

Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Научно-образовательный центр "Высшая ИТ школа"

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК

Руководитель ООП
доктор физико-математических
наук, профессор

О.А. Змеев

подпись

« 09 » 06 2023 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

РАЗРАБОТКА СЕРВИСА, РЕАЛИЗУЮЩЕГО БАЙЕСОВСКИЕ СЕТИ ДЛЯ
РЕШЕНИЯ ЗАДАЧИ ПОИСКА ЛОЖНОПОЛОЖИТЕЛЬНЫХ ОШИБОК МЕНЕДЖЕРА
В УПРАВЛЕНИИ ПРОЕКТАМИ ПО РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

по направлению подготовки 09.03.04 Программная инженерия
направленность (профиль) «Программная инженерия»

Фрейдин Валерий Игоревич

Руководитель ВКР
д-р. физ.-мат. наук, профессор
О.А. Змеев

подпись

« 06 » 06 2023 г.

Консультант ВКР
кандидат технических наук,
доцент НОЦ "Высшая ИТ школа"
Д.О. Змеев

подпись

« 05 » 06 2023 г.

Автор работы
Студент группы №971902
В.И. Фрейдин

подпись

« 05 » 06 2023 г.

Министерство науки и высшего образования Российской Федерации.
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
НОЦ «Высшая ИТ школа»

УТВЕРЖДАЮ
руководитель ООП
д-р.физ.-мат. наук, профессор
О.А.Змеев
« 16 » 02 2023 г.

ЗАДАНИЕ

по выполнению выпускной квалификационной работы бакалавра обучающегося
Фрейдину Валерию Игоревичу

(Ф.И.О. обучающегося)

по направлению подготовки Программная инженерия, направленность «Программная инженерия»

1. Тема выпускной квалификационной работы бакалавра

Разработка сервиса, реализующего байесовские сети для решения задачи поиска

ложноположительных ошибок менеджера в управлении проектами по разработке

программного обеспечения

2. Срок сдачи обучающимся выполненной выпускной квалификационной работы:

а) в учебный офис – « 09 » июня 2023 г.

б) в ГЭК – « 10 » июня 2023 г.

3. Исходные данные к работе:

Изучение теоретических основ байесовских сетей и систем поддержки принятия

цели и задачи ВКР, ожидаемые результаты

решений. Разработка алгоритма построения байесовских сетей для системы поддержки

принятия решений. Оптимизация вычисления прямого вероятностного вывода для


получения вероятности правдоподобия мнения менеджера. Разработка данного сервиса.

Организация, по тематике которой выполняется работа

Руководитель выпускной квалификационной работы

Академический руководитель, учебный
офис НОЦ «Высшая ИТ школа»

(должность, место работы)


(подпись)


/ О. А. Змеев

(И.О. Фамилия)

Консультант выпускной квалификационной работы

Доцент, учебный офис
НОЦ «Высшая ИТ школа»

(должность, место работы)


(подпись)

/ Д. О. Змеев

(И.О. Фамилия)

Задание принял к исполнению

« 16 » июня 2023 г.

(дата)


(подпись)

/ В. И. Фрейдин

(И.О. Фамилия)

ОГЛАВЛЕНИЕ

Введение	3
1 Байесовские сети как инструмент анализа данных	6
1.1 Теория: Байесовская сеть.....	6
1.2 Теория: Прямой вероятностный вывод	9
2 Техническая реализация Байесовских сетей в системе поддержки принятия решений.....	11
2.1 Обзор исходной системы.....	13
2.1.1 Хранение основной структуры Байесовской сети	13
2.1.2 Хранение вероятностного вектора	15
2.1.3 Прямой вероятностей вывод в прототипе системы принятия решений	18
2.2 Сравнительный анализ инструментов по работе с Байесовскими сетями	20
2.3 Разработка системы для работы с Байесовскими сетями	22
2.3.1 Хранение структуры Байесовской сети	22
2.3.2 Внедрение рабочих продуктов в систему принятия решений..	25
2.3.3 Получение вероятностного вектора	26
2.3.4 Изменение прямого вероятностного вывод.....	29
2.4 Оптимизации прямого вероятностного вывода в Байесовских сетях	33
2.4.1 Сравнение языков программирования.....	33
2.4.1.1Горутин и как они работают	36
2.4.2 Предобработка данных	38
2.4.3 Параллелизация вычислений.....	39
3 Полученные результаты.....	42
3.1 Реализация взаимодействия с Neo4j	44
3.2 Взаимодействие с внешней системой.....	45
3.3 Результаты.....	48
Заключение.....	50
Список использованных источников.....	51

ВВЕДЕНИЕ

Современный общественный контекст характеризуется увеличением количества информации и растущей сложностью данных, встречающихся в различных научных и практических областях. Этот явный феномен, связанный с происходящими в мире научно-техническими прорывами, информационной революцией и всеобъемлющим проникновением цифровых технологий, имеет существенное влияние на наше понимание окружающего мира и способы его исследования. Данный феномен сталкивает нас с парадоксальным вызовом: с одной стороны, мы обладаем более широким доступом к данным, чем когда-либо прежде, благодаря расширению интернета, цифровизации информации и развитию мощных вычислительных систем; с другой стороны, возникает сложность анализа, интерпретации и использования этих обширных и разнообразных данных.

В силу постоянного роста объема информации, принятие надежных решений в современном мире представляет значительную сложность. Этот вызов в особенности актуален в контексте научных исследований и практических применений, где корректные выводы и эффективные стратегии действий сильно зависят от адекватной обработки и анализа данных.

В условиях растущей сложности и разнообразия проблем, с которыми сталкиваются современные общественные и научные дисциплины, требуется постоянное развитие и применение новых математических методов для эффективного принятия решений. Традиционные подходы и инструменты становятся недостаточными для обработки и анализа объемных данных, а также для построения надежных моделей, способных объяснять и предсказывать явления в реальном мире.

При работе с вероятностным моделированием возникают различные трудности, которые условно можно подразделить на две группы: технические сложности, такие как сложность вычислений и комбинаторных задач, и идейные проблемы, которые связаны с неопределенностью и постановкой задач

с точки зрения математического смысла. Для решение данной проблемы можно использовать Байесовские сети, поскольку в них можно сочетать эмпирические данные с субъективными оценками, что отличает их от других методов моделирования и придает им важные практические преимущества.

Взяв за основную математическую модель Байесовскую сеть, автор диссертации [1] предложил прототип в виде системы поддержки принятия решений, построенного на стандарте OMG Essence [2].

Предложенный инструмент позволяет рассчитать вероятность того, что менеджер допустит ложноположительную ошибку при определении текущего статуса проекта. Стандарт OMG Essence предоставляет набор утверждений, относящихся к проекту, которые менеджер оценивает на предмет их истинности или ложности в данный момент времени. Инструмент вычисляет вероятность ошибки менеджера, сравнивая их оценку с рассчитанной вероятностью истинности каждого утверждения.

В рамках прототипа была построена сеть, состоящая из 646 вершин и суммарно более 1800 связями между ними. Также в рамках прототипа была предложена система, имплементирующая байесовские сети в систему, но она не удовлетворяла требованию к скорости выполнения прямого вероятностного вывода.

По причине того, что предложенный вариант системы не подходил для реализации удобного в использовании инструмента для поддержки принятия решений в сфере разработки программного обеспечения, было принято решение реализовать новую систему, которая бы удовлетворяла данным требованиям.

Целью данной выпускной квалификационной работы является разработка сервиса, реализующего Байесовские сети для решения задачи поиска ложноположительных ошибок менеджера.

Для достижения этой цели были поставлены следующие задачи:

1. Изучить алгоритмические основы применения байесовских сетей и прямого вероятностного вывода для поиска ложноположительных ошибок менеджера.
2. Проанализировать инструменты для работы с байесовскими сетями.
3. Реализовать сервис для интеграции байесовских сетей в систему поддержки принятия решений.
4. Оптимизировать алгоритм прямого вероятностного вывода до “разумного” времени выполнения.

1 БАЙЕСОВСКИЕ СЕТИ КАК ИНСТРУМЕНТ АНАЛИЗА ДАННЫХ

Байесовские сети представляют собой мощный инструмент для анализа данных в условиях неопределенности. В данном параграфе будут рассмотрены основы Байесовских сетей и особенности прямого вероятностного вывода.

1.1 Теория: Байесовская сеть

Байесовские сети представляют собой графические модели, используемые для описания вероятностных зависимостей между множеством переменных, а также для выполнения вероятностного вывода на основе доступной информации. Такие модели включают графическую структуру, отражающую отношения между переменными, и вероятностные таблицы, описывающие условные вероятности, которые используются для расчета вероятностей событий в рамках сети.

Байесовские сети находят свое применение в областях, характеризующихся присущей им неопределенностью, которая может возникать из-за различных факторов, таких как:

- неполное понимания предметной области;
- неполные знания;
- когда задача характеризуется случайностью.

Поэтому байесовские сети используются для моделирования ситуаций, включающих некоторую степень неопределенности.

Использование причинно-следственных связей может упростить процесс оценки вероятностей событий. На практике оценка обычно проводится в направлении от "наблюдателя" или "причины" к "наблюдению" или "следствию". Это направление часто является более сложным для оценки по сравнению с противоположным направлением, " следствие -> эффект", которое предполагает оценку вероятностей с точки зрения "следствия".

