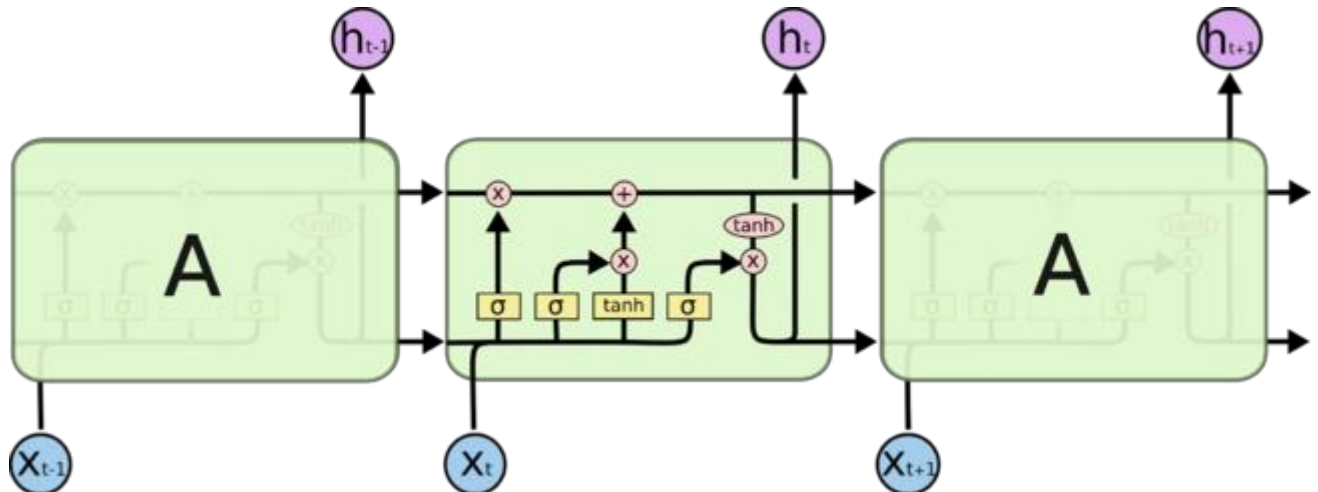


Рисунок 8. Схема работы LSTM модуля (источник: <https://neerc.ifmo.ru/wiki>).

Для прогнозирования социально-экономического процесса необходимо работать с временными рядами, поэтому архитектура LSTM является наиболее релевантной при разработке алгоритма прогнозирования значений.

Существует несколько методов обработки временных рядов для их подготовки к обучению моделей рекуррентных нейронных сетей:

1. метод скользящего окна;
2. заполнение пропущенных значений;
3. удаление сезонности;
4. нормализация;
5. стабилизация дисперсии (например, логарифмирование или метод кокса-бокса);
6. переход к разностям (приведение временного ряда к стационарному виду) [20].

В данном исследовании будут применены метод скользящего окна, как основной метод подготовки датасета для обучения модели, переход к разностям, как попытка обучения модели на стационарном ряде. Удаление сезонности приведет к искажению функционирования социально-экономической системы, изображенной на когнитивном графе, а заполнение пропущенных значений в данном случае не требует ввиду их отсутствия.

Рассмотри также различные стратегии предсказания будущих значений при

использовании рекуррентных нейронных сетей:

1. рекурсивный (recursive);
2. прямой (direct);
3. несколько значений на вход, несколько значений на выходе (multi-input multi-output);
4. прямой с несколькими значениями на выходе (direct multi-output);
5. прямо-рекурсивная (dirrec strategy) [21].

Первый подход предполагает обучение одной модели, что не требует больших затрат времени и памяти. Данная стратегия является одной из самых распространённых, так как проста для понимания и обучения. Однако, при накоплении ошибки, предсказанные значения будут некорректны, так что данный подход подходит только для кратковременных прогнозов.

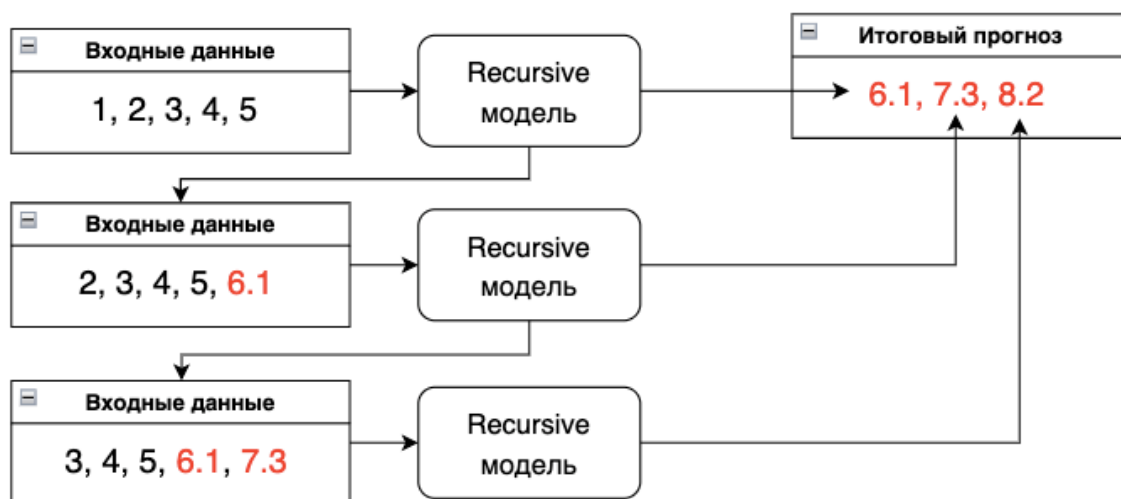


Рисунок 9. Схема работы рекурсивного подхода предсказания значений временного ряда (источник: <https://habr.com/ru/articles/701798/>).

Прямой подход предполагает, что каждое новое значение будет предсказывать отдельно обученная модель (их количество равно числу периодов, которые предсказывают), что также требует затрат по памяти, однако обучение может проводиться параллельно, что сокращает затраты по времени, но такая возможность недоступна в некоторых языках программирования.

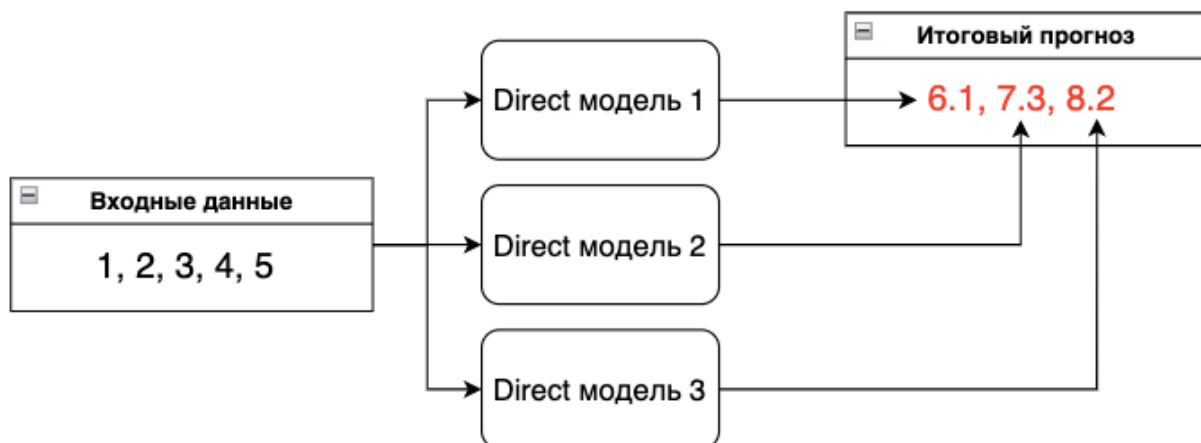


Рисунок 10. Схема работы прямой стратегии предсказания значений временного ряда (источник: <https://habr.com/ru/articles/701798/>).

Стратегия Multi-Input Multi-Output позволяет избежать отсутствие условной независимости прогнозных значений, которая может наблюдаться в прямом и рекурсивном подходе, за счет того, что модель сразу имеет нужное количество значений на выходе. Однако, горизонт прогнозирования в данном случае фиксирован, его можно изменить только заново обучив модель.

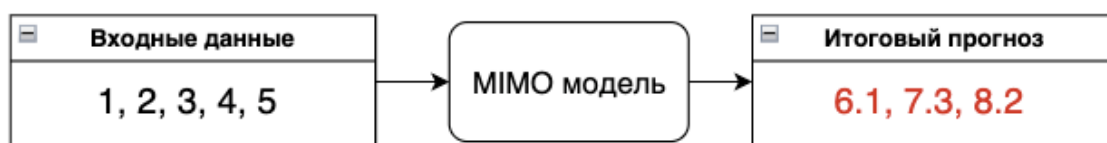


Рисунок 11. Схема работы MIMO стратегии предсказания значений временного ряда (источник: <https://habr.com/ru/articles/701798/>).

Direct Multi-Output стратегия является улучшенной версией второго и третьего подхода. Каждая модель предсказывает несколько значений, что сокращает количество обучаемых моделей.

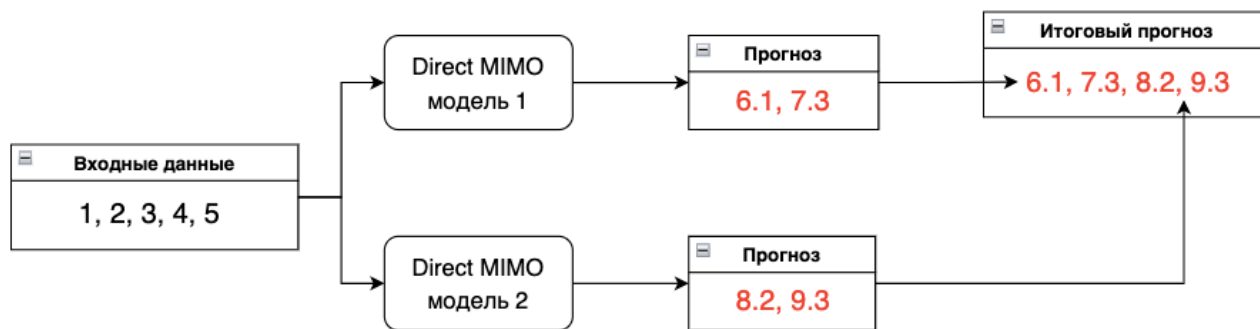


Рисунок 12. Схема работы Direct MIMO стратегии предсказания значений временного ряда (источник: <https://habr.com/ru/articles/701798/>).

Последняя стратегия является смесью первого и второго подхода, что улучшает качество предсказаний, но не решает проблему затрат по памяти и времени.

В своем исследовании я применю 1 и 3 подходы и сравню их эффективность.

3.2. Разработка алгоритма прогнозирования социально-экономического процесса на основе когнитивного графа

Разработка алгоритма прогнозирования социально-экономического процесса проводилась на языке программирования Python при помощи библиотек PyTorch, NetworkX, NumPy. Визуализация результатов осуществлялась при помощи библиотеки Matplotlib.

Поскольку модели социально-экономических систем в виде когнитивных графов на данный момент не имеют широкого распространения, необходимо создать пример такого графа и сгенерировать данные для обучения рекуррентной нейронной сети. За образец был взят граф Модели 7. Каждой вершине будет присвоен массив чисел – массив «бюджетов», которыми располагали вершины в t момент времени. В свою очередь, вес каждого ребра – это множитель, на который умножается актуальное (последнее добавленное) значение (признак) вершины, из которого выходит ребро, и таким образом получается новое значение вершины, в которое приходит ребро. Положительным ребрам присваивались веса из равномерного распределения в интервале от 1 до 1,5, отрицательным – в интервале от 0 до 1. Первым значением, которое содержит вершина под номером 1 было выбрано 10. Также для достоверной

имитации реальных процессов при формировании «бюджетов» не только происходило умножение весов ребер на значения вершин, но и добавлялся белый шум, то есть случайное число из нормального распределения с математическим ожиданием равному 0 и дисперсией равной 1.

Для достаточного количества данных для обучения модели было симулировано 20 кратное движение по графу.

Все «бюджеты» были сохранены в массив, который представляет собой мультипликативный временной ряд, в котором явно просматривается цикличность (см. Рисунок 13).