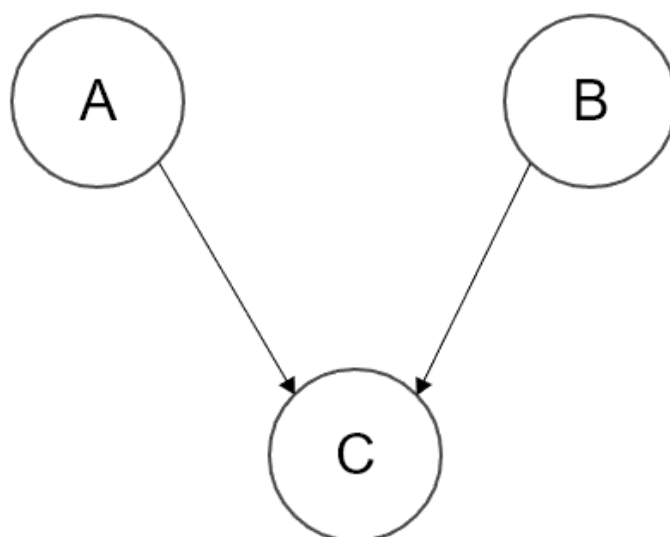


Рисунок 1 – Пример простой Байесовской сети

Рассмотрим пример сети, в которой вероятность пребывания вершины "C" в различных состояниях (C_j) зависит от состояний (A_j, B_j) вершин "A" и "B" и, в общем случае, определяется выражением:

$$P(C_k) = \sum_i \sum_j P(C_k | A_i, B_j) \times P(A_i, B_j) \quad (1)$$

Где $P(C_k | A_i, B_j)$ – условная вероятность пребывания вершины "C" в состоянии (C_k) в зависимости от состояний (A_j, B_j), в которых находятся вершин "A" и "B".

Но так как события, представленные вершинами "A" и "B" являются независимыми, то есть между ними отсутствует причинно-следственная связь, то их совместная вероятность может быть представлена в виде:

$$P(A_i, B_j) = P(A_i) * P(B_j) \quad (2)$$

Рассмотрим пример более сложной сети, которая является частью сети построенной в рамках объекта данной ВКР.



Рисунок 2 – Пример сложной Байесовской сети

Для оценки вероятности пребывания вершины "O1" в различных состояниях следует использовать приведенное выше выражение. Что же касается вершин "O13" и "O12", то для их оценки будем использовать выражения, аналогичные тому, что и для вычисления $P(C_k)$, тогда:

$$P(O12_i) = \sum_m P(O12_i|O11_m) \times P(O11_m) \quad (3)$$

$$P(O13_j) = \sum_m P(O13_j|O12_m) \times P(O12_m) \quad (4)$$

Применение Байесовских сетей в качестве инструмента принятия решений требует использования прямого вероятностного вывода. Прямой вероятностный вывод является фундаментальной процедурой в Байесовских сетях, которая позволяет оценивать вероятности различных событий и принимать решения на основе имеющихся данных и модели.

1.2 Теория: Прямой вероятностный вывод

Прямой вероятностный вывод [3] в Байесовских сетях основан на теореме Байеса и использует условные вероятности для обратного вычисления неизвестных вероятностей. Этот процесс начинается с известных фактов или наблюдений и использует модель Байесовской сети для вычисления вероятностей интересующих событий.

Чтобы подробно изучить прямой вероятностный вывод, необходимо рассмотреть пример оценки достоверности гипотезы с использованием теоремы Байеса, а также обосновать ее применение в экспертных системах.

Для начала необходимо представить более расширенную формулу Байеса, учитывающую событие E , которое связано с событиями H_1, H_2, \dots, H_n . При условии, что одно из событий H_1, H_2, \dots, H_n произошло, вероятности события E обозначаются как $P(E/H_1), P(E/H_2), \dots, P(E/H_n)$. Предположим, что событие E уже произошло. В этом случае вероятность того, что любое из событий H_i ($i = 1, \dots, n$) произошло, определяется по Формуле 5:

$$P(H_i | E) = (P(E | H_i) * P(H_i)) / (P(E | H_1) * P(H_1) + P(E | H_2) * P(H_2) + \dots + P(E | H_n) * P(H_n)) = P(EH_i) / P(E) \quad (5)$$

H_1, H_2, \dots, H_n представляют гипотезы, а событие E является свидетельством. Вероятности гипотез $P(H_i)$ до учета свидетельства (т.е. до того, как произошло событие E) называются априорными, а вероятности $P(H_i/E)$ после учета свидетельства - апостериорными. Величина $P(EH_i)$ обозначает совместную вероятность событий E и H_i , то есть вероятность того, что оба события произойдут вместе. Величина $P(E)$ представляет полную (безусловную) вероятность события E .

В рамках экспертных систем, заключения, которые являются результатами рассуждений и рекомендаций, могут рассматриваться как гипотезы, а посылки, которые являются фактами или свидетельствами, могут

рассматриваться как доказательства или дополнительная информация. Таким образом, прямой вероятностный вывод может применяться для определения вероятностей этих гипотез на основе имеющихся свидетельств.

Теорема Байеса устанавливает связь между априорной вероятностью гипотезы и вероятностью гипотезы при наличии свидетельств. Она позволяет обновлять и пересчитывать вероятности гипотез на основе новой информации. В контексте экспертных систем, априорные вероятности могут быть определены на основе предварительных знаний или опыта экспертов.

Применение прямого вероятностного вывода в экспертных системах позволяет эффективно учитывать неопределенность и изменчивость входных данных. Путем обновления вероятностей гипотез с учетом новых свидетельств, экспертная система может предоставлять более точные и надежные выводы и рекомендации.

Одним из способов использования прямого вероятностного вывода в экспертных системах является представление продукционных правил. Продукционные правила представляют собой условные высказывания вида "если-то", которые связывают посылки (факты или свидетельства) с заключениями (выводами). Применение прямого вероятностного вывода позволяет оценить вероятности заключений на основе достоверности посылок, а также учета априорных вероятностей гипотез.

Таким образом, применение прямого вероятностного вывода в экспертных системах способствует более точному и рациональному принятию решений, основанному на имеющихся данных и знаниях. Это позволяет снизить зависимость от экспертов-людей и повысить эффективность и надежность системы в решении специфических проблемных ситуаций.

2 ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ БАЙЕСОВСКИХ СЕТЕЙ В СИСТЕМЕ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

В данном разделе будет представлено подробное описание этапов, связанных с процессом внедрения Байесовских сетей в систему поддержки принятия решений. Для наглядности приведена Таблица 1, в которой отражены основные этапы, выбранные технологии для оптимизации и интеграции Байесовских сетей в систему поддержки принятия решений, а также затраченное время на выполнение прямого вероятностного вывода. Эта таблица позволяет наглядно проследить эволюцию процесса интеграции Байесовских сетей, начиная с первоначальной версии, описанной и реализованной в диссертации Змеева Дениса Олеговича [1], и прогрессирующей до последней версии с наивысшей оптимизацией и учетом усложняющих факторов.

Таблица 1- Этапы интеграции байесовских сетей в систему поддержки принятия решений.

Итерация	Технологии	Время	Описание
1	Rube	15 минут	Прямой вероятностный вывод вшит в структуру плагина для Redmine [5]. Основная структура сети храниться в XML файле. Вероятностный вектор подсчитан заранее.
2	Python	2 минуты	Прямой вероятностный вывод был переписан на ЯП python.
3	Neo4j	45 минут	Структура сети перенесена в графовую базу данных Neo4j [16]. API для взаимодействие с базой реализовано на Python.
4	-	4 часа	Внедрение в структуру сети «Рабочих продуктов», что привело к требованию динамически перестраивать байесовскую сеть.
5	-	30 минут	Отказ от хранения заранее подсчитанного вероятностного вектора. Переход на его подсчёт, используя метод Log-odds form.
6	Python + Golang	18 минут	Перенос расчётной части прямого вероятностного вывода на язык программирования Golang. На Python остаётся только сервер, который отвечает API приложения.
7	Golang	6 минут	Вся структура приложения была переписана на Golang, в том числе и API для взаимодействия с Neo4j.
8	-	2 минуты	Полный отказ от рекурсивного прохода по вершинам байесовской сети.
9	-	4 секунды	Многопоточный расчет вероятностей групп не зависимых вершин байесовской сети.

2.1 Обзор исходной системы

В прототипе системы принятия решений [1] уже была интегрирована работа с Байесовской сетью.

В данном разделе будет представлен обзор исходной системы, которая является отправной точкой для последующего улучшения. Описывая архитектуру старой системы, получится углубиться в её основные компоненты. Анализируя эту систему, получится лучше понять ее ограничения и слабые места, что в свою очередь поможет подобрать более эффективные подходы для оптимизации.

2.1.1 Хранение основной структуры Байесовской сети

В процессе разработки исследовательского прототипа была создана байесовская сеть, состоящая из 646 вершин. Максимальное количество вершин-родителей для одной вершины составляло 15.

В рамках данной задачи Байесовская сеть представляет собой модель, в которой каждая альфа из ядра Essense является вершиной, а связи между ними описывают отношения.

Показательный пример сети, созданной в данном прототипе, показан на Рисунке 3.

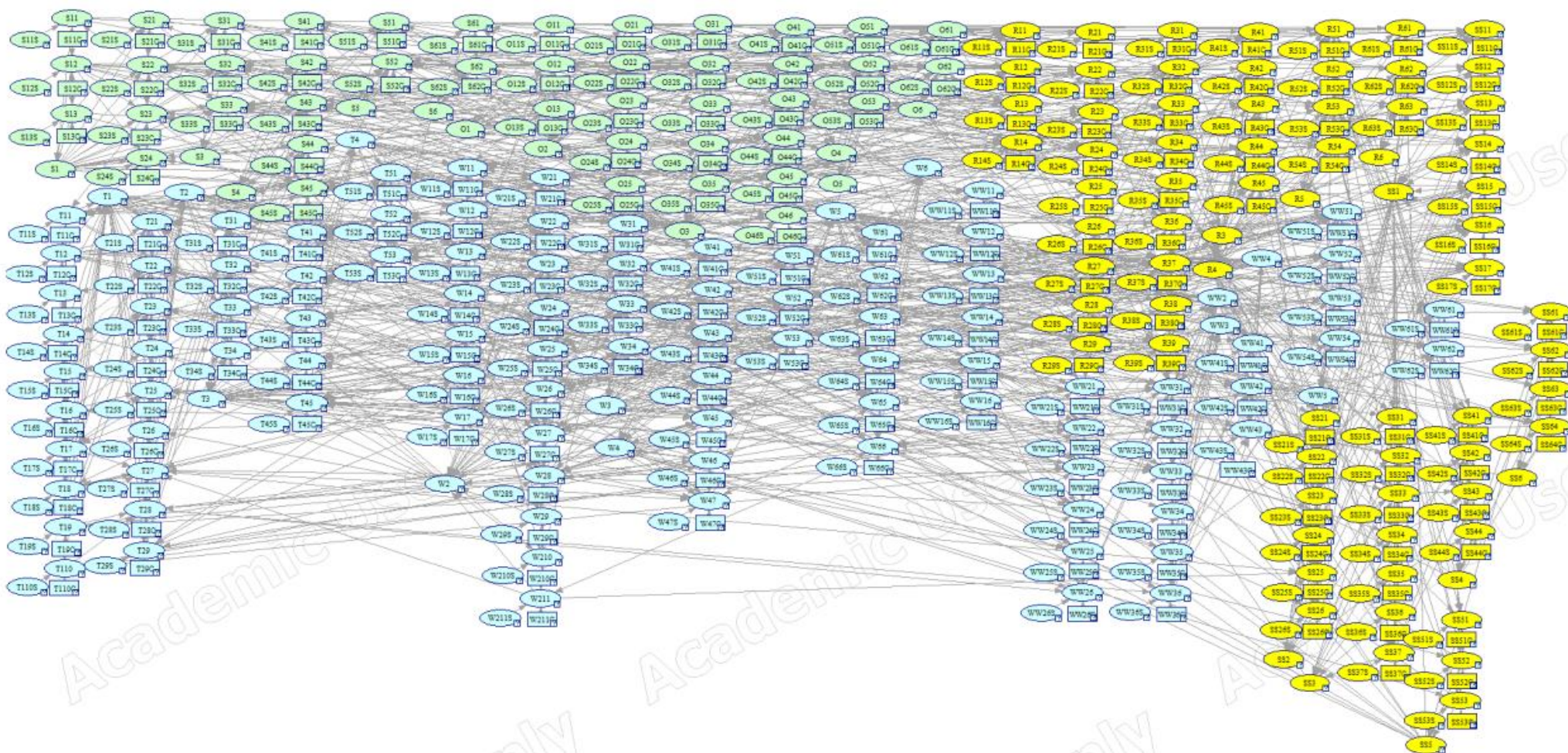


Рисунок 3 - Байесовская сеть для прототипа принятия решений

Данная Байесовская сеть, была построена с помощью прикладной программы GeNIe Modeler [4]. GeNIe Modeler является широко распространенным программным инструментом, который широко применяется для разработки и построения байесовских сетей. Он известен своим удобным интерфейсом, надежными функциями и возможностью выполнения вероятностных выводов. Однако существует одно заметное ограничение этого инструмента - невозможность построения сетей с вершинами, имеющими более чем 20 родителей. Это ограничение связано с особенностью хранения данных в GeNIe Modeler, который использует файлы XML-формата для хранения всей информации о байесовской сети. Такое ограничение может стать существенным недостатком.

Помимо хранения структуры Байесовской сети, для возможности произвести прямой вероятностный вывод, требуется также хранить и вероятностный вектор.

2.1.2 Хранение вероятностного вектора

Аналогично структуре байесовской сети, вероятностный вектор, представляющий собой данные о вероятностях для каждой вершины, хранился в формате XML-файлов. На Листинге 1 показан пример конкретной вершины

```
<cpt id="SS12">
  <state id="True" />
  <state id="False"/>
  <parents>SS12C R14 R22 SS11</parents>
  <probabilities>
    0.99 0.01 0.797767 0.202233 0.797767 0.202233
    0.605534 0.394466 0.797767 0.202233 0.605534 0.394466
    0.605534 0.394466 0.413301 0.586699 0.576699 0.423301
    0.384466 0.615534 0.384466 0.615534 0.192233 0.807767
    0.384466 0.615534 0.192233 0.807767
    0.192233 0.807767 0.0 1.0
  </probabilities>
</cpt>
```

Листинг 1 – пример хранения вершины Байесовской сети в файле XML-формата

Рассмотрим подробнее назначение тегов из примера файла.

Корневой элемент `src` имеет ключ `id`, в котором хранится идентификатор чекпоинта (идентификаторы чекпоинтов формируются по схеме «первая буква названия альфы, номер состояния, номер чекпоинта») и несколько вложенных элементов.

Вложенный элемент `state` показывает в каких состояниях может находиться данный чекпоинт, истинна или ложно.

Вложенный элемент `parents` содержит вектор родителей данного чекпоинта.

Вложенный элемент `probabilities` содержит вероятностный вектор, который показывает вероятность чекпоинта быть истинным или ложным при конкретной комбинации состояния родителей.

Состояние родителей на каждой позиции вектора можно представить в виде битовых комбинаций для n -го количество родителей, где 0 – состояние False, а 1 – состояние True.

Допустим у вершины есть 3 родителя, пример битовой комбинации для этой вершины представлен на Рисунке 4.

000,001,010,011,100,101,110,111

Рисунок 4 - Пример битовой комбинации родителей вершины

Порядок состояний родителей, соответствует их порядку во вложенном элементе `parents`.

Таблица 2 - Отношение комбинации вершин–родителей к вероятности вершины быть истинной или ложной

Комбинации	Вероятности
SS12C: 1, R14: 1, R22: 1, SS11: 1	True: 0.99, False: 0.01
SS12C: 1, R14: 1, R22: 1, SS11: 0	True: 0.797767, False: 0.202233
SS12C: 1, R14: 1, R22: 0, SS11: 1	True: 0.797767, False: 0.202233
SS12C: 1, R14: 1, R22: 0, SS11: 0	True: 0.605534, False: 0.394466
SS12C: 1, R14: 0, R22: 1, SS11: 1	True: 0.797767, False: 0.202233
SS12C: 1, R14: 0, R22: 1, SS11: 0	True: 0.605534, False: 0.394466
SS12C: 1, R14: 0, R22: 0, SS11: 1	True: 0.605534, False: 0.394466
SS12C: 1, R14: 0, R22: 0, SS11: 0	True: 0.413301, False: 0.586699
SS12C: 0, R14: 1, R22: 1, SS11: 1	True: 0.576699, False: 0.231068
SS12C: 0, R14: 1, R22: 1, SS11: 0	True: 0.384466, False: 0.615534
SS12C: 0, R14: 1, R22: 0, SS11: 1	True: 0.384466, False: 0.615534
SS12C: 0, R14: 1, R22: 0, SS11: 0	True: 0.192233, False: 0.807767
SS12C: 0, R14: 0, R22: 1, SS11: 1	True: 0.384466, False: 0.615534
SS12C: 0, R14: 0, R22: 1, SS11: 0	True: 0.192233, False: 0.807767
SS12C: 0, R14: 0, R22: 0, SS11: 1	True: 0.192233, False: 0.807767
SS12C: 0, R14: 0, R22: 0, SS11: 0	True: 0.0, False: 1.0

Хранение данных в файлах накладывает ограничение на число родителей из-за большого размера файла, поскольку размер вероятностного вектора зависит от числа родителей в соотношении 2^n , где n – число родителей.

Такое ограничение накладывается конкретно на данную задачу, поскольку все вершины в данной структуре Байесовской сети, могут иметь только два состояния – истинна или ложь. С увеличением числа состояний размер вероятностного вектора будет иметь отношение X^n , где x –число доступных состояний, а n – число родителей.

Когда вопросы с хранением структуры Байесовской сети и вероятностного вектора решены, можно перейти к выполнению прямого вероятностного вывода.

2.1.3 Прямой вероятностей вывод в прототипе системы принятия решений

Прямой вероятностный вывод представляет собой метод получения вероятностей каждой вершины быть истинной, алгоритм для получения данных вероятностей показан на Рисунке 5.