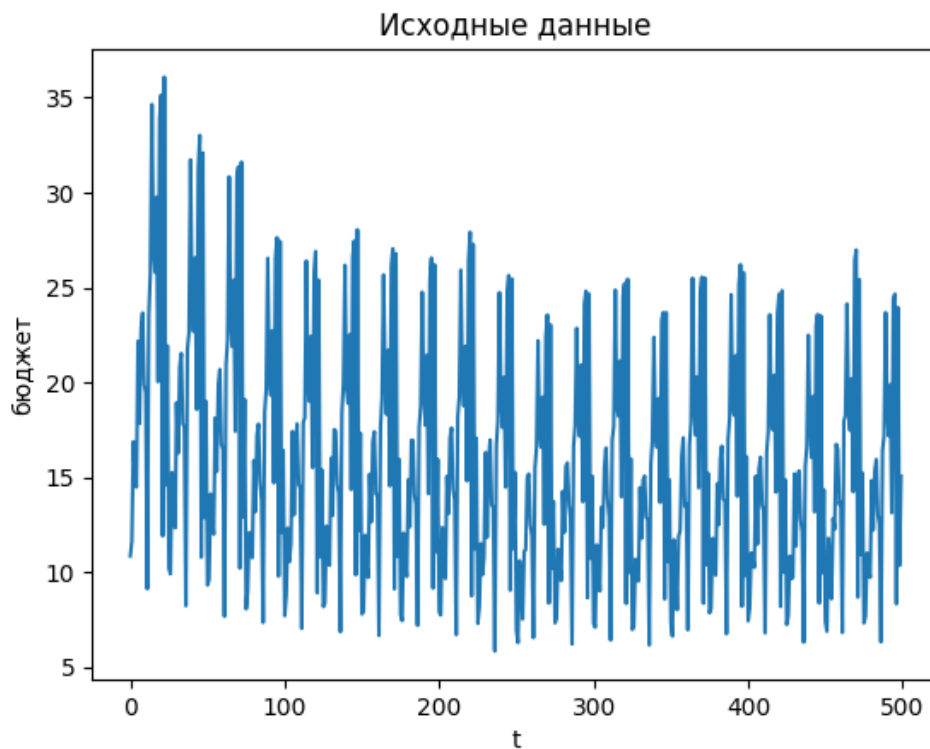


Рисунок 13. Временной ряд признаков вершин.

Далее были сформированы обучающая и тестовая выборки в соотношении 80/20. Размер окна (сколько значений из прошлого подавать на вход) был взят 25, так как предсказание на один проход по графу вперед требует 25 значений.

Для моделирования первого подхода прогнозирования (recursive) на вход в модель подаются массивы размерностью 25 на 1, на выходе массивы размерностью 1, то есть предсказывается значение на 1 шаг вперед.

Модель состоит из двух частей: LSTM модуля с двумя слоями и простого

линейного слоя. Скрытый слой состоит из 20 нейронов. Функцией ошибки, которую оптимизирует алгоритм Adam со скоростью обучения (learning rate) = 0.001, была выбрана среднеквадратическая ошибка (MSE) [22]:

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

После 1000 эпох модель показала стабильный темп обучения и отсутствие переобучения, однако случались всплески ошибки. (см. Рисунок 14)

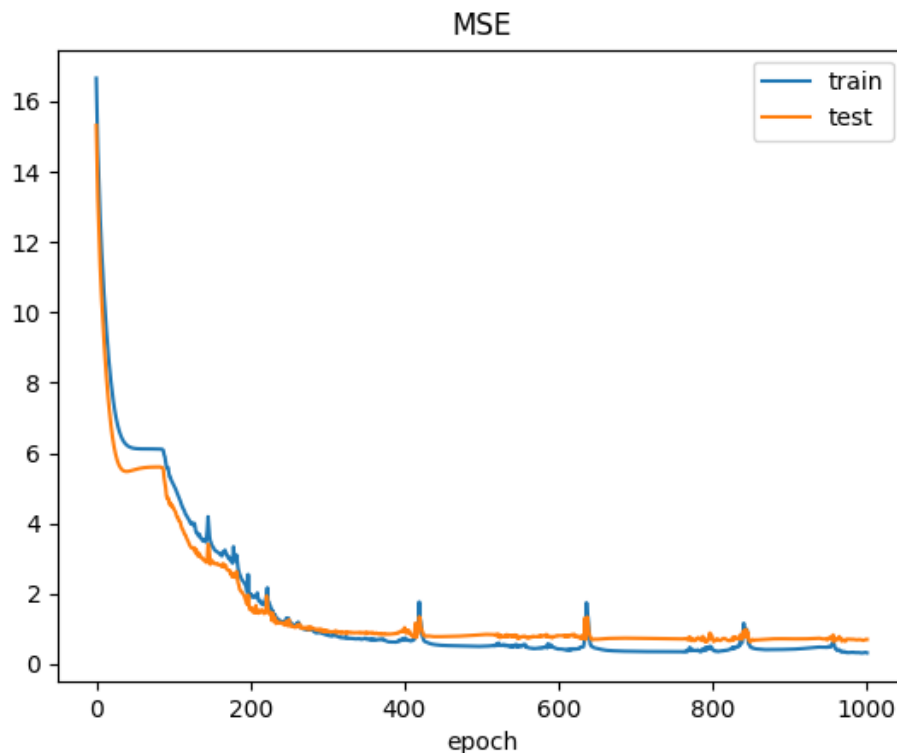


Рисунок 14. График значений функции ошибки в течение обучения модели рекурсивной стратегии.

Предсказанные значения и на тренировочном, и на тестовом множестве близки к реальным значениям (см. Рисунок 15).

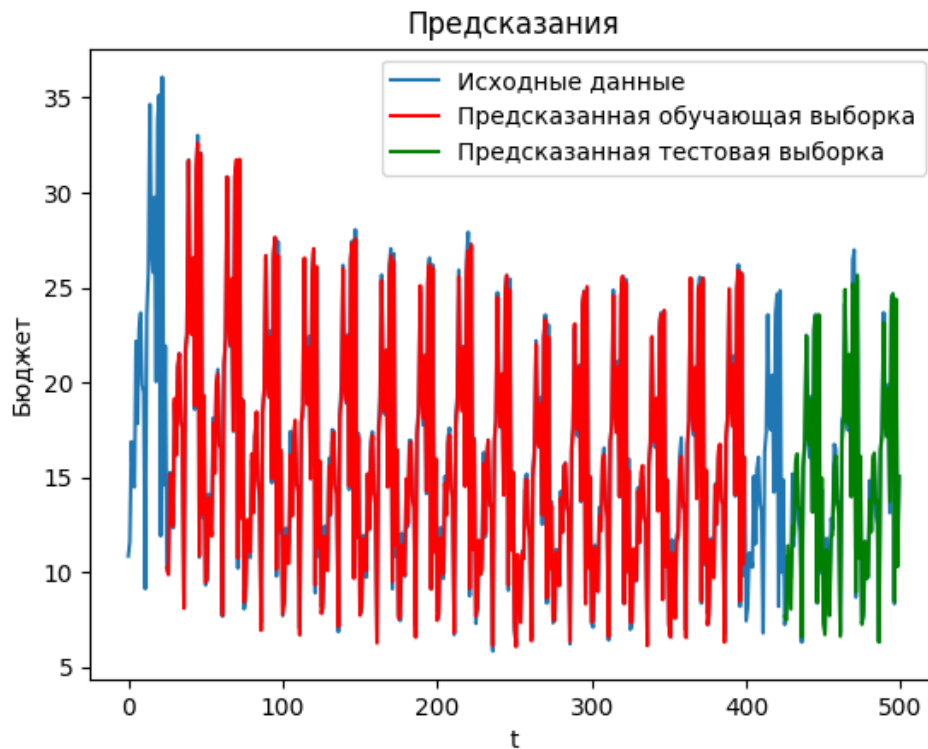


Рисунок 15. График предсказаний значений модели рекурсивной стратегии на тренировочном и тестовом множествах.

Мы видим, что тенденции угадываются точно, что является критичным в принятии решений на основе данной модели и когнитивного графа. Однако модель не сразу подстраивается по мультипликативную составляющую временного ряда (см. Рисунок 16).

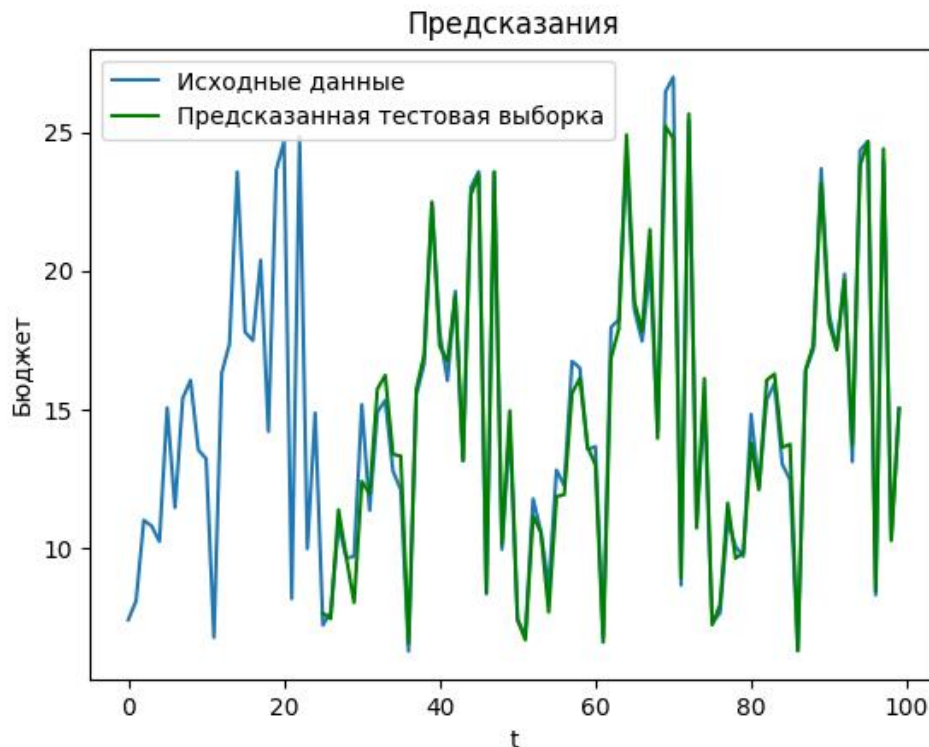


Рисунок 16. График предсказаний значений модели рекурсивной стратегии на тестовом множестве.

При предсказании значений на 25 периодов вперед модель показала относительно точное угадывание тенденций движений графика на первых значениях, однако далее значения не повторяют паттерн графика (см. Рисунок 17). Данное явление является ярким примером накопления ошибки при применении рекурсивной стратегии предсказания. При добавлении новых актуальных значений в модель данная проблема будет устраняться, что демонстрирует график предсказанных значений на тестовом множестве.

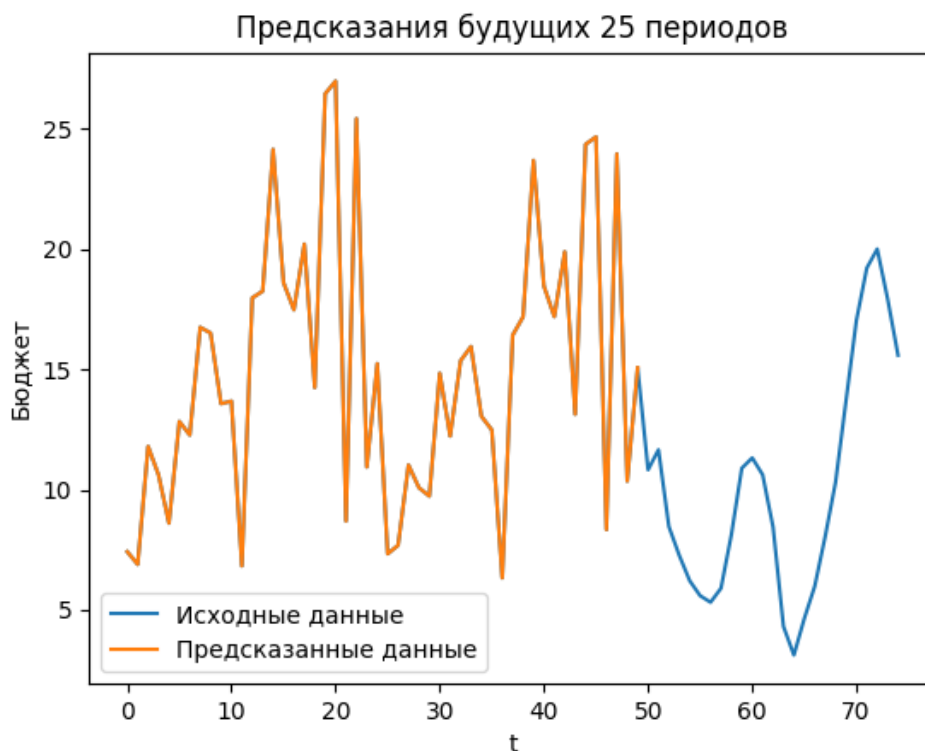


Рисунок 17. График предсказаний будущих значений модели рекурсивной стратегии.