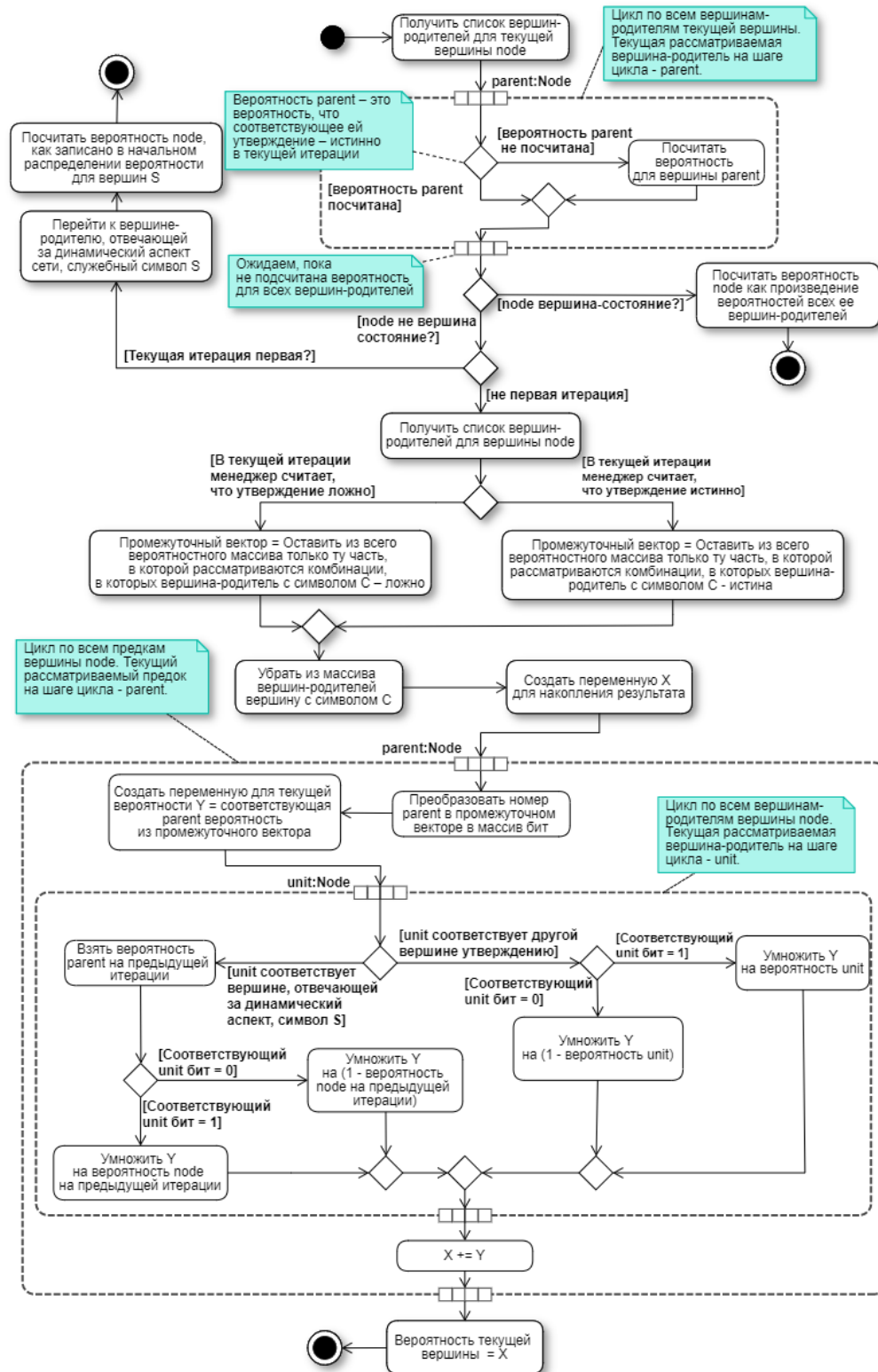


Рисунок 5 – Базовый алгоритм прямого вероятностного вывода

С использованием вышеописанного алгоритма, можно вычислить вероятность истинности всех утверждений в текущей итерации с учетом мнения менеджера о них. После проведения расчетов, чтобы выявить ложноположительные ошибки менеджера, необходимо выполнить два простых шага:

1. Выбрать все утверждения, которые менеджер считает истинными.
В рамках данной процедуры мы учитываем мнение менеджера и выбираем все утверждения, которые он признал верными на текущий момент.
2. Из этого набора утверждений выбрать те, вероятность истинности которых ниже порогового значения. В данном случае мы рассматриваем два возможных пороговых значения: строгий критерий достоверности равный 0,95 и допустимый порог для текущего проекта, например, равный 0,5. Утверждения, вероятность истинности которых ниже выбранного порога, будут считаться ложноположительными ошибками менеджера.

Таким образом, путем применения этих двух шагов возможно выделить утверждения, которые менеджер считает истинными, но имеют низкую вероятность истинности, что позволяет выявить возможные ложноположительные ошибки.

Данный алгоритм был реализован как часть плагина для Redmine и написан на языке программирования Ruby. Время выполнения данного алгоритма с учётом используемых технологий для хранения структуры Байесовской сети и вероятностного вектора в среднем равнялось пятнадцати минутам, что является неприемлемо длительным временем для постоянного использования. Это вызвало серьезные проблемы, поскольку ожидалось, что система поддержки принятия решений будет использоваться в реальном времени.

Кроме того, хранение всех данных в XML-файлах привело к ограничениям в построении Байесовской сети.

В свете вышеупомянутых проблем, первым шагом к их решению состояло в поиске готового инструмента, способного эффективно работать с Байесовскими сетями. Целью было найти средство, которое бы устраняло недостатки, такие как длительное время выполнения прямого вероятностного вывода и ограничения на построение сети, связанные с хранением данных в XML-формате.

2.2 Сравнительный анализ инструментов по работе с Байесовскими сетями

С учетом специфики рассматриваемой задачи, необходимо определить требования к инструменту, который будет использоваться для работы с Байесовскими сетями. Удовлетворение этих требований критически важно для обеспечения эффективной и надежной функциональности системы поддержки принятия решений.

1. Поддержка байесовской сети с большим количеством вершин (более 1000). Сеть, используемая в диссертации [1] содержит 646 вершины, однако дальнейшие планы по использованию больших возможностей стандарта OMG Essence подразумевают существенный рост числа вершин в сети.
2. Интеграция с внешней системой. С целью использования данного подхода в различных системах.
3. Поддержка работы с большим количеством у конкретной вершины сети вершин-родителей (более 20 вершин-родителей) с адекватным временем выполнения прямого вероятностного вывода.
4. Возможность импорта готовой сети из инструмента его построения.
5. Наличие бесплатной версии. Поскольку данный проект носит исследовательский характер.
6. Возможность работы с инструментом в рамках отсутствия обучающей выборки.

В Таблице 3 приведены результаты анализа наиболее популярных на момент написания ВКР инструментов.

Таблица 3 – Анализ инструментов по работе с байесовскими сетями

Критерий	Agena Risk [6]	Bayes Fusion [7]	Netica [8]	Bayesia Lab [9]	Bayes AML [10]	MSBNx [11]	Bayes Server [12]
1	+	+	+	+	+	+	+
2	+	–	–	+	+	–	+
3	–	–	–	–	–	–	–
4	+	+	+	+	+	+	+
5	–	+	–	+	–	+	–
6	+	–	–	+	+	+	+

Проведенный анализ указывает на необходимость разработки специализированного инструмента, способного удовлетворить ранее описанные требования. Это обусловлено выявленными проблемами, связанными с обработкой большого количества вершин-родителей у конкретной вершины байесовской сети и необходимостью достижения оптимального для постоянного использования системы, времени выполнения прямого вероятностного вывода.

В свете указанных требований и необходимости создания специализированного инструмента, в следующем разделе будет представлено описание этапов разработки данного инструмента.

2.3 Разработка системы для работы с Байесовскими сетями

На основе анализа предыдущих разработок было принято решение о необходимости разработки новой системы, предназначенной для работы с Байесовскими сетями, которая полностью удовлетворяла бы всем требованиям для интеграции в систему принятия решений. Для реализации новой системы возникла необходимость решить следующие задачи:

1. Продумать и реализовать систему хранения основной структуры Байесовской сети, а также получение и хранение вероятностного вектора. Одной из ключевых задач было определение эффективного механизма хранения структуры сети, чтобы обеспечить быстрый доступ к данным и эффективную обработку информации. Кроме того, необходимо было разработать механизмы для получения и сохранения вероятностного вектора, который представляет собой данные о вероятностях для каждой вершины в сети.
2. Реализовать и оптимизировать прямой вероятностный вывод. Это важная задача, которая требует разработки алгоритмов и методов для эффективного выполнения вероятностных вычислений в Байесовских сетях. Прямой вероятностный вывод позволяет определить вероятности для различных событий и их взаимосвязей на основе имеющихся данных. Оптимизация этого процесса имеет важное значение, поскольку может существенно повлиять на скорость и точность анализа данных.

2.3.1 Хранение структуры Байесовской сети

В контексте хранения структуры байесовской сети, основной проблемой состояла в невозможности осуществления быстрого доступа к необходимым значениям. В связи с этим, возникла необходимость в поиске решения, которое позволило бы эффективно и быстро отвечать на запросы.

Для решения данной проблемы, требовалось найти оптимальный механизм или подход, который обеспечивал бы эффективное извлечение требуемых значений из структуры Байесовской сети.

Важными критериями при выборе решения были скорость получения ответов на запросы и общая производительность системы хранения структуры Байесовской сети. Оптимальное решение должно было обеспечивать высокую эффективность и минимизацию времени, необходимого для обработки запросов и получения нужных значений из структуры.

Для решения указанной проблемы было принято решение исследовать возможность использования графовых систем управления базами данных (СУБД), поскольку Байесовская сеть представляет собой направленный ациклический граф. Графовые СУБД выявляются как значительно более производительные по сравнению с другими типами СУБД в задачах, связанных с созданием, сохранением и изменением схемы базы данных, которые имеют естественную графовую структуру данных [15].

Выбор графовых СУБД обусловлен их способностью эффективно обрабатывать и оперировать с графовыми структурами данных, что в нашем случае соответствует структуре Байесовской сети. Такие СУБД предлагают набор оптимизированных алгоритмов и механизмов, которые позволяют выполнять операции создания, сохранения и модификации графовых структур базы данных с высокой производительностью.

В качестве СУБД был выбран инструмент Neo4j Graph Database [16], поскольку он использует модель ориентированного графа, имеет свой собственный язык запросов Cypher, предоставляет удобное API для взаимодействия с БД, имеет инструменты визуализации, обширную документацию, а также доступен для бесплатного использования.

Каждая вершина байесовской сети транслируется в вершину графа в БД, зависимости между вершинами сети, аналогично, становятся ребрами между вершинами графа. Пример Байесовской сети для поставленной задачи представлен на Рисунке 6.