Далее по рекурсивной стратегии модель была обучена на временном ряде, в котором был применен переход к разностям. Для избежания переобучения размер скрытого слоя был увеличен до 30, а количество эпох снижено до 500. Модель продемонстрировала несколько сильных всплесков ошибки, но не показала переобучения, о чем свидетельствует график ошибки MSE (см. Рисунок 18).

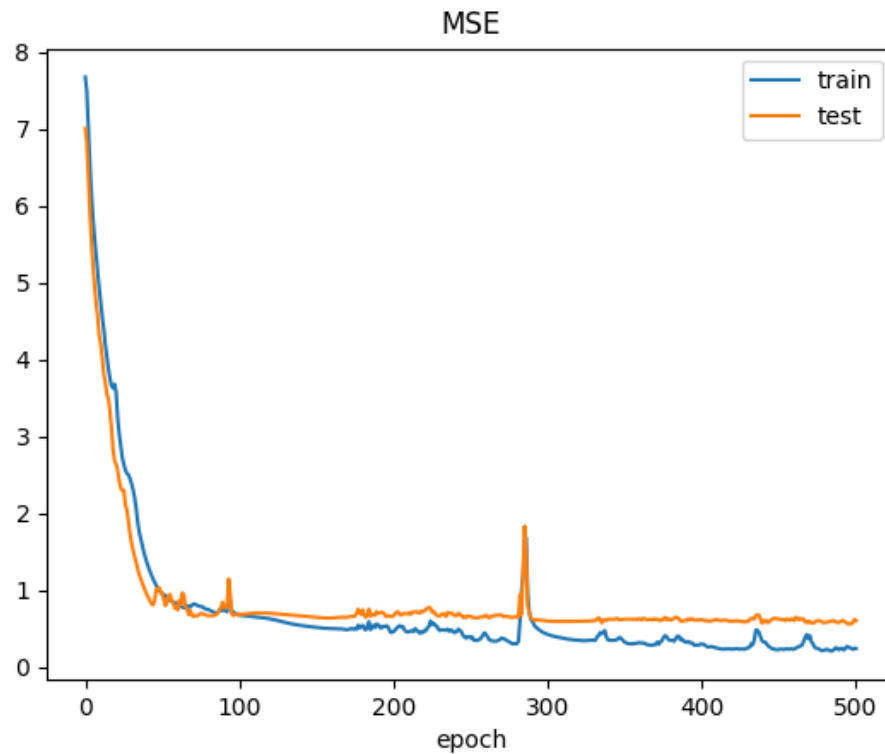


Рисунок 18. График значений функции ошибки в течение обучения модели рекурсивной стратегии на скорректированных данных.

По графику предсказанных значений тренировочного и тестового множества (см. Рисунки 19 - 20) видно, что прогнозы точны и лучше угадывают значения, чем модель, обученная на первоначальных данных, модели удастся предсказать мультипликативную составляющую временного ряда.

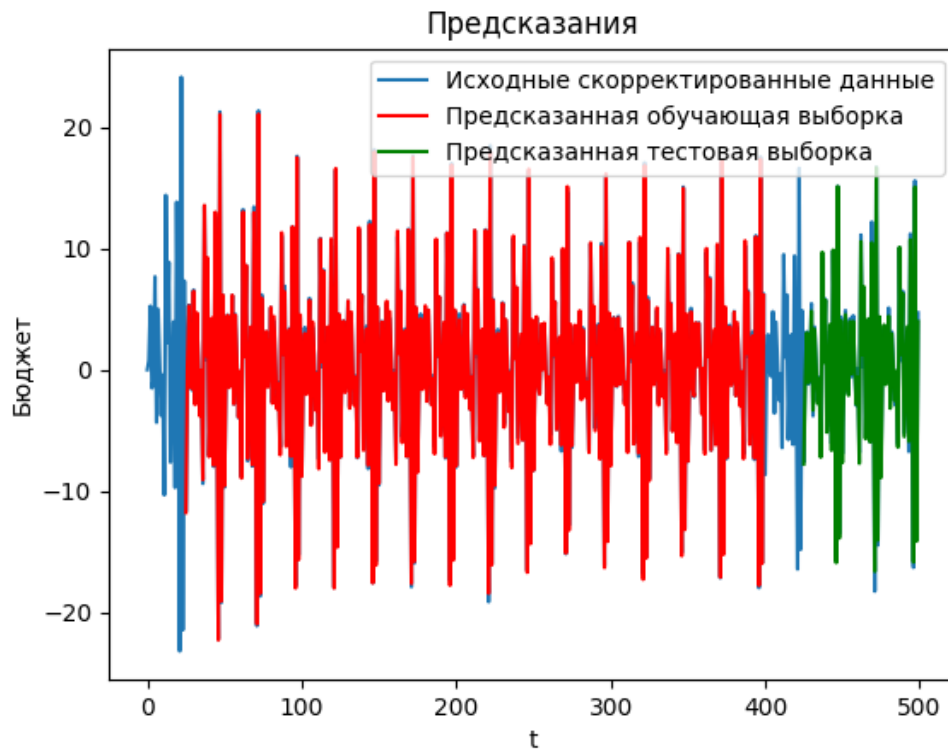


Рисунок 19. График предсказаний значений модели рекурсивной стратегии на скорректированных данных на тренировочном и тестовом множествах.

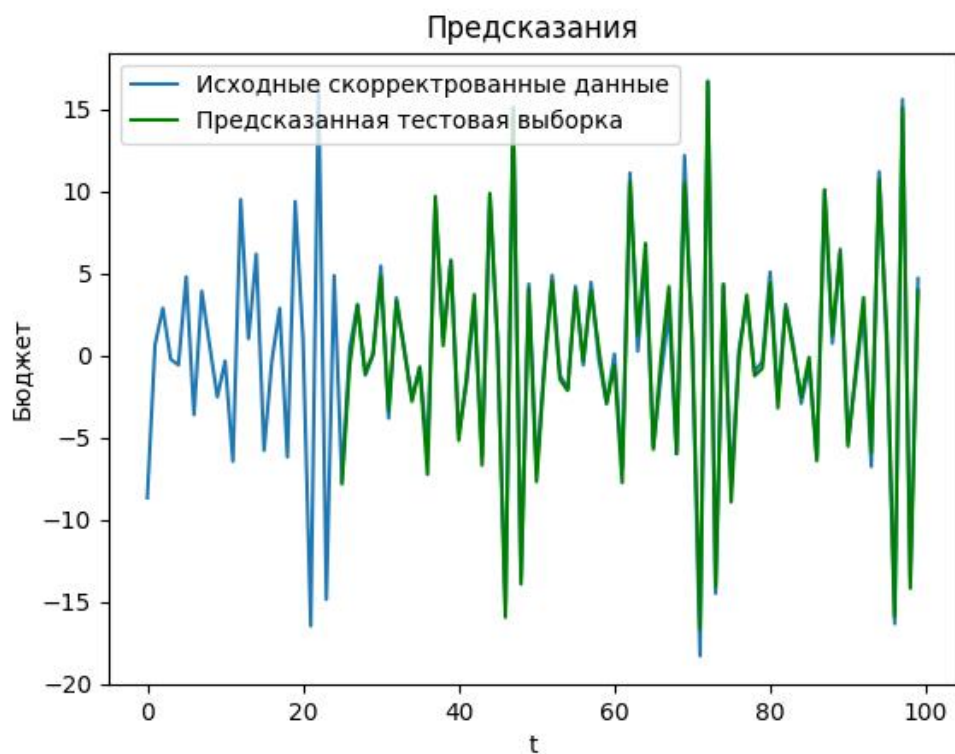


Рисунок 20. График предсказаний значений модели рекурсивной стратегии на скорректированных данных на тестовом множестве.

Более того, при предсказании будущих 25 значений результат оказался похож на реальный паттерн графика. Однако модель все равно не может подстроиться под амплитуду временного ряда (см. Рисунок 21).

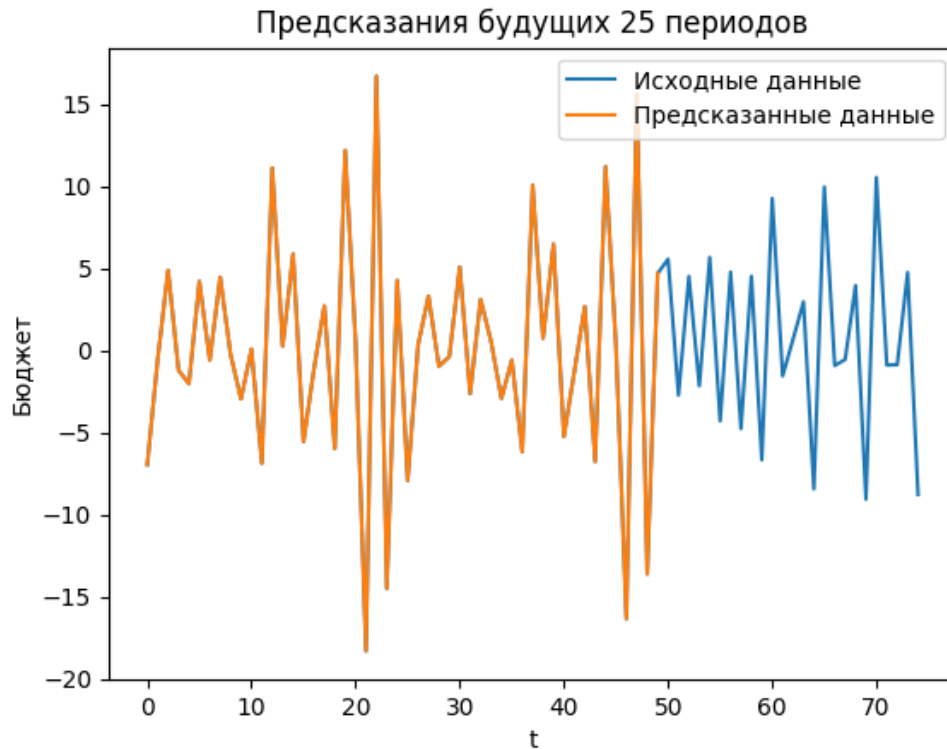


Рисунок 21. График предсказаний будущих значений модели рекурсивной стратегии на скорректированных данных.

Наконец, на изначальном временном ряде была обучена модель, получающая на вход тензоры размерности 5 на 1, а на выходе – 25 на 1. Обучение походило без всплесков MSE, однако ее значения достаточно высоки, что гораздо хуже чем в предыдущих моделях.

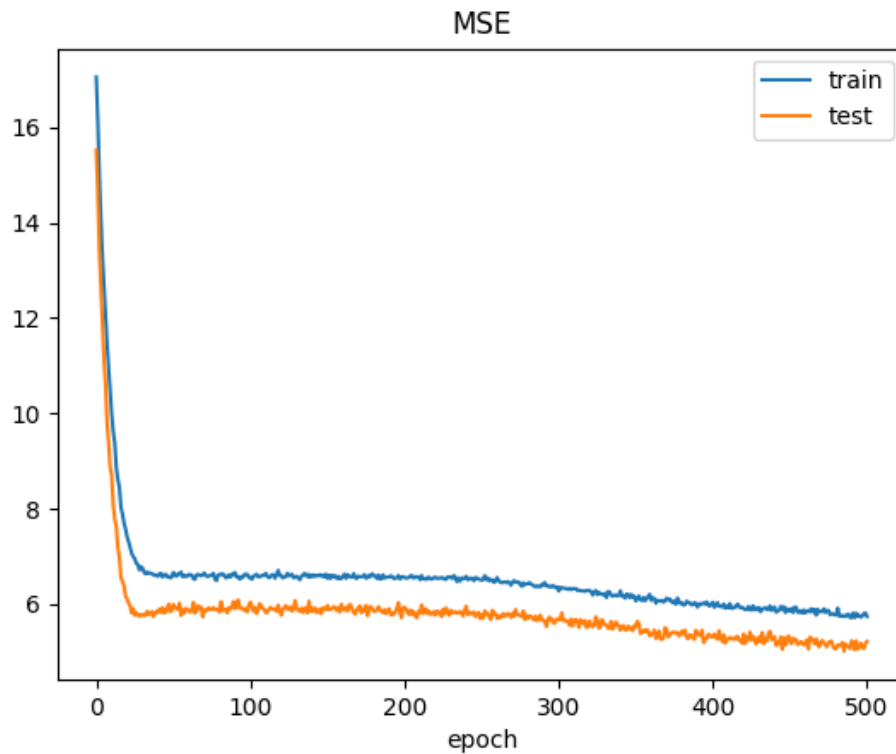


Рисунок 22. График значений функции ошибки в течение обучения модели ММО стратегии.

Предсказания будущих 25 значений в данной модели оказалось самым неточным на всех интервалах, кроме самого первого (см. Рисунок 23). Это обусловлено высокими значениями ошибки даже на тренировочном множестве.

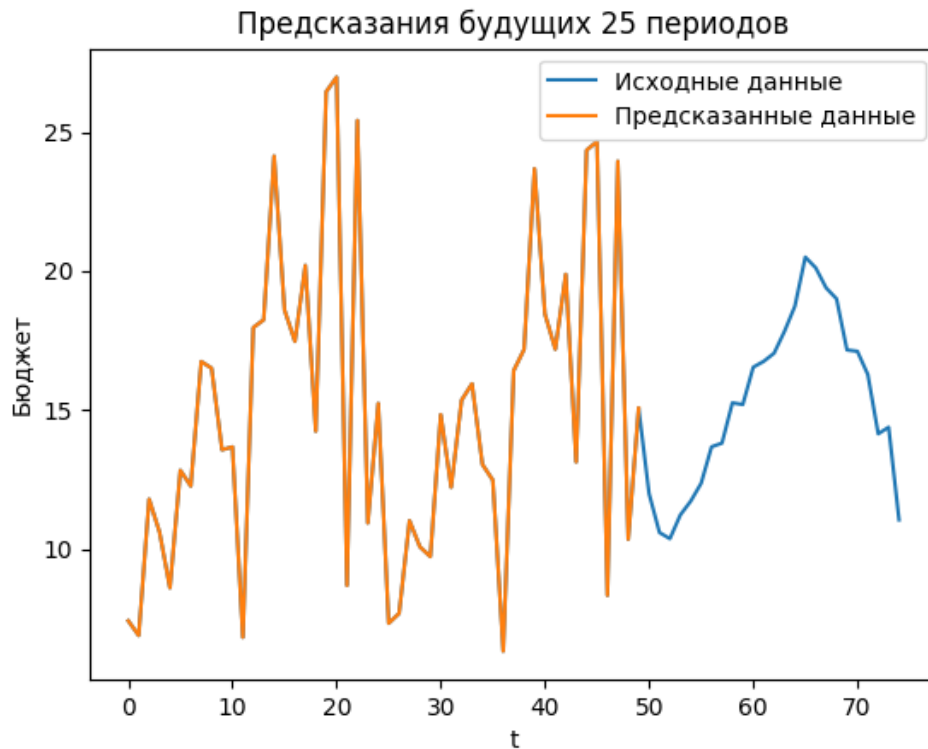


Рисунок 23. График предсказаний будущих значений модели MIMO стратегии.

Поскольку большинство исследований в области предсказаний временных рядов при помощи LSTM модуля свидетельствуют о том, что стратегия MIMO является эффективной, было принято решение применить ее к измененному временному ряду, в котором хранятся дельты между соседними значениями исходного временного ряда.

Мы видим, что модель не переобучилась, а кривые обучения без сильных всплесков, в отличие от моделей рекурсивной стратегии (см. Рисунок 24).

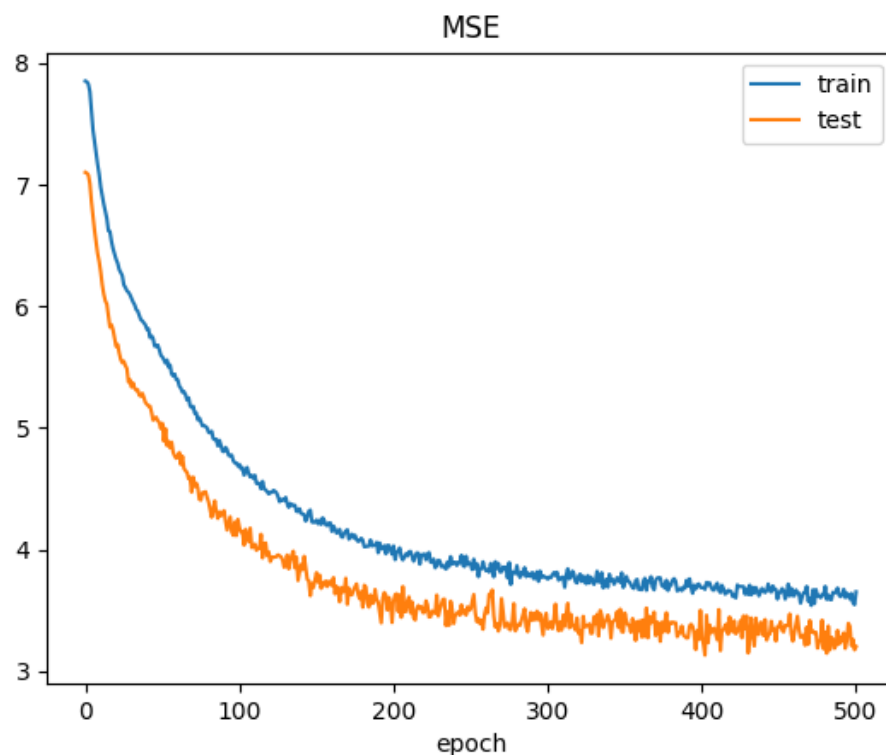


Рисунок 24. График значений функции ошибки в течение обучения модели ММО стратегии на скорректированных данных.

При предсказании будущих 25 значений данная модель продемонстрировала самый точных вариант из всех обученных моделей. Она угадывает паттерн исходного графика и учитывает его амплитуду (см. Рисунок 25).

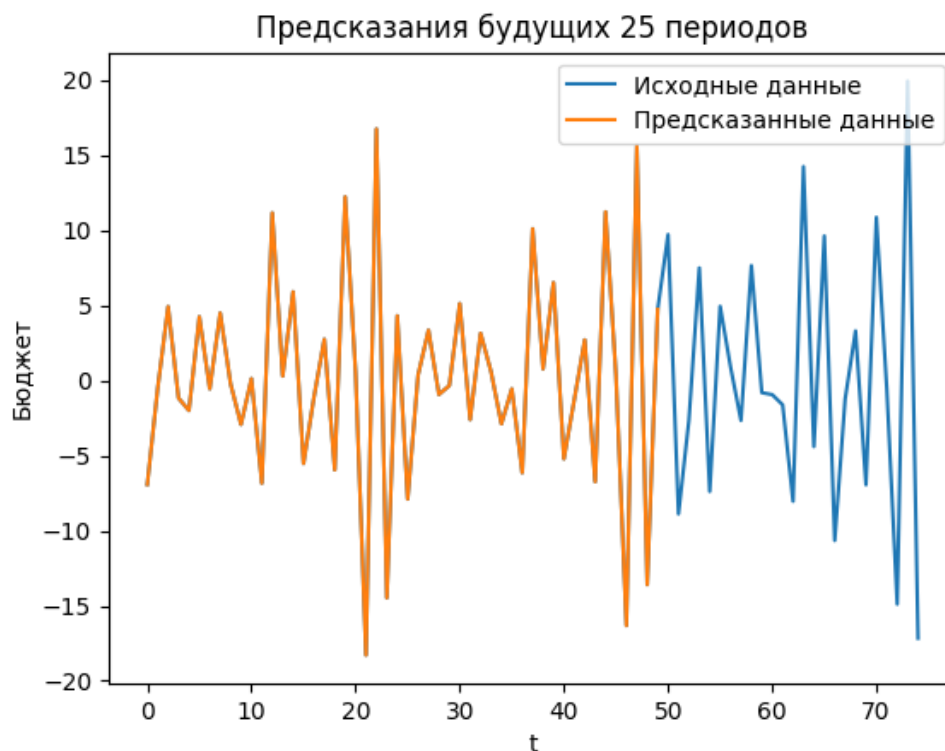


Рисунок 25. График предсказаний будущих значений модели MIMO стратегии на скорректированных данных.

3.3. Принятия решений на основе алгоритма прогнозирования социально-экономического процесса

После обучения модели, способной прогнозировать значения «бюджетов» на вершинах когнитивного графа, необходимо объяснить алгоритм принятия решений на основе модели и когнитивного графа.

Во-первых, эксперты должны понимать, что составленный когнитивный граф по определённому социально-экономическому процессу может претерпевать изменения ввиду кризисов или определенных метаморфоз в общества (реформы, революции и т. д.). Поэтому при получении новых реальных значений на вершине графа необходимо следить за полярность ребра, коррелирует ли она с модельным значением ребра. Если она изменилась, то необходимо в первую очередь пересмотреть модель когнитивного графа, найти лишние или недостающие связи и вершины.

Во-вторых, модель предсказания необходимо периодически обучать заново при накоплении новых данных. Желательно, при каждом новом полученном актуальном значении «бюджета» запускать новый цикл обучения, чтоб модель калибровалась к современным изменениям. При накоплении большой базы данных модель сможет предсказывать значения точнее. Также при изменении когнитивного графа модель также надо обучать заново, так как закономерности поменялись.

В перспективе необходима разработка программного обеспечения, которая позволит наглядно сопоставлять когнитивные графы и результаты предсказания рекуррентной нейронной сети.

Прототип веб-приложения был написан на языке программирования Python при помощи фреймворка Dash (см. Рисунок 26). Он позволяет полноценно использовать возможности языков программирования HTML, CSS и JavaScript без их глубокого изучения.

В данном приложении выводится на экран построенный когнитивный граф, с наглядной визуальной классификацией ребер (красные – положительные, синие – отрицательные). У пользователя есть возможность в модель загрузить новые данные, которые хранятся в определенных вершинах.

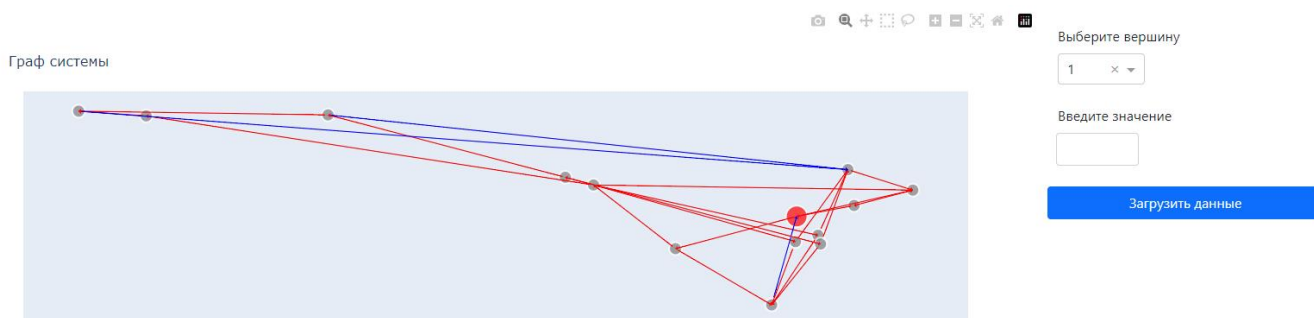


Рисунок 26. Прототип приложения для пользователей-экспертов по анализу когнитивного графа.

Заключение

Результаты данного исследования показывают, что разработанный математический аппарат, основанный на применении свойств сбалансированности орграфов, может использоваться для изучения и описания динамики процессов в социально-экономической системе. Полученные выводы подтвердили необходимость обнаружения закономерностей устойчивого развития социально-экономических систем и разработки доступной и относительно дешевой методики измерения устойчивого развития, которая позволяет проводить надежный анализ с помощью сбалансированных свойств графов.

В ходе данного исследования был проведен анализ системы оценки мер государственной поддержки в Российской Федерации и выявлены слабые места, которые возникли в современных условиях санкций. Также был построен наиболее эффективный когнитивный граф, который позволяет на перспективе избегать больших финансовых потерь в оценке эффективности мер государственной поддержки на многие годы вперед.

Таким образом, в рамках настоящего исследования разработан и развит математический аппарат сбалансированных орграфов для анализа динамики экономических процессов в социально-экономических системах, которая подразумевает решение следующих задач:

1. анализ динамики социально-экономических систем на основе определения свойств сбалансированности орграфа, описывающего развитие социально-экономической системы и выяснение характера поведения (колебания или монотонные изменения) показателей, характеризующих социально-экономическую систему;
2. описание изменения качественных показателей одних элементов системы, используя косвенную информацию об изменениях количественных показателей других элементов системы и информацию о взаимном влиянии элементов системы;
3. краткосрочный прогноз взаимодействия отдельных факторов социально-экономической системы.