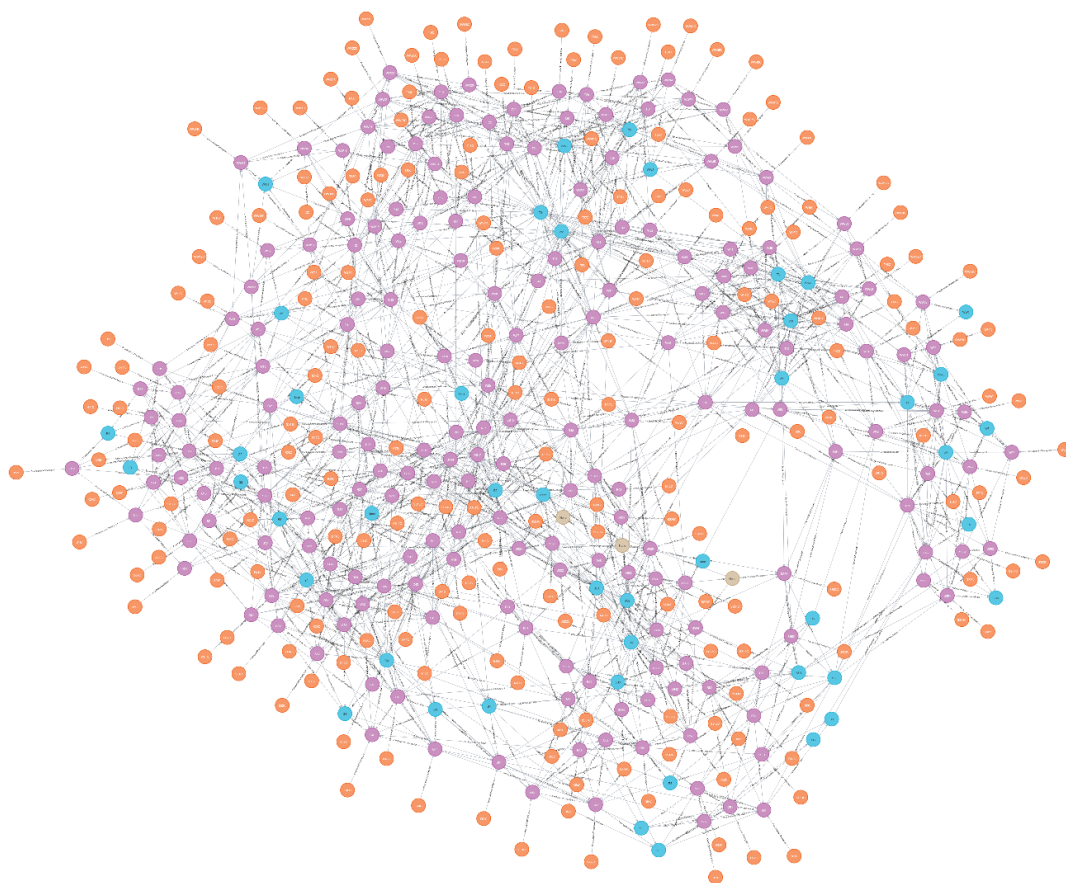


Рисунок 6 - Байесовская сеть из Neo4j

Каждая вершина графа в БД содержит следующую информацию:

- Идентификатор;
- Имя вершины.

Пример хранения аргументов вершины, представлен на Рисунке 7.

checkpoint		
<id>	89	
guid	217ad4bf-35be-4551-85c4-98cad128be24	
name	S44	

Рисунок 7 - Аргументы у вершины в Neo4j

2.3.2 Внедрение рабочих продуктов в систему принятия решений

В рамках исследования [17] было произведено внедрение рабочих продуктов в систему, которые представляют собой артефакты, служащие подтверждением или опровержением утверждения о вершине, принимающей истинное значение. Пример таких артефактов показан на Рисунке 8.

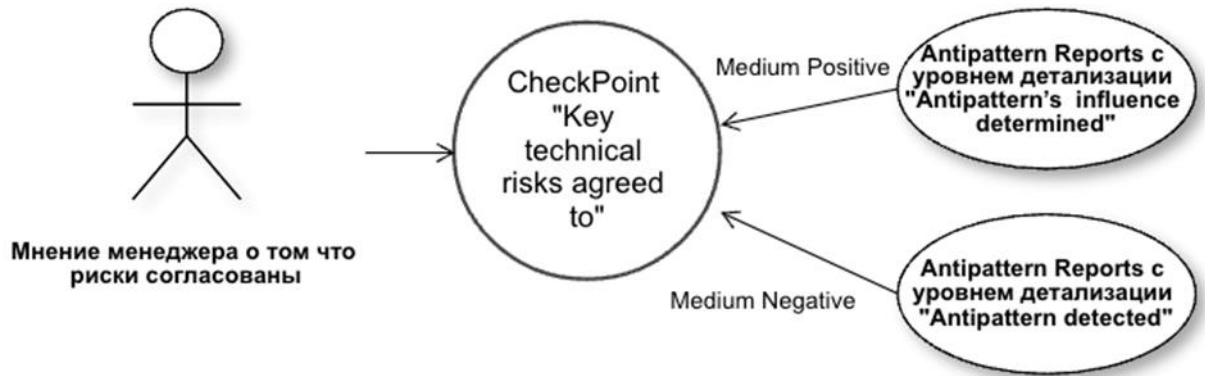


Рисунок 8 – Пример влияние артефактов на вершину

Поскольку такие артефакты могут добавляться к вершинам на любом этапе разработки, возникла потребность в динамическом обновлении Байесовской сети.

Внедрение данного нововведения привело к необходимости пересчета вероятностного вектора для всех вершин при каждом запуске алгоритма, перед тем как приступить к прямому вероятностному выводу. Такой подход привел к существенному увеличению времени выполнения прямого вероятностного вывода.

В связи с динамическим характером добавления артефактов и требованием обновления Байесовской сети, необходимость пересчета вероятностного вектора стала значительной нагрузкой для системы. При каждом запуске алгоритма происходило полное обновление вероятностного вектора для всех вершин, что приводило к заметному увеличению времени, затрачиваемого на выполнение прямой вероятностного вывода.

Таким образом, интеграция рабочих продуктов и динамическое обновление байесовской сети вызвали существенное увеличение времени

выполнения прямого вероятностного вывода. Это следует учитывать при разработке и оптимизации системы, чтобы минимизировать время, необходимое для проведения прямого вероятностного вывода при динамических изменениях в структуре сети.

В попытке решить данную проблему, было принято решение считать вектор вероятностей во время выполнения прямого вероятностного вывода.

2.3.3 Получение вероятностного вектора

Для оптимизации прямого вероятностного вывода после внедрения рабочих продуктов, которые потребовали динамически обновлять структуру Байесовской сети и при каждом запуске пересчитывать вероятностные вектора для всех вершин, надо было найти способ, который не хранит значения вероятностей, а рассчитывает их вместе с выполнением прямого вероятностного вывода. Решением стал метод Log-odds form, который позволит рассчитывать значения вероятностного вектора на лету. Об этом методе подробно рассказывают в статье [18].

Описание метода:

Форма коэффициентов правила Байеса гласит, что предыдущие коэффициенты, умноженные на отношение правдоподобия, равны последующим коэффициентам. Мы можем взять логарифм обеих сторон этого уравнения, получив эквивалентное уравнение, в котором используется сложение вместо умножения.

Позволяя H_i и H_j обозначать гипотезы, а e обозначать доказательства, логарифмическая форма правила Байеса представлена в Формуле 6:

$$\log \left(\frac{P(H_i|e)}{P(H_j|e)} \right) = \log \left(\frac{P(H_i)}{P(H_j)} \right) + \log \left(\frac{P(e|H_i)}{P(e|H_j)} \right) \quad (6)$$

Это может быть эффективно с числовой точки зрения, когда вы выполняете множество обновлений одно за другим. Это как раз то, что происходит в нашем проекте.

Представим, что гипотеза H так же вероятно, как $\neg H$. Тогда мы получаем пять отдельных независимых обновлений в пользу H . Наши шансы для H варьируются от (1:1) до (32:1). Таким образом вероятности будут от $\frac{1}{2} = 50\%$ до $\frac{32}{33} \approx 97\%$. Графическое представление данных изменений вероятностей показана на Рисунке 9.

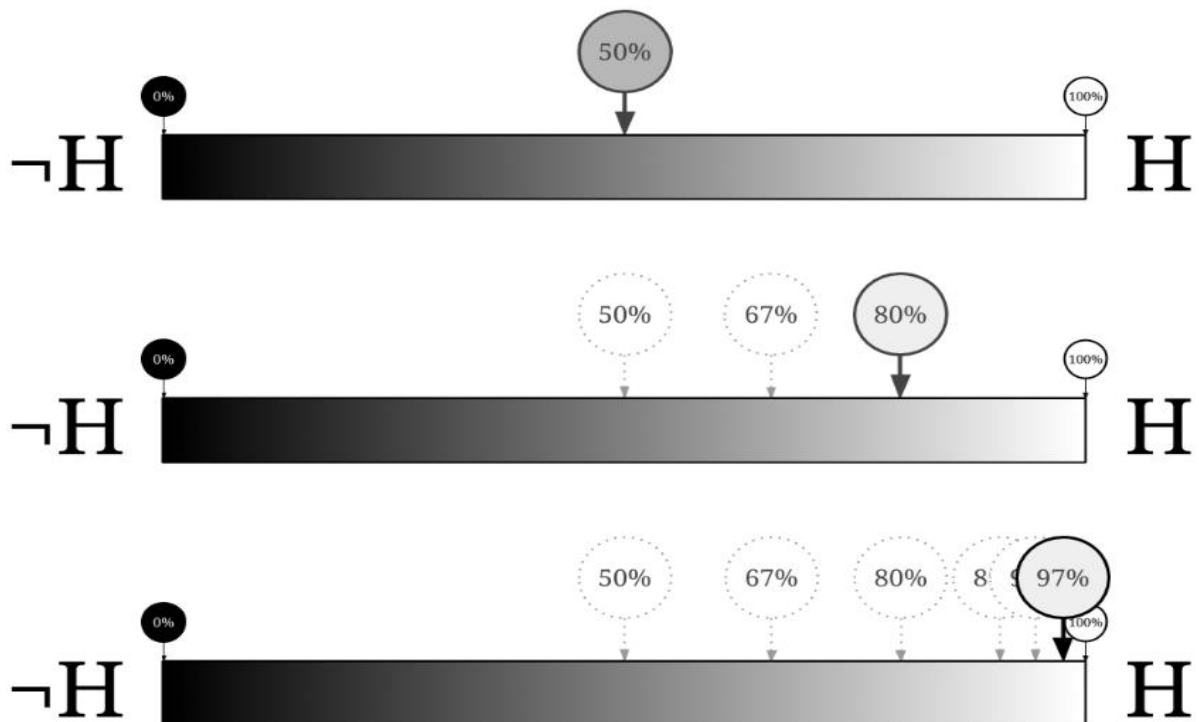


Рисунок 9 - Линии Байса в процентном соотношении

Мы наблюдаем ситуацию, что вероятность приближается к 1, но никогда не достигает её. Если мы преобразуем вероятности в логарифмические коэффициенты, то картина получится намного приятнее. Вероятность 50% становится равно 0 бит в пользу утверждения, а каждое независимое наблюдение в пользу утверждение сдвигает вероятность на одну единицу вдоль линии, данный пример показан на Рисунке 10.