Предложенный математический аппарат опирается на базовые понятия и свойства теории графов.

Также была построена рекуррентная нейронная сеть с использованием LSTM модуля, которая позволяет предсказывать значения на вершинах графа на один цикл (полный проход по графу) вперед. Точность предсказаний необходимо в дальнейшем улучшать за счет накопления и добавления в модель новых данных.

Итак, формализация социально-экономического процесса в виде когнитивного графа является эффективной для управления процессами как на государственном уровне, так и на уровне небольшого предприятия. Визуальная составляющая помогает менеджерам быстрее ориентироваться в сложных процессах, а полярность ребер (веса) оценивать устойчивость системы, которую они моделируют. Выявление несбалансированных циклов позволяет оперативно и эффективно перестроить процессы в системе так, чтобы данные циклы исчезли или оказывали не столь сильный негативный эффект на процессы.

При анализе когнитивного графа и полученных данных из рекуррентной нейронной сети возможно предсказать неожиданные всплески в социально-экономической системе, подготовить ряд мер, которые будут внедрены при наступлении кризисов, что позволит снизить денежные потери, которые неизбежны в нестабильных ситуациях.

Данная работа выполнена мною самостоятельно

«22» мая 2023 г.



СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ИНТЕРНЕТ-РЕСУРСОВ

1. Долгая краткосрочная память // neerc.ifmo.ru [Электронный ресурс] URL: https://neerc.ifmo.ru/wiki/index.php?title=%D0%94%D0%BE%D0%BB%D0%B3%D0%B0%D1%8F_%D0%BA%D1%80%D0%B0%D1%82%D0%BA%D0%BE%D1%81%D1%80%D0%BE%D1%87%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%8C(дата обращения 24.04.2023)
2. Свиридов О.А., Тарасов Д.А. Факторы развития социально-экономических систем // cyberleninka.ru [Электронный ресурс] URL: <https://cyberleninka.ru/article/n/factory-razvitiya-sotsialno-ekonomicheskikh-sistem>
3. Морозов В.А. Типы и формы взаимодействия социально-экономических систем // nwww.researchgate.net [Электронный ресурс] URL: https://www.researchgate.net/publication/305990122_Tipy_i_formy_vzaimodejstviya_sotsialno-ekonomicheskikh-sistem (дата обращения 14.04.2023)
4. Социология : учебник / А.В. Верецагина, В.В. Ковалев, С.И. Самыгин. — Москва : КНОРУС, 2023. — 382 с. — (Бакалавриат). ISBN 978-5-406-11104-8 <https://book.ru/books/947837>
5. Ткаченко А.А. Сущность и виды социально-экономических отношений // cyberleninka.ru [Электронный ресурс] URL: <https://cyberleninka.ru/article/n/suschnost-i-vidy-sotsialno-ekonomicheskikh-otnosheniy> (дата обращения 13.04.2023)
6. Социально-экономические отношения // bigenc.ru [Электронный ресурс] URL: <https://bigenc.ru/c/sotsial-no-ekonomicheskije-otnosheniia-731a5a> (дата обращения 14.04.2023)
7. Бранский В.П. Социальная синергетика и акмеология: Теория самоорганизации индивидуума и социума / В.П. Бранский, С.Д. Пожарский. - СПб., 2011. – 159 с.
8. Светульников И.С., Светульников С.Г. Методы социально-экономического прогнозирования. Том 1. Теория и методология. — 1. — Москва: Юрайт,

2015. — С. 30. — 351 с. — ISBN 978-5-9916-4903-2, 978-5-9916-4905-6.

9. Прогнозирование социально-экономических процессов:
Учебнометодическое пособие. // Капитанова О.В. – Нижний Новгород:
Нижегородский госуниверситет, 2016. – 74 с.
http://www.unn.ru/books/met_files/Kapitanova.pdf
10. Метод «Дельфи» // ru.wikipedia.org [Электронный ресурс] URL:
https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%C2%AB%D0%94%D0%B5%D0%BB%D1%8C%D1%84%D0%B8%C2%BB
(дата обращения 04.04.2023)
11. Теория графов и классические задачи прикладной математики в эконмике:
учебное пособие / А.А. Кочкаров, Д.В. Яцкин – Москва: КНОРУС, 2021. – 248
с. – (Бакалавриат) ISBN 978-5-406-08688-9 <https://book.ru/books/940478>
12. Г.П. Горелко, Д.И. Коровин Математическое моделирование динамики
изменения качественных показателей социально-экономической системы с
помощью взвешенных орграфов. Известия высших учебных заведений.
Серия: Экономика, финансы и управление производством, 2013. № 4 (18). -
84-91 с.
13. Алгоритм Джонсона// ru.wikipedia.org [Электронный ресурс] URL:
https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%94%D0%B6%D0%BE%D0%BD%D1%81%D0%BE%D0%BD%D0%B0 (дата обращения 04.04.2023)
14. Алгоритм Джонсона // neerc.ifmo.ru [Электронный ресурс] URL:
https://neerc.ifmo.ru/wiki/index.php?title=%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%94%D0%B6%D0%BE%D0%BD%D1%81%D0%BE%D0%BD%D0%B0 (дата обращения 04.04.2023)
15. Ибрагимова Э.И. Сбалансированные графы и их применение. // www.elibrary.ru [Электронный ресурс] URL:
<https://elibrary.ru/item.asp?id=40563334> (дата обращения 07.04.2023)
16. Harary F. Structural Balance: A Generalization of Heider's Theory // Psychological Review. 1956. V. 63 (5). P. 277–293.

- 17.Кулинич А.А. Семиотические когнитивные карты. Ч. 1. Когнитивный и семиотический подходы в информатике и управлении // cyberleninka.ru [Электронный ресурс] URL: <https://cyberleninka.ru/article/n/semioticheskie-kognitivnye-karty-ch-1-kognitivnyy-i-semioticheskiy-podhody-v-informatike-i-upravlenii> (дата обращения 14.04.2023)
- 18.Есекина Б.К., Сапаргали Ш. К вопросу об оценке устойчивости социально-экономических систем // Проблемы устойчивого развития: иллюзии, реальность, прогноз. Материалы шестого постоянно действующего научного семинара «Самоорганизация устойчивых целостностей в природе и обществе». – 2002.
- 19.Имитационное моделирование // ru.wikipedia.org [Электронный ресурс] URL: https://ru.wikipedia.org/wiki/%D0%98%D0%BC%D0%B8%D1%82%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5 (дата обращения 04.04.2023)
- 20.Как лучше обучать RNN для прогнозирования временных рядов? // habr.com [Электронный ресурс] URL: <https://habr.com/ru/articles/701798/> (дата обращения: 26.04.2023)
- 21.A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition // arxiv.org [Электронный ресурс] URL: <https://arxiv.org/abs/1108.3259> (дата обращения: 25.04.2023)
- 22.Mean squared error// en.wikipedia.org [Электронный ресурс] URL: https://en.wikipedia.org/wiki/Mean_squared_error (дата обращения 28.04.2023)
- 23.Документация библиотеки NetworkX // networkx.org [Электронный ресурс] URL: <https://networkx.org/documentation/stable/index.html> (дата обращения: 25.04.2023).
- 24.Документация библиотеки Tqgm // tqdm.github.io [Электронный ресурс] URL: <https://tqdm.github.io/> (дата обращения: 25.04.2023).
- 25.Документация библиотеки Matplotlib // matplotlib.org [Электронный ресурс] URL: <https://matplotlib.org/stable/index.html> (дата обращения: 25.04.2023).

26. Документация библиотеки PyTorch // [pytorch.org](https://pytorch.org/docs/stable/index.html) [Электронный ресурс] URL: <https://pytorch.org/docs/stable/index.html> (дата обращения: 25.04.2023).
27. Документация библиотеки Torchmetrics // torchmetrics.readthedocs.io [Электронный ресурс] URL: <https://torchmetrics.readthedocs.io/en/stable/> (дата обращения: 25.04.2023).
28. Документация библиотеки NumPy // [numpy.org](https://numpy.org/doc/) [Электронный ресурс] URL: <https://numpy.org/doc/> (дата обращения: 28.04.2023).
29. Кулинич А.А. Семиотические когнитивные карты. Ч. 2. Основные определения и алгоритмы // cyberleninka.ru [Электронный ресурс] URL: <https://cyberleninka.ru/article/n/semioticheskie-kognitivnye-karty-ch-2-osnovnye-opredeleniya-i-algoritmy> (дата обращения: 18.04.2023).
30. Доказательная политика // ru.wikipedia.org [Электронный ресурс] URL: [https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BA%D0%B0%D0%B7%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BE%D0%BB%D0%B8%D1%82%D0%B8%D0%BA%D0%B0#:~:text=%D0%94%D0%BE%D0%BA%D0%B0%D0%B7%D0%B0%CC%81%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F%20%D0%BF%D0%BE%D0%BB%D0%B8%CC%81%D1%82%D0%B8%D0%BA%D0%B0%20\(%D0%B0%D0%BD%D0%B3%D0%BB,_%D0%BC%D0%B5%D0%B4%D0%B8%D1%86%D0%B8%D0%BD%D1%8B%20%D0%BD%D0%B0%20%D0%BE%D0%B1%D0%BB%D0%B0%D1%81%D1%82%D1%8C%20%D0%B3%D0%BE%D1%81%D1%83%D0%B4%D0%B0%D1%80%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%B9%20%D0%BF%D0%BE%D0%BB%D0%B8%D1%82%D0%B8%D0%BA%D0%B8](https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BA%D0%B0%D0%B7%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BE%D0%BB%D0%B8%D1%82%D0%B8%D0%BA%D0%B0#:~:text=%D0%94%D0%BE%D0%BA%D0%B0%D0%B7%D0%B0%CC%81%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F%20%D0%BF%D0%BE%D0%BB%D0%B8%CC%81%D1%82%D0%B8%D0%BA%D0%B0%20(%D0%B0%D0%BD%D0%B3%D0%BB,_%D0%BC%D0%B5%D0%B4%D0%B8%D1%86%D0%B8%D0%BD%D1%8B%20%D0%BD%D0%B0%20%D0%BE%D0%B1%D0%BB%D0%B0%D1%81%D1%82%D1%8C%20%D0%B3%D0%BE%D1%81%D1%83%D0%B4%D0%B0%D1%80%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%B9%20%D0%BF%D0%BE%D0%BB%D0%B8%D1%82%D0%B8%D0%BA%D0%B8). (дата обращения 25.03.2023)
31. Галлямова Э. М. Доказательная государственная политика: возможности и ограничения // cyberleninka.ru [Электронный ресурс] URL: <https://cyberleninka.ru/article/n/dokazatelnaya-gosudarstvennaya-politika-vozmozhnosti-i-ogranicheniya> (дата обращения: 25.03.2023).
32. Михайлова О.В., Батоврина Е.В. Перспективы доказательной политики

- в России: взгляд с позиции государственных управленцев // Вестник Санкт-Петербургского университета. Социология. 2022. Т. 15. Вып. 2. С. 123–136.
<https://doi.org/10.21638/spbu12.2022.202>
33. Кузнецов, О.Л. Устойчивое развитие — синтез естественных и гуманитарных наук / О.Л. Кузнецов, П.Г. Кузнецов, Б.Е. Большаков. -М.: РАЕН; Международный университет природы, общества и человека «Дубна», 2001. – 278 с.
34. Рябов В.Б. Управление качеством трудовой жизни в организации с использованием когнитивных карт // Экспериментальная психология. — 2013. — Т. 6, № 3. — С. 110—121.
35. Волошинская А., Комаров В. Доказательная государственная политика: проблемы и перспективы // Вестник ИЭРАН. - 2015. - № 4. - С. 90102.
36. David F Marks. Perspectives on evidence-based practice, 2002
37. Meghan Guevara, Crime and Justice Institute, Implementing Evidence-Based Policy and Practice in Community Corrections. 2009
38. Giada De Marchi, Giulia Lucertini, Alexis Tsoukias. From Evidence-Based Policy-Making to Policy Analytics, 2012
39. Pfeffer J., Sutton R.I. Evidence-based management // Harvard business review. 2006. Vol. 84, no. 1. [Электронный ресурс]
URL: <https://hbr.org/2006/01/evidence-based-management> (дата обращения: 16.11.2022).
40. Brownson R.C., Chiqui J.F., Stamatakis K.A. Understanding evidence-based public health policy // American journal of public health. 2009. Vol. 99, no. 9. P. 1576–1583.

ПРИЛОЖЕНИЕ

Характеристики компьютера и время выполнения программ

Тип процессора: Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz, 1190 МГц, ядер: 4, логических процессоров: 8

Память: 8,00 ГБ

Кэш-память: 6 MB Intel® Smart Cache

Название программы	Время работы
Код 01. Построение и определение циклов в графах.ipynb	7.88 ms
Код 02. Построение алгоритма предсказания значений на вершинах графа.ipynb	7min 54s

Код программ

Код 01. Построение и определение циклов в графах

```
import networkx as nx
import numpy as np

# создание графа Модели 1
g1 = nx.DiGraph()
for i in range(1,9+1):
    g1.add_node(i)
g1.add_weighted_edges_from([(1,2,1),(1,4,1),(4,9,1),(9,1,1),(2,3,1),(3,4,-1),(4,5,-1),
                             (4,6,1),(5,8,1),(5,7,1),(6,8,1),(6,7,1),(7,1,1),(8,1,1)])

# создание графа Модели 2
g2 = nx.DiGraph()
for i in range(1,9+1):
    g2.add_node(i)
g2.add_weighted_edges_from([(1,2,1),(1,4,1),(4,9,1),(9,1,1),(2,3,1),(3,4,1),(3,5,-1),(4,5,-1),
                             (4,6,1),(5,8,1),(5,7,1),(6,8,1),(6,7,1),(7,1,1),(8,1,1)])

# создание графа Модели 3
g3 = nx.DiGraph()
```



```

for i in range(1,6+1):
    g3.add_node(i)
g3.add_weighted_edges_from([(1,2,1),(1,4,1),(4,6,1),(6,1,1),(2,3,1),(3,4,-1),(3,5,1),(5,1,-1)])

# создание графа Модели 4
g4 = nx.DiGraph()
for i in range(1,6+1):
    g4.add_node(i)
g4.add_weighted_edges_from([(1,2,1),(1,4,1),(4,6,1),(6,1,1),(2,3,1),(3,4,1),(2,5,1),(5,1,-1)])

# создание графа Модели 5
g5 = nx.DiGraph()
for i in range(1,9+1):
    g5.add_node(i)
g5.add_weighted_edges_from([(1,2,1),(1,4,1),(4,6,1),(6,1,1),(2,3,1),(3,4,1),(2,5,1),(5,1,-1),
                             (3,7,1),(3,8,1),(7,9,1),(8,9,1),(9,4,1)])

# создание графа Модели 6
g6 = nx.DiGraph()
for i in range(1,11+1):
    g6.add_node(i)
g6.add_weighted_edges_from([(1,2,1),(1,4,1),(4,6,1),(6,1,1),(2,3,1),(3,4,1),(2,5,1),(5,1,-1),
                             (3,7,1),(3,8,1),(7,10,1),(10,11,1),(10,9,-1),(8,11,1),(11,9,-1),(9,4,1)])

# создание графа Модели 7
g7 = nx.DiGraph()
for i in range(1,14+1):
    g7.add_node(i)
g7.add_weighted_edges_from([(1,2,1),(1,4,1),(4,6,1),(6,1,1),(2,3,1),(3,4,1),(2,5,1),(5,1,-1),
                             (3,7,1),(3,12,1),(3,14,1),(3,8,1),(7,13,1),(7,10,1),(12,9,1)
                             ,(12,5,1),(13,9,1),(13,5,1),(10,9,-1),(10,11,1)
                             ,(11,9,-1),(14,9,1),(14,5,1),(8,11,1),(9,4,1)])

def find_all_cycles(g):
    w_d = {'сбалансированные':[],'несбалансированные':[]}
    A = nx.to_numpy_array(g)
    try:
        l_cycle = nx.simple_cycles(g)

```

```

for i in l_cycle:
    w = 1
    for c in range(len(i)):
        if c!= len(i)-1:
            w *= A[i[c]-1,i[c+1]-1]
        else:
            w *= A[i[c]-1,i[0]-1]
    if w<0:
        i.append(i[0])
        w_d['несбалансированные'].append(i)
    else:
        i.append(i[0])
        w_d['сбалансированные'].append(i)
except:
    print('Найдены отрицательные циклы')
else: pass
return w_d

```

```

for k,i in enumerate([g1,g2,g3,g4,g5,g6,g7],1):
    print(f'Модель {k}')
    cycles = find_all_cycles(i)
    for c in cycles:
        print(f'{c}: {cycles[c]}')

```

Код 02. Построение алгоритма предсказания значений на вершинах графа

```

# загрузка необходимых библиотек
!pip install dgl torchmetrics

# импорт всех используемых библиотек
import networkx as nx
import numpy as np
import random
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
import torch.utils.data as ut_data

```

```

from tqdm.notebook import tqdm
import matplotlib.pyplot as plt
import copy

# создание графа Модели 7 с весами, которые задаются случайно

g7 = nx.DiGraph()

g7.add_edge(1,2,weight = random.uniform(1, 1.5))
g7.add_edge(1,4,weight = random.uniform(1, 1.5))
g7.add_edge(4,6,weight = random.uniform(1, 1.5))
g7.add_edge(6,1,weight = random.uniform(1, 1.5))
g7.add_edge(2,3,weight = random.uniform(1, 1.5))
g7.add_edge(2,5,weight = random.uniform(1, 1.5))
g7.add_edge(5,1,weight = random.uniform(0,1))
g7.add_edge(3,4,weight = random.uniform(1, 1.5))
g7.add_edge(3,7,weight = random.uniform(1, 1.5))
g7.add_edge(3,12,weight = random.uniform(1, 1.5))
g7.add_edge(3,14,weight = random.uniform(1, 1.5))
g7.add_edge(3,8,weight = random.uniform(1, 1.5))
g7.add_edge(7,13,weight = random.uniform(1, 1.5))
g7.add_edge(7,10,weight = random.uniform(1, 1.5))
g7.add_edge(12,9,weight = random.uniform(1, 1.5))
g7.add_edge(12,5,weight = random.uniform(1, 1.5))
g7.add_edge(13,9,weight = random.uniform(1, 1.5))
g7.add_edge(13,5,weight = random.uniform(1, 1.5))
g7.add_edge(10,9,weight = random.uniform(0,1))
g7.add_edge(10,11,weight = random.uniform(1, 1.5))
g7.add_edge(11,9,weight = random.uniform(0,1))
g7.add_edge(14,9,weight = random.uniform(1, 1.5))
g7.add_edge(14,5,weight = random.uniform(1, 1.5))
g7.add_edge(8,11,weight = random.uniform(1, 1.5))
g7.add_edge(9,4,weight = random.uniform(1, 1.5))

# создание признаков вершин графа (в данном случае бюджет который был получен каждым объектом системы)

# принцип: бюджет вершины источника умножается на вес ребра и получается бюджет вершины получателя

```

```

dict_nodes_feature = {
    1: [10],
    2: [],
    3: [],
    4: [],
    5: [],
    6: [],
    7: [],
    8: [],
    9: [],
    10: [],
    11: [],
    12: [],
    13: [],
    14: []

}

# вектор всех потоков денег
total_budget = []
for _ in range(20):
    for i in g7.edges(data=True):
        # print(dict_nodes_feature[i[0]][-1])
        dict_nodes_feature[i[1]].append(copy.copy(dict_nodes_feature[i[0]][-1])*i[2]['weight'] + random.random())
        total_budget.append(dict_nodes_feature[i[1]][-1])

# график
plt.plot(total_budget)
plt.title('Исходные данные')
plt.xlabel('t')
plt.ylabel('бюджет')
plt.savefig('/content/drive/MyDrive/Colab Notebooks/Диплом/time_series1.png')
plt.show()

## RNN предсказывающая одно будущее значение

```

Формирование выборок

```
def create_dataset_output(dataset, lookback):
    X, y = [], []
    for i in range(len(dataset)-lookback):
        feature = dataset[i:i+lookback]
        target = dataset[i+lookback]
        X.append(feature)
        y.append(target)
    return torch.tensor(X, dtype = torch.float32), torch.tensor(y, dtype = torch.float32)

train_size = int(len(total_budget) * 0.80)
test_size = len(total_budget) - train_size
train, test = np.array(total_budget[:train_size]).reshape(-1,1), np.array(total_budget[train_size:]).reshape(-1,1)
train.shape

lookback = 25 #предсказание на 1 цикл вперед
X_train, y_train = create_dataset_output(train, lookback=lookback)
X_test, y_test = create_dataset_output(test, lookback=lookback)

### Модель

class LSTMModel_mine(nn.Module):
    def __init__(self, input_size, hidden_size, num_layers):
        super().__init__()
        self.input_size = input_size
        self.hidden_size = hidden_size
        self.num_layers = num_layers
        self.lstm = nn.LSTM(input_size=self.input_size, hidden_size=self.hidden_size, num_layers=self.num_layers,
batch_first=True)
        self.linear = nn.Linear(self.hidden_size, 1)
    def forward(self, x):
        h0 = torch.zeros(self.num_layers, x.size(0), self.hidden_size)
        c0 = torch.zeros(self.num_layers, x.size(0), self.hidden_size)
        out, (hn, cn) = self.lstm(x, (h0, c0))
        out = self.linear(out[:, -1, :])
        return out
```

Обучение

```
model = LSTMModel_mine(1, 20, 2)
lr = 0.001
optimizer = optim.Adam(model.parameters(), lr = lr)
loss_fn = nn.MSELoss()
loader = ut_data.DataLoader(ut_data.TensorDataset(X_train, y_train), batch_size=16)

n_epochs = 1000

train_loss, test_loss = [], []
for epoch in tqdm(range(n_epochs+1)):
    model.train()
    for X_batch, y_batch in loader:
        y_pred = model(X_batch)
        loss = loss_fn(y_pred, y_batch)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
    # Оценка
    model.eval()
    with torch.no_grad():
        y_pred = model(X_train)
        train_rmse = np.sqrt(loss_fn(y_pred, y_train))
        train_loss.append(train_rmse)
        y_pred = model(X_test)
        test_rmse = np.sqrt(loss_fn(y_pred, y_test))
        test_loss.append(test_rmse)
    if epoch % 100 == 0:
        print("Epoch %d: train RMSE %.4f, test RMSE %.4f" % (epoch, train_rmse, test_rmse))
```

Результаты обучения

```
plt.plot(train_loss, label = 'train')
plt.plot(test_loss, label = 'test')
plt.legend()
plt.title('MSE')
plt.xlabel('epoch')
```

```
plt.savefig('/content/drive/MyDrive/Colab Notebooks/Диплом/mse1.png')
plt.show()
```

```
with torch.no_grad():
```

```
    train_plot = (np.ones(len(total_budget)+100) * np.nan).reshape(-1,1)
    y_pred = model(X_train)
    y_pred = y_pred.reshape(-1,1)
    train_plot[lookback:train_size] = model(X_train).reshape(-1,1)
    test_plot = (np.ones_like(total_budget) * np.nan).reshape(-1,1)
    test_plot[train_size+lookback:len(total_budget)] = model(X_test).reshape(-1,1)
```

```
# График
```

```
plt.plot(total_budget, label = 'Исходные данные')
plt.plot(train_plot, c='r', label = 'Предсказанная обучающая выборка')
plt.plot(test_plot, c='g', label = 'Предсказанная тестовая выборка')
plt.title('Предсказания')
plt.xlabel('t')
plt.ylabel('Бюджет')
plt.legend()
plt.savefig('/content/drive/MyDrive/Colab Notebooks/Диплом/predicted1.png')
plt.show()
```

```
plt.plot(total_budget[400:], label = 'Исходные данные')
plt.plot(test_plot[400:], c='g', label = 'Предсказанная тестовая выборка')
plt.title('Предсказания')
plt.xlabel('t')
plt.ylabel('Бюджет')
plt.legend()
plt.savefig('/content/drive/MyDrive/Colab Notebooks/Диплом/predicted_test1.png')
plt.show()
```

```
### Предсказание на 25 периодов вперед
```

```
model.eval()
```

```
input_seq = torch.Tensor(X_test).reshape(-1,1).unsqueeze(0)
```

```
n = 25
```

```

predicted_s = []
for i in range(n):
    with torch.no_grad():
        output = model(input_seq)
        predicted_value = output.item()
    input_seq = torch.cat((input_seq[:, 1:, :], torch.Tensor([[[[predicted_value]]])), dim=1)
    predicted_s.append(predicted_value)
    print("Predicted value for period {}: {}".format(i+1, predicted_value))

dict_nodes_feature_pred = {
    1: [],
    2: [],
    3: [],
    4: [],
    5: [],
    6: [],
    7: [],
    8: [],
    9: [],
    10: [],
    11: [],
    12: [],
    13: [],
    14: []
}

total_budget_pred = []
for k,i in enumerate(g7.edges(data=True)):
    dict_nodes_feature_pred[i[1]].append(predicted_s[k])
    total_budget_pred.append(dict_nodes_feature_pred[i[1]][-1])

plt.plot(total_budget[450:]+predicted_s, label = 'Исходные данные')
plt.plot(total_budget[450:], label = 'Предсказанные данные')
plt.title('Предсказания будущих 25 периодов')
plt.xlabel('t')
plt.ylabel('Бюджет')
plt.legend()
plt.savefig('/content/drive/MyDrive/Colab Notebooks/Диплом/predicted_future1.png')