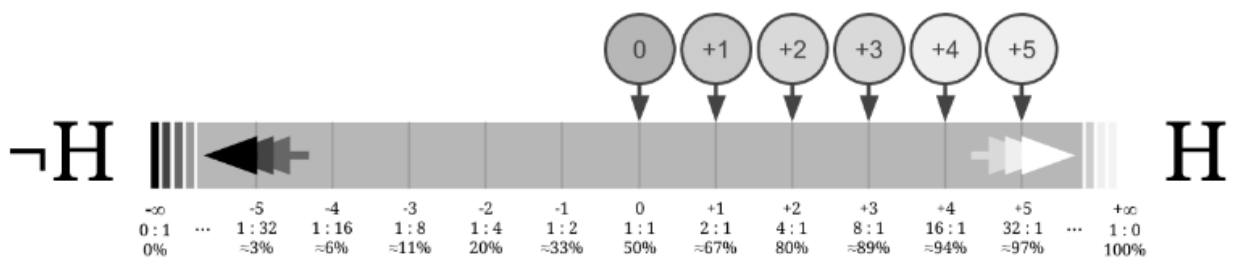


Рисунок 10 - Линия Байса в байтовом представлении

Рассмотрим, как данный метод используется для получения вероятности вершины быть истинной в зависимости от комбинации родителей.

Для получения вероятности используется формула полной вероятности Формула 7.

$$P(A) = \sum_{i=1}^n P(H_i) P(A|H_i) \quad (7)$$

Рассмотрим вершину с минимальным числом родителей, такой вершиной является вершина S11. Подграф с данной вершиной показан на Рисунке 11.



Рисунок 11 - Подграф родителей для вершины S11

У вершины S11 имеется 2 предка – другая вершина W14 и вершина S11C представляющая мнение менеджера о состоянии вершины S11.

За количество байт в пользу одного из двух утверждений (вершина истина или ложна) возьмем числовое представление степени влияния родителя на вершину. На данный момент в задаче присутствует 3 вида степеней влияния:

- 1) низкая(week) – числовое представление 1;
- 2) средняя(medium) – числовое представление 2;
- 3) высокая(strong) – числовое представление 5.

Для получения соотношения используем выражение 2^x , где x – числовое представление степени влияния родителя. Представим данное выражение в виде функции $VMS(x) = 2^{\text{степень влияния } X}$, где x – вершина. Тогда итоговая полная вероятность примет вид, представленный в Формуле 7.

$$\frac{VMS(W14)VMS(S11C)P(W14 = 1)}{VMS(W14)VMS(S11C) + 1} + \frac{VMS(W14)VMS(S11C)(1 - P(W14 = 1))}{VMS(W14) + VMS(S11C)} \quad (8)$$

После того как был проанализирован метод получения вероятностного вектора, надо было интегрировать его в алгоритм прямого вероятностного вывода.

2.3.4 Изменение прямого вероятностного вывод

После внедрения рабочих продуктов и перехода к расчету вероятностного вектора, алгоритм прямого вероятностного вывода требовал значительных доработок.

На Рисунке 12 представлен алгоритм прямого вероятностного вывода в общем виде для определенной вершины. В рамках данного алгоритма происходит итерационная обработка каждой вершины в байесовской сети. Алгоритм демонстрирует рекурсивную природу, где происходит глубокий обход вершин-родителей до достижения вершины, не имеющей родителей.

При вычислении вероятности для текущей рассматриваемой вершины применяется метод, зависящий от ее типа. Кроме того, значение текущей итерации также имеет значение, поскольку для вершин, не являющихся вершинами-состояниями альфа Ядра, вероятность на первой итерации равна нулю.

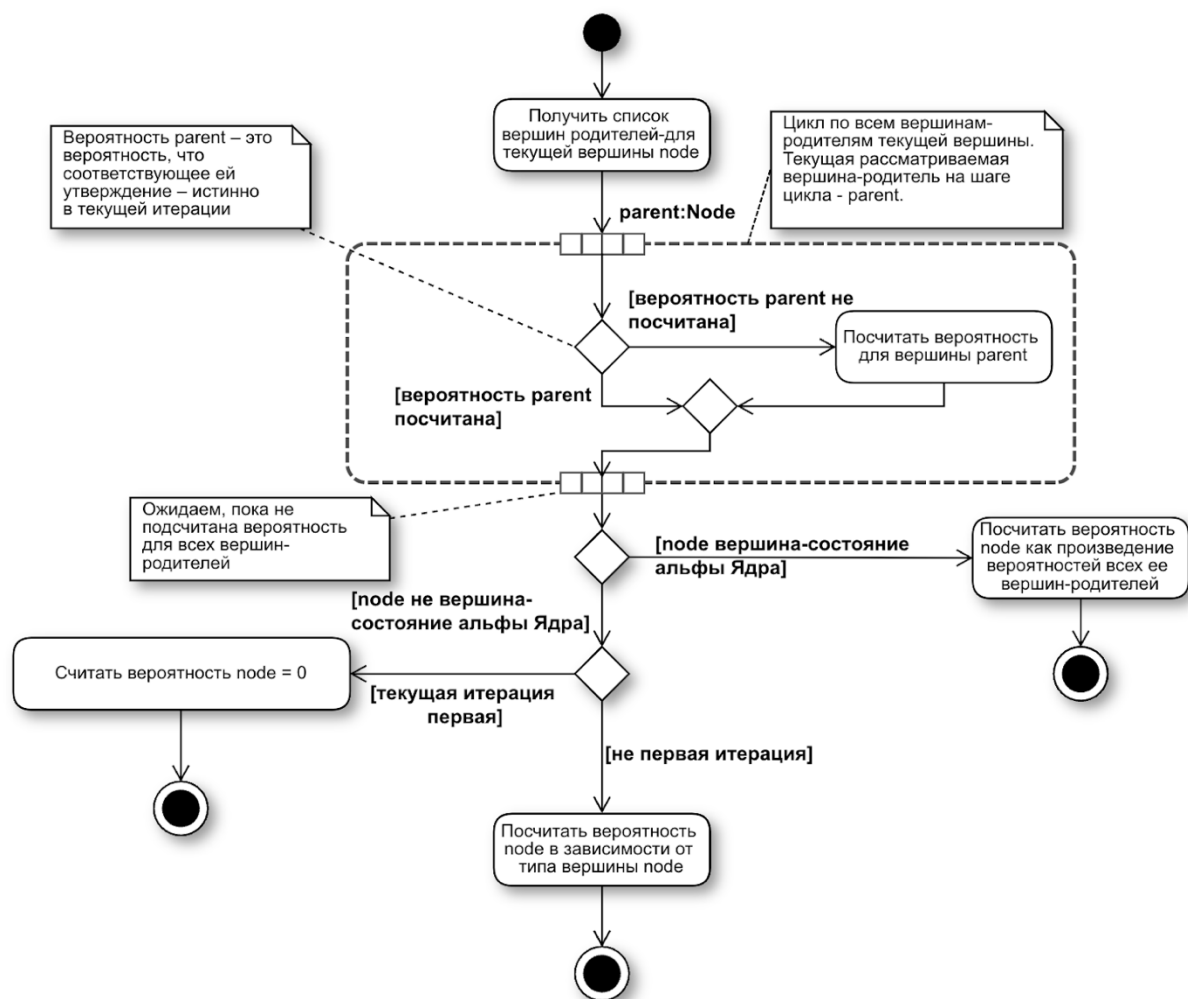


Рисунок 12 - Общий вид алгоритма прямого вероятностного вывода

Декомпозиция шага “Рассчитать вероятность node в зависимости от типа вершины” представлена на рисунке 13.