```



```

plt.show()

## RNN предсказывающая одно будущее значение при временном ряде разностей

### Формирование выборок

total_budget_deltas = np.array(total_budget) - np.roll(total_budget, 1)
total_budget_deltas[0] = 0
total_budget_deltas

train_size = int(len(total_budget_deltas) * 0.80)
test_size = len(total_budget_deltas) - train_size
train, test = np.array(total_budget_deltas[:train_size]).reshape(-1,1),
np.array(total_budget_deltas[train_size:]).reshape(-1,1)
train.shape

lookback = 25 #предсказание на 1 цикл вперед
X_train, y_train = create_dataset_output(train, lookback=lookback)
X_test, y_test = create_dataset_output(test, lookback=lookback)
print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
print(X_train.dtype, y_train.dtype)
print(X_test.dtype, y_test.dtype)

### Обучение

model = LSTMModel_mine(1, 30, 2)
lr = 0.001
optimizer = optim.Adam(model.parameters(), lr = lr)
loss_fn = nn.MSELoss()
loader = ut_data.DataLoader(ut_data.TensorDataset(X_train, y_train), batch_size=16)

n_epochs = 500

train_loss, test_loss = [], []
for epoch in tqdm(range(n_epochs+1)):
    model.train()

```

```

for X_batch, y_batch in loader:
    y_pred = model(X_batch)
    loss = loss_fn(y_pred, y_batch)
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()

# оценка
model.eval()
with torch.no_grad():
    y_pred = model(X_train)
    train_rmse = np.sqrt(loss_fn(y_pred, y_train))
    train_loss.append(train_rmse)
    y_pred = model(X_test)
    test_rmse = np.sqrt(loss_fn(y_pred, y_test))
    test_loss.append(test_rmse)
if epoch % 50 == 0:
    print("Epoch %d: train RMSE %.4f, test RMSE %.4f" % (epoch, train_rmse, test_rmse))

```

Результаты обучения

```

plt.plot(train_loss, label = 'train')
plt.plot(test_loss, label = 'test')
plt.legend()
plt.title('MSE')
plt.xlabel('epoch')
plt.savefig('/content/drive/MyDrive/Colab Notebooks/Диплом/mse2.png')
plt.show()

```

```

with torch.no_grad():
    train_plot = (np.ones(len(total_budget_deltas)) * np.nan).reshape(-1,1)
    y_pred = model(X_train)
    y_pred = y_pred.reshape(-1,1)
    train_plot[lookback:train_size] = model(X_train).reshape(-1,1)
    test_plot = (np.ones_like(total_budget_deltas) * np.nan).reshape(-1,1)

    test_plot[train_size+lookback:len(total_budget_deltas)] = model(X_test).reshape(-1,1)

```

График

```

plt.plot(total_budget_deltas, label = 'Исходные скорректированные данные')
plt.plot(train_plot, c='r', label = 'Предсказанная обучающая выборка')
plt.plot(test_plot, c='g', label = 'Предсказанная тестовая выборка')
plt.title('Предсказания')
plt.xlabel('t')
plt.ylabel('Бюджет')
plt.legend()
plt.savefig('/content/drive/MyDrive/Colab Notebooks/Диплом/predicted2.png')
plt.show()

plt.plot(total_budget_deltas[400:], label = 'Исходные скорректированные данные')
plt.plot(test_plot[400:], c='g', label = 'Предсказанная тестовая выборка')
plt.title('Предсказания')
plt.xlabel('t')
plt.ylabel('Бюджет')
plt.legend()
plt.savefig('/content/drive/MyDrive/Colab Notebooks/Диплом/predicted_test2.png')
plt.show()

```

Предсказание на 25 периодов вперед

```

model.eval()

input_seq = torch.Tensor(X_test).reshape(-1,1).unsqueeze(0)

n = 25
predicted = []
for i in range(n):
    with torch.no_grad():
        output = model(input_seq)
        predicted_value = output.item()
        input_seq = torch.cat((input_seq[:, 1:, :], torch.Tensor([[[predicted_value]]])), dim=1)
        predicted.append(predicted_value)
    print("Predicted value for period {}: {}".format(i+1, predicted_value))

dict_nodes_feature_pred = {
    1: [],
    2: [],

```

```

3: [],
4: [],
5: [],
6: [],
7: [],
8: [],
9: [],
10: [],
11: [],
12: [],
13: [],
14: []
}
total_budget_pred = []
for k,i in enumerate(g7.edges(data=True)):
    dict_nodes_feature_pred[i[1]].append(predicteds[k])
    total_budget_pred.append(dict_nodes_feature_pred[i[1]][-1])

plt.plot(total_budget_deltas[450:].tolist()+predicteds, label = 'Исходные данные')
plt.plot(total_budget_deltas[450:].tolist(), label = 'Предсказанные данные')
plt.title('Предсказания будущих 25 периодов')
plt.xlabel('t')
plt.ylabel('Бюджет')
plt.legend()
plt.savefig('/content/drive/MyDrive/Colab Notebooks/Диплом/predicted_future2.png')
plt.show()

## RNN предсказывающая несколько будущих значений

### Формирование выборок

def create_dataset_output_1(dataset, lookback, lookfuture):
    X, y = [], []
    for i in range(len(dataset)-lookback-lookfuture):
        feature = dataset[i:i+lookback]
        target = dataset[i+lookback:i+lookback+lookfuture]
        X.append(feature)
        y.append(target)

```

```

return torch.tensor(X, dtype = torch.float32), torch.tensor(y, dtype = torch.float32).squeeze()

train_size = int(len(total_budget) * 0.80)
test_size = len(total_budget) - train_size
train, test = np.array(total_budget[:train_size]).reshape(-1,1), np.array(total_budget[train_size:]).reshape(-1,1)
train.shape

lookback = 5
lookfuture = 25 #предсказание на n циклов вперед
X_train, y_train = create_dataset_output_1(train, lookback=lookback, lookfuture = lookfuture)
X_test, y_test = create_dataset_output_1(test, lookback=lookback, lookfuture = lookfuture)

### Обучение

model = LSTMModel_mine(1,25, 30, 2)
lr = 0.001
optimizer = optim.Adam(model.parameters(), lr = lr)
loss_fn = nn.MSELoss()
loader = ut_data.DataLoader(ut_data.TensorDataset(X_train, y_train), batch_size=16)

n_epochs = 500

train_loss, test_loss = [], []
for epoch in tqdm(range(n_epochs+1)):
    model.train()
    for X_batch, y_batch in loader:
        y_pred = model(X_batch)
        loss = loss_fn(y_pred, y_batch)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
    # Оценка
    model.eval()
    with torch.no_grad():
        y_pred = model(X_train)
        train_rmse = np.sqrt(loss_fn(y_pred, y_train))
        train_loss.append(train_rmse)

```

```

y_pred = model(X_test)
test_rmse = np.sqrt(loss_fn(y_pred, y_test))
test_loss.append(test_rmse)
if epoch % 50 == 0:
    print("Epoch %d: train RMSE %.4f, test RMSE %.4f" % (epoch, train_rmse, test_rmse))

```

Результаты обучения

```

plt.plot(train_loss, label = 'train')
plt.plot(test_loss, label = 'test')
plt.legend()
plt.title('MSE')
plt.xlabel('epoch')
plt.savefig('/content/drive/MyDrive/Colab Notebooks/Диплом/mse3.png')
plt.show()

```

```

with torch.no_grad():
    train_plot = (np.ones(len(total_budget)+100) * np.nan).reshape(-1,1)
    y_pred = model(X_train)
    y_pred = y_pred.reshape(-1,1)
    test_plot = (np.ones_like(total_budget) * np.nan).reshape(-1,1)

    test_plot[train_size+lookback:len(total_budget)] = model(X_test).reshape(-1,1)

```

График

```

plt.plot(total_budget, label = 'Исходные данные')
plt.plot(train_plot, c='r', label = 'Предсказанная обучающая выборка')
plt.plot(test_plot, c='g', label = 'Предсказанная тестовая выборка')
plt.title('Предсказания')
plt.xlabel('t')
plt.ylabel('Бюджет')
plt.legend()
plt.savefig('/content/drive/MyDrive/Colab Notebooks/Диплом/predicted3.png')
plt.show()

```

```

plt.plot(total_budget[400:])
plt.plot(train_plot[400:], c='r')
plt.plot(test_plot[400:], c='g')

```

```
plt.show()
```

```
### Предсказание на 25 периодов вперед
```

```
model.eval()
```

```
input_seq = torch.Tensor(X_test)
```

```
n = 25
```

```
predicteds = []
```

```
with torch.no_grad():
```

```
    output = model(input_seq)
```

```
    predicted_value = output[-1,:].tolist()
```

```
dict_nodes_feature_pred = {
```

```
    1: [],
```

```
    2: [],
```

```
    3: [],
```

```
    4: [],
```

```
    5: [],
```

```
    6: [],
```

```
    7: [],
```

```
    8: [],
```

```
    9: [],
```

```
    10: [],
```

```
    11: [],
```

```
    12: [],
```

```
    13: [],
```

```
    14: []
```

```
}
```

```
total_budget_pred = []
```

```
for k,i in enumerate(g7.edges(data=True)):
```

```
    dict_nodes_feature_pred[i[1]].append(predicted_value[k])
```

```
    total_budget_pred.append(dict_nodes_feature_pred[i[1]][-1])
```

```
plt.plot(total_budget[450:]+predicted_value, label = 'Исходные данные')
```

```
plt.plot(total_budget[450:], label = 'Предсказанные данные')
```

```
plt.title('Предсказания будущих 25 периодов')
plt.xlabel('t')
plt.ylabel('Бюджет')
plt.legend()
plt.savefig('/content/drive/MyDrive/Colab Notebooks/Диплом/predicted_future3.png')
plt.show()
```

RNN предсказывающая несколько будущих значений на скорректированном ряде

Формирование выборок

```
total_budget_deltas = np.array(total_budget) - np.roll(total_budget, 1)
total_budget_deltas[0] = 0
total_budget_deltas
```

```
train_size = int(len(total_budget_deltas) * 0.80)
test_size = len(total_budget_deltas) - train_size
train, test = np.array(total_budget_deltas[:train_size]).reshape(-1,1),
np.array(total_budget_deltas[train_size:]).reshape(-1,1)
train.shape
```

```
lookback = 5
```

```
lookfuture = 25 #предсказание на n циклов вперед
```

```
X_train, y_train = create_dataset_output_1(train, lookback=lookback, lookfuture = lookfuture)
```

```
X_test, y_test = create_dataset_output_1(test, lookback=lookback, lookfuture = lookfuture)
```

Модель

```
class LSTMModel_mine(nn.Module):
```

```
    def __init__(self, input_size, output_size, hidden_size, num_layers):
```

```
        super().__init__()
```

```
        self.input_size = input_size
```

```
        self.output_size = output_size
```

```
        self.hidden_size = hidden_size
```

```
        self.num_layers = num_layers
```

```
        self.lstm = nn.LSTM(input_size=self.input_size, hidden_size=self.hidden_size, num_layers=self.num_layers,
```



```

batch_first=True)
    self.linear = nn.Linear(self.hidden_size, self.output_size)
def forward(self, x):
    h0 = torch.zeros(self.num_layers, x.size(0), self.hidden_size)
    c0 = torch.zeros(self.num_layers, x.size(0), self.hidden_size)
    out, (hn, cn) = self.lstm(x, (h0, c0))
    out = F.dropout(out, 0.3)
    out = self.linear(out[:, -1, :])
    return out

### Обучение

model = LSTMModel_mine(1, 25, 30, 2)
lr = 0.001
optimizer = optim.Adam(model.parameters(), lr=lr)
loss_fn = nn.MSELoss()
loader = ut_data.DataLoader(ut_data.TensorDataset(X_train, y_train), batch_size=16)

n_epochs = 500

train_loss, test_loss = [], []
for epoch in tqdm(range(n_epochs+1)):
    model.train()
    for X_batch, y_batch in loader:
        y_pred = model(X_batch)
        loss = loss_fn(y_pred, y_batch)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
    # Оценка
    model.eval()
    with torch.no_grad():
        y_pred = model(X_train)
        train_rmse = np.sqrt(loss_fn(y_pred, y_train))
        train_loss.append(train_rmse)
        y_pred = model(X_test)
        test_rmse = np.sqrt(loss_fn(y_pred, y_test))
        test_loss.append(test_rmse)

```

```
if epoch % 50 == 0:
    print("Epoch %d: train RMSE %.4f, test RMSE %.4f" % (epoch, train_rmse, test_rmse))
```

```
### Результаты обучения
```

```
plt.plot(train_loss, label = 'train')
plt.plot(test_loss, label = 'test')
plt.legend()
plt.title('MSE')
plt.xlabel('epoch')
plt.savefig('/content/drive/MyDrive/Colab Notebooks/Диплом/mse4.png')
plt.show()
```

```
### Предсказание на 25 периодов вперед
```

```
model.eval()
```

```
input_seq = torch.Tensor(X_test)
```

```
n = 25
```

```
predicted = []
```

```
with torch.no_grad():
```

```
    output = model(input_seq)
```

```
    predicted_value = output[-1,:].tolist()
```

```
dict_nodes_feature_pred = {
```

```
    1: [],
```

```
    2: [],
```

```
    3: [],
```

```
    4: [],
```

```
    5: [],
```

```
    6: [],
```

```
    7: [],
```

```
    8: [],
```

```
    9: [],
```

```
    10: [],
```

```
    11: [],
```

```

12: [],
13: [],
14: []
}
total_budget_pred = []
for k,i in enumerate(g7.edges(data=True)):
    dict_nodes_feature_pred[i[1]].append(predicted_value[k])
    total_budget_pred.append(dict_nodes_feature_pred[i[1]][-1])

plt.plot(total_budget_deltas[450:].tolist()+predicted_value, label = 'Исходные данные')
plt.plot(total_budget_deltas[450:].tolist(), label = 'Предсказанные данные')
plt.title('Предсказания будущих 25 периодов')
plt.xlabel('t')
plt.ylabel('Бюджет')
plt.legend()
plt.savefig('/content/drive/MyDrive/Colab Notebooks/Диплом/predicted_future4.png')
plt.show()

```

Код 03. Макет веб-приложения для ввода актуальных данных

```

import dash
import pandas as pd
from dash import Dash, html, dcc, ctx
from dash.dependencies import Input, Output, State
import plotly.graph_objs as go
import networkx as nx
import dash_bootstrap_components as dbc

g7 = nx.DiGraph()

g7.add_edge(1, 2, weight=1.2)
g7.add_edge(1, 4, weight=1.5)
g7.add_edge(4, 6, weight=1.5)
g7.add_edge(6, 1, weight=1.2)
g7.add_edge(2, 3, weight=1.01)
g7.add_edge(2, 5, weight=1.2)
g7.add_edge(5, 1, weight=-2)
g7.add_edge(3, 4, weight=1.2)

```

```

g7.add_edge(3, 7, weight=1.2)
g7.add_edge(3, 12, weight=1.2)
g7.add_edge(3, 14, weight=1.3)
g7.add_edge(3, 8, weight=1.2)
g7.add_edge(7, 13, weight=1.1)
g7.add_edge(7, 10, weight=1.2)
g7.add_edge(12, 9, weight=1.3)
g7.add_edge(12, 5, weight=1.2)
g7.add_edge(13, 9, weight=1.2)
g7.add_edge(13, 5, weight=1.6)
g7.add_edge(10, 9, weight=-1.4)
g7.add_edge(10, 11, weight=1.2)
g7.add_edge(11, 9, weight=-1)
g7.add_edge(14, 9, weight=1.2)
g7.add_edge(14, 5, weight=1.5)
g7.add_edge(8, 11, weight=1.2)
g7.add_edge(9, 4, weight=1.1)

```

```
pos = nx.layout.spring_layout(g7)
```

```

edge_trace = go.Scatter(
    x=[],
    y=[],
    line=dict(width=0.5, color='#888'),
    hoverinfo='none',
    mode='lines')
for edge in g7.edges():
    x0, y0 = pos[edge[0]]
    x1, y1 = pos[edge[1]]
    edge_trace['x'] += tuple([x0, x1, None])
    edge_trace['y'] += tuple([y0, y1, None])

```

```

node_trace = go.Scatter(
    x=[],
    y=[],
    # text=[],
    mode='markers',
    hoverinfo='text',

```

```

marker=dict(
    color=[],
    size=10,
    line=dict(width=2)))

for node in g7.nodes():
    x, y = pos[node]
    node_trace['x'] += tuple([x])
    node_trace['y'] += tuple([y])

fig = go.Figure(data=[edge_trace, node_trace],
                layout=go.Layout(
                    title='<br>Граф системы',
                    titlefont=dict(size=16),
                    showlegend=False,
                )
            )

drop_down_nodes = dbc.Row([dbc.Label('Выберите вершину', html_for='drop-down-nodes'),
                            dcc.Dropdown(list(range(1, len(g7.nodes())+1)),
                                         value=1,
                                         id='drop-down-nodes',
                                         style = {'width':'60%'}
                                         )
                            ])

input_changes = dbc.Row([dbc.Label('Введите значение', html_for='input'),
                            dbc.Input(type='text', style = {'width':'30%', 'margin-left':'10px'}, id = 'input')
                            ])

button_add = dbc.Row([
    dbc.Button('Загрузить данные', id = 'button-add', className= 'col = 6 mx-auto')
])

app = Dash(external_stylesheets=[dbc.themes.BOOTSTRAP])

```

```

app.layout = dbc.Container([dbc.Row([
    dbc.Col([dcc.Graph(id='Graph', figure=fig)], width=8),
    dbc.Col([html.Br(), drop_down_nodes,html.Br(), input_changes, html.Br(), button_add,
        dcc.Download(id = 'download')], width=2),
    dbc.Col([], width = 2)
])
], fluid=True)

```

```

@app.callback(
    Output('Graph', 'figure'),
    [Input('drop-down-nodes', 'value')])
def display_selected_data(node_from_dr):
    print(node_from_dr, type(node_from_dr))
    edge_trace = go.Scatter(
        x=[],
        y=[],
        line=dict(width=0.5, color='#888'),
        hoverinfo='none',
        mode='lines')
    for edge in g7.edges():
        x0, y0 = pos[edge[0]]
        x1, y1 = pos[edge[1]]
        edge_trace['x'] += tuple([x0, x1, None])
        edge_trace['y'] += tuple([y0, y1, None])

```

```

node_trace = go.Scatter(
    x=[],
    y=[],
    text=[],
    mode='markers',
    hoverinfo='text',
    marker=dict(
        color=[],
        size=[],
        line=dict(width=2)))

```

```

for nodes in g7.nodes():
    x, y = pos[nodes]

```

```

node_trace['x'] += tuple([x])
node_trace['y'] += tuple([y])
node_trace['text'] += tuple([nodes])

for nodes in g7.nodes():
    if nodes == node_from_dr:
        node_trace['marker']['color'] += tuple(['red'])
        node_trace['marker']['size'] += tuple([25])
    else:
        node_trace['marker']['color'] += tuple(['grey'])
        node_trace['marker']['size'] += tuple([15])

df = pd.DataFrame(columns=['start', 'end', 'type'])

for i in g7.edges(data=True):
    if len(df):
        fix_len = len(df)
    else:
        fix_len = 0
    df.loc[fix_len, 'start'] = pos[i[0]]
    df.loc[fix_len, 'end'] = pos[i[1]]
    df.loc[fix_len, 'type'] = '-' if i[2]['weight'] < 1 else '+'
print(df)

fig = go.Figure(data=[edge_trace, node_trace],
                 layout=go.Layout(
                     title='<br>Граф системы',
                     titlefont=dict(size=16),
                     showlegend=False,
                     hovermode='closest',
                     annotations=[dict(
                         arrowwidth=1,
                         arrowcolor="red" if df.loc[i]['type'] == '+' else 'blue',
                         arrowsize=0.8,
                         showarrow=True,
                         arrowhead=1,
                         ax=df.loc[i]['start'][0],
                         ay=df.loc[i]['start'][1],

```

```

        axref='x',
        ayref='y',
        x=df.loc[i]['end'][0],
        y=df.loc[i]['end'][1],
        xref='x',
        yref='y',
    ) for i in range(len(df)),
    xaxis=dict(showgrid=False, zeroline=False, showticklabels=False),
    yaxis=dict(showgrid=False, zeroline=False, showticklabels=False)
))
return fig

@app.callback(
    Output('download', 'data'),
    [Input('button-add', 'n_clicks')],
    [State('input', 'value'), State('drop-down-nodes', 'value')],
    prevent_initial_call=True)
def display_selected_data(button, inp, drop):
    button_clicked = ctx.triggered_id
    inp = int(inp)
    if button_clicked == 'button-add':
        df_for_model = pd.DataFrame(columns= ['start','end','value_end'])
        for edge in g7.edges():
            if len(df_for_model):
                fix_len = len(df_for_model)
            else:
                fix_len = 0
            if edge[1] == drop:
                df_for_model.loc[fix_len, 'start'] = edge[0]
                df_for_model.loc[fix_len, 'end'] = edge[1]
                df_for_model.loc[fix_len, 'value_end'] = inp
        return dcc.send_data_frame(df_for_model.to_csv, filename="some_name.csv")
    return dash.no_update

app.run_server(debug=False)

```


Список файлов

Имя файла	Имя программы
Рисунок 1. Граф Модели 1.	app.diagrams.net
Рисунок 2. Граф Модели 2.	app.diagrams.net
Рисунок 3. Граф Модели 3.	app.diagrams.net
Рисунок 4. Граф Модели 4.	app.diagrams.net
Рисунок 5. Граф Модели 5.	app.diagrams.net
Рисунок 6. Граф Модели 6.	app.diagrams.net
Рисунок 7. Граф Модели 7.	app.diagrams.net
Рисунок 8. Схема работы LSTM модуля	https://neerc.ifmo.ru/wiki
Рисунок 9. Схема работы рекурсивного подхода предсказания значений временного ряда	https://habr.com/ru/articles/701798/
Рисунок 10. Схема работы прямой стратегии предсказания значений временного ряда	https://habr.com/ru/articles/701798/
Рисунок 11. Схема работы ММО стратегии предсказания значений временного ряда	https://habr.com/ru/articles/701798/
Рисунок 12. Схема работы Direct ММО стратегии предсказания значений временного ряда	https://habr.com/ru/articles/701798/
Рисунок 13. Временной ряд признаков вершин.	Код 02. Построение алгоритма предсказания значений на вершинах графа
Рисунок 14. График значений функции ошибки в течение обучения модели рекурсивной стратегии.	Код 02. Построение алгоритма предсказания значений на вершинах графа
Рисунок 15. График предсказаний значений модели рекурсивной стратегии на тренировочном и тестовом множествах.	Код 02. Построение алгоритма предсказания значений на вершинах графа
Рисунок 16. График предсказаний значений модели рекурсивной стратегии на тестовом множестве.	Код 02. Построение алгоритма предсказания значений на вершинах графа
Рисунок 17. График предсказаний будущих значений модели рекурсивной стратегии.	Код 02. Построение алгоритма предсказания значений на вершинах графа

Рисунок 18. График значений функции ошибки в течение обучения модели рекурсивной стратегии на скорректированных данных.	Код 02. Построение алгоритма предсказания значений на вершинах графа
Рисунок 19. График предсказаний значений модели рекурсивной стратегии на скорректированных данных на тренировочном и тестовом множествах.	Код 02. Построение алгоритма предсказания значений на вершинах графа
Рисунок 20. График предсказаний значений модели рекурсивной стратегии на скорректированных данных на тестовом множестве.	Код 02. Построение алгоритма предсказания значений на вершинах графа
Рисунок 21. График предсказаний будущих значений модели рекурсивной стратегии на скорректированных данных.	Код 02. Построение алгоритма предсказания значений на вершинах графа
Рисунок 22. График значений функции ошибки в течение обучения модели ММО стратегии.	Код 02. Построение алгоритма предсказания значений на вершинах графа
Рисунок 23. График предсказаний будущих значений модели ММО стратегии.	Код 02. Построение алгоритма предсказания значений на вершинах графа
Рисунок 24. График значений функции ошибки в течение обучения модели ММО стратегии на скорректированных данных.	Код 02. Построение алгоритма предсказания значений на вершинах графа
Рисунок 25. График предсказаний будущих значений модели ММО стратегии на скорректированных данных.	Код 02. Построение алгоритма предсказания значений на вершинах графа
Рисунок 26. Прототип приложения для пользователей-экспертов по анализу когнитивного графа.	Код 03. Макет веб-приложения для ввода актуальных данных