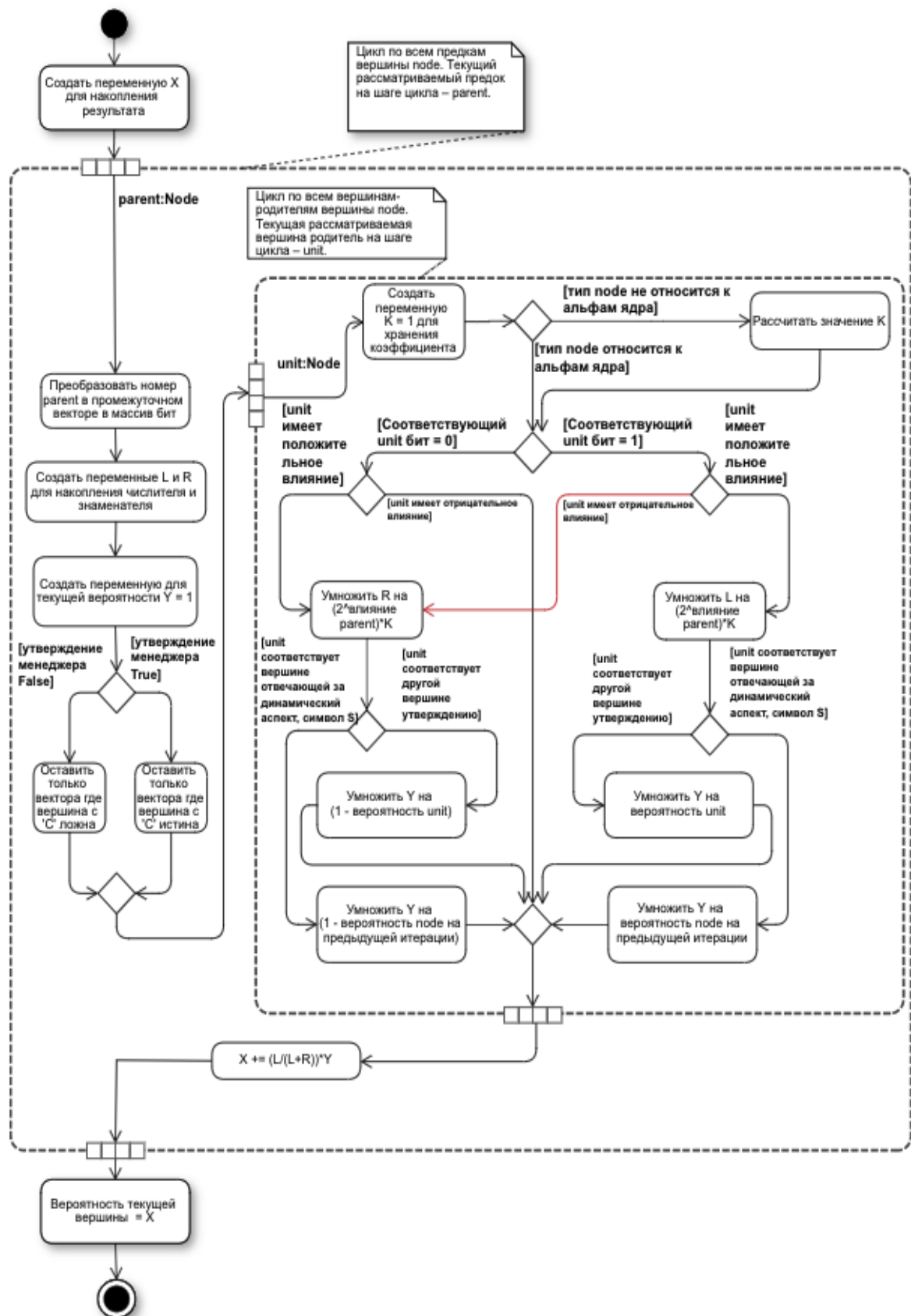


Рисунок 13 - Декомпозиция шага “Рассчитать вероятность node в зависимости от типа вершины”

С учётом рабочих продуктов и логарифмического правила для расчёта вероятностного вектора, алгоритм расчета принимает вид, показанный на Рисунке 14.

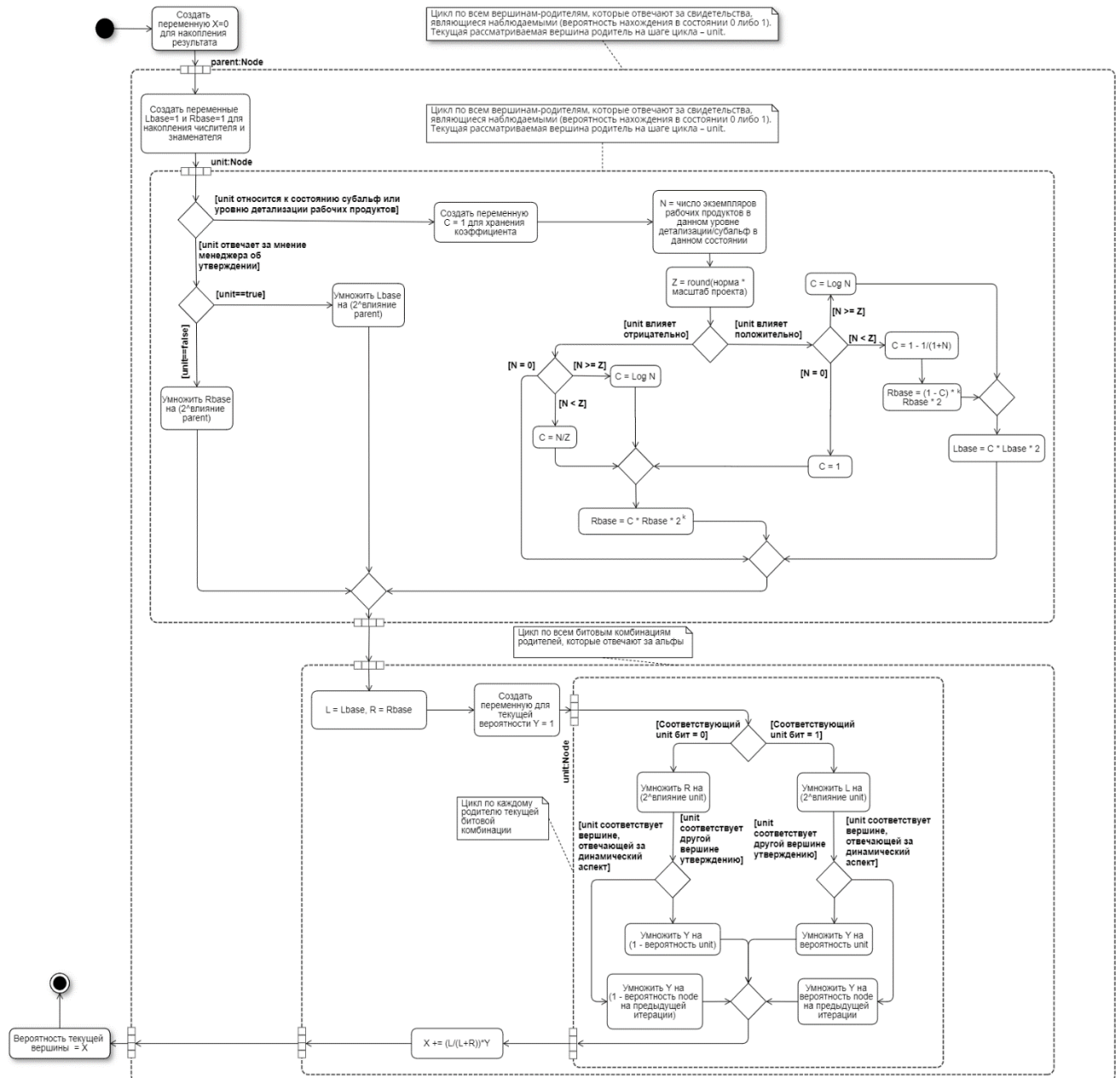


Рисунок 14 – Алгоритм прямого вероятностного вывода с учетом рабочих продуктов

2.4 Оптимизации прямого вероятностного вывода в Байесовских сетях

В рамках оптимизации алгоритма прямого вероятностного вывода требовалось решить ещё множество задач, чтобы довести время выполнения до минимально возможной границы.

Все дальнейшие тесты были проведены на оборудовании (Intel(R) Xeon(R) CPU X5675 @ 3.07GHz, 100 Gb RAM DDR2)

2.4.1 Сравнение языков программирования

На протяжении значительного времени наиболее распространенные языки программирования успешно проявили свои возможности и определили свои сферы применения.

Тем не менее, для решения сложных задач, требующих обработки больших объемов вычислений, возникала необходимость провести сравнительный анализ нескольких популярных языков программирования, с целью определить язык, который обладает высокой производительностью и способностью параллелизации вычислений.

В качестве тестовой задачи для вычислений рассмотрим функцию "is prime", которая предназначена для определения простоты числа. Мы рассмотрим базовый алгоритм проверки на простоту, который включает следующие шаги:

- Проверка, является ли число нечётным;
- Проверка, не делится ли число на меньшие числа в диапазоне от 3 до квадратного корня из данного числа.

Пример алгоритма показан на Рисунке 15.

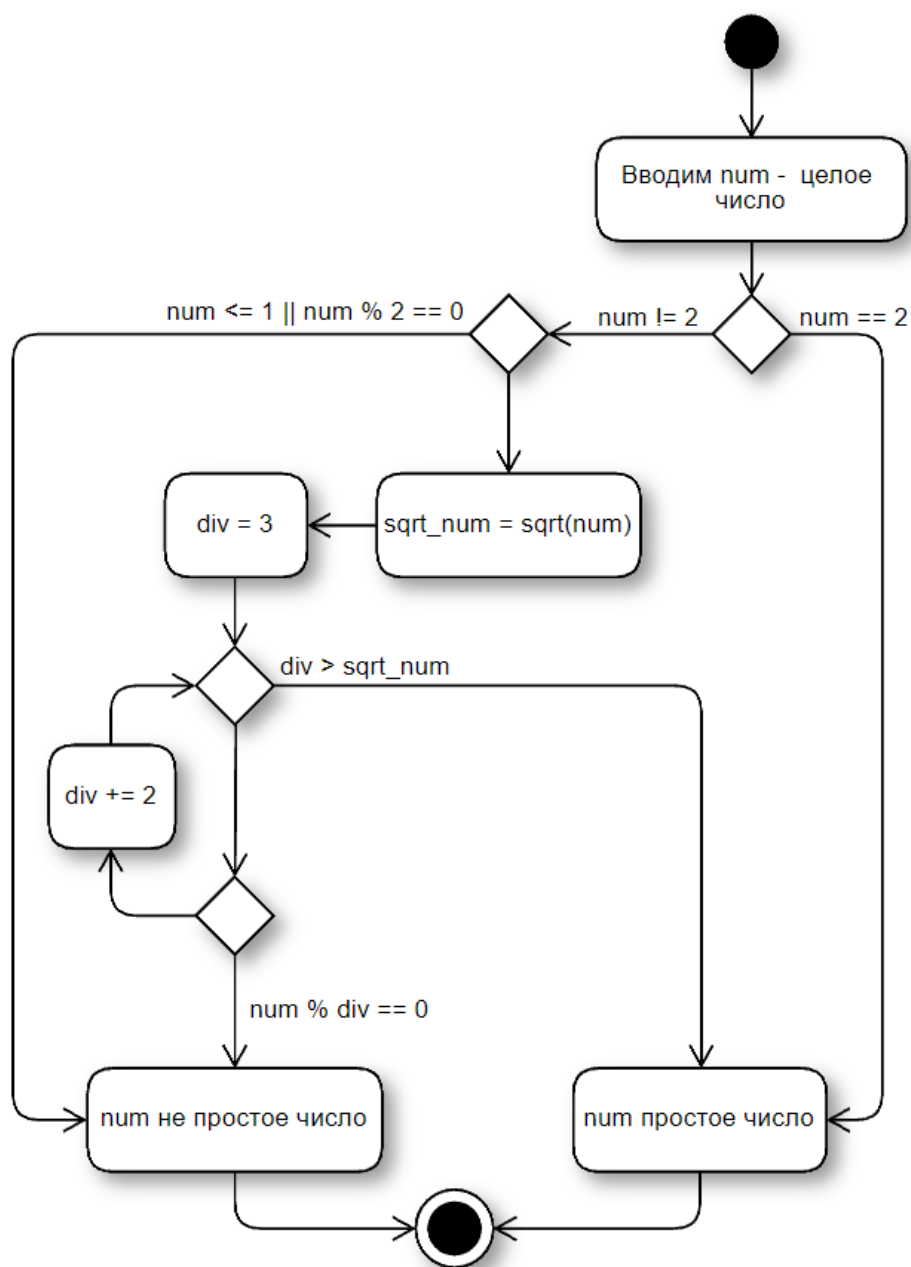


Рисунок 15 - алгоритм «is prime»

Для выполнения вычислений необходимо сформировать последовательность простых чисел, которые не превышают заданного максимального значения, равного 10 000 000.

В ходе проведения тестирования рассматривались следующие языки программирования: C++, Golang, Java, PHP8 и Python.

Результат тестирования показан на Рисунке 16.

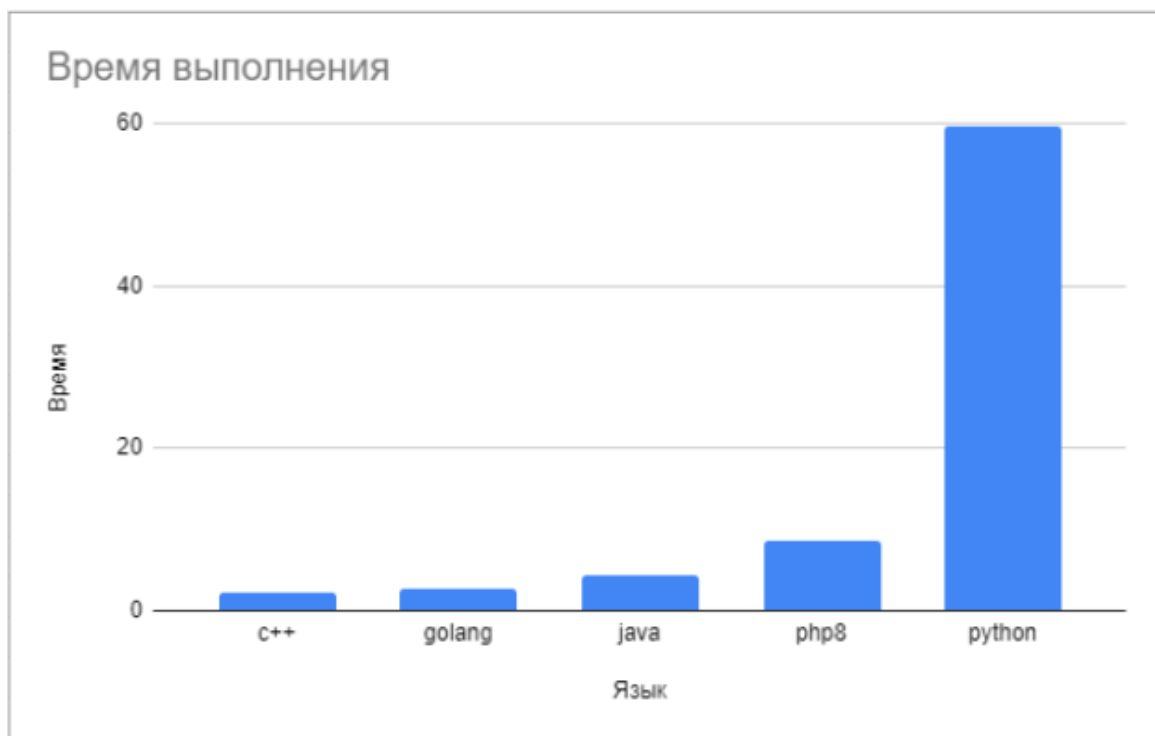


Рисунок 16– Гистограмма времени выполнения алгоритма «is prime»

В результате проведенного тестирования были отобраны два наиболее перспективных кандидата для дальнейшего рассмотрения: C++ и Golang. Последующая фаза тестирования включала параллелизацию алгоритма "is prime". Результаты данного теста представлены на Рисунке 17.

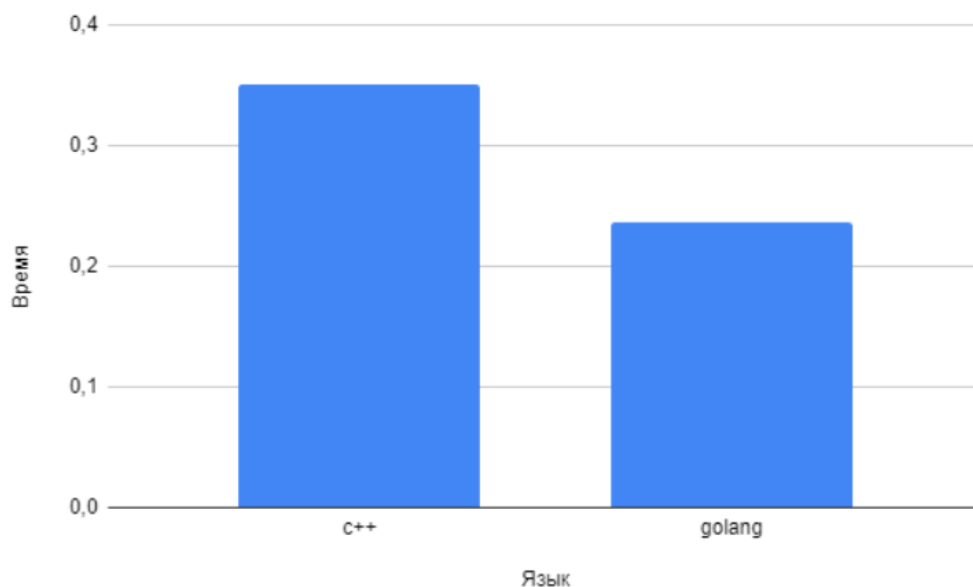


Рисунок 17- Гистограмма времени выполнения алгоритма «is prime» в многопоточном режиме

Результаты тестирования свидетельствуют о том, что при использовании многопоточного программирования Golang демонстрирует более высокую эффективность по сравнению с C++. Это объясняется использованием внутреннего механизма в Golang, известного как "горутин" (goroutines) [19] - легковесных потоков, которые реализуют конкурентное программирование в языке Golang. Горутин являются "легковесными", поскольку их выполнение контролируется средой выполнения (runtime) языка, а не операционной системой. Это позволяет сократить затраты на переключение контекста и уменьшить расход памяти по сравнению с использованием потоков операционной системы.

2.4.1.1 Горутин и как они работают

В основе функционирования горутин лежит Планировщик горутин. Этот механизм является посредником для управления объектами G (горутин Golang), M (потоки операционной системы) и P (планировщики операционной системы).

- Объект "G" представляет собой простую горутину в языке Golang;
- Объект "M" представляет собой поток операционной системы, который может быть активным или находиться в состоянии бездействия;
- Объект "P" можно рассматривать как "центральный процессор" в планировщике операционной системы; он предоставляет необходимые ресурсы для выполнения нашего кода на Go, такие как планировщик или состояние распределителя памяти.

Основной целью планировщика является установление соответствия между каждым кодом G, который требуется выполнить, и соответствующим

местом выполнения М, а также доступными правами и ресурсами Р, необходимыми для выполнения данного кода.

По завершении выполнения кода в объекте М, он возвращает свой набор прав и ресурсов Р в пул доступных Р. Для продолжения выполнения кода на языке Go необходимо повторно получить доступ к этим ресурсам. Аналогично, при завершении горутин, объект G возвращается в пул доступных G и может быть повторно использован для выполнения другой горутин.

При запуске горутин ей выделяется начальный размер стека в минимальном объеме, составляющем 2 килобайта, который может динамически увеличиваться и уменьшаться по мере необходимости, без риска исчерпания стекового пространства. В статье Дэйва Чейни [14] подробно описывается работа данного механизма.

Хотя минимальный размер стека составляет 2048 байт, рантайм Golang также ограничивает максимальный размер стека горутин; это значение зависит от архитектуры и составляет 1 ГБ для 64-разрядных систем и 250 МБ для 32-разрядных систем.

Опираясь на принципы функционирования горутин, можно составить таблицу сравнения горутин и потоков операционной системы, которая представлена в Таблице 3.

Таблица 3 – Сравнение горутин и стандартных потоков ОС

Потоки	Горутин
Потоки ОС управляются ядром ОС	Горутин управляются “рантаймом” Go
Потоки ОС имеют фиксированный размер в 1-2МВ	Горутин обычно имеют размер стэка 2КВ
Размер стэка определяется во время компиляции и не может увеличиться	Размер стэка определяется во время рантайма
У потоков нету простого способа коммуникации между собой. Такая коммуникация имеет большую задержку	Горутин используют “каналы” для быстрого общения между собой с маленькой задержкой

2.4.2 Предобработка данных

Для дальнейшей оптимизации алгоритма прямого вероятностного вывода требовалась предварительная обработка данных для возможности производить многопоточные вычисления.

Первым шагом было преобразование вершин в очередь, при котором обработка производится, начиная с вершин без предков и заканчивая самыми "богатыми" по количеству предков вершинами. Подробности алгоритма построения очереди приведены на Рисунке 18.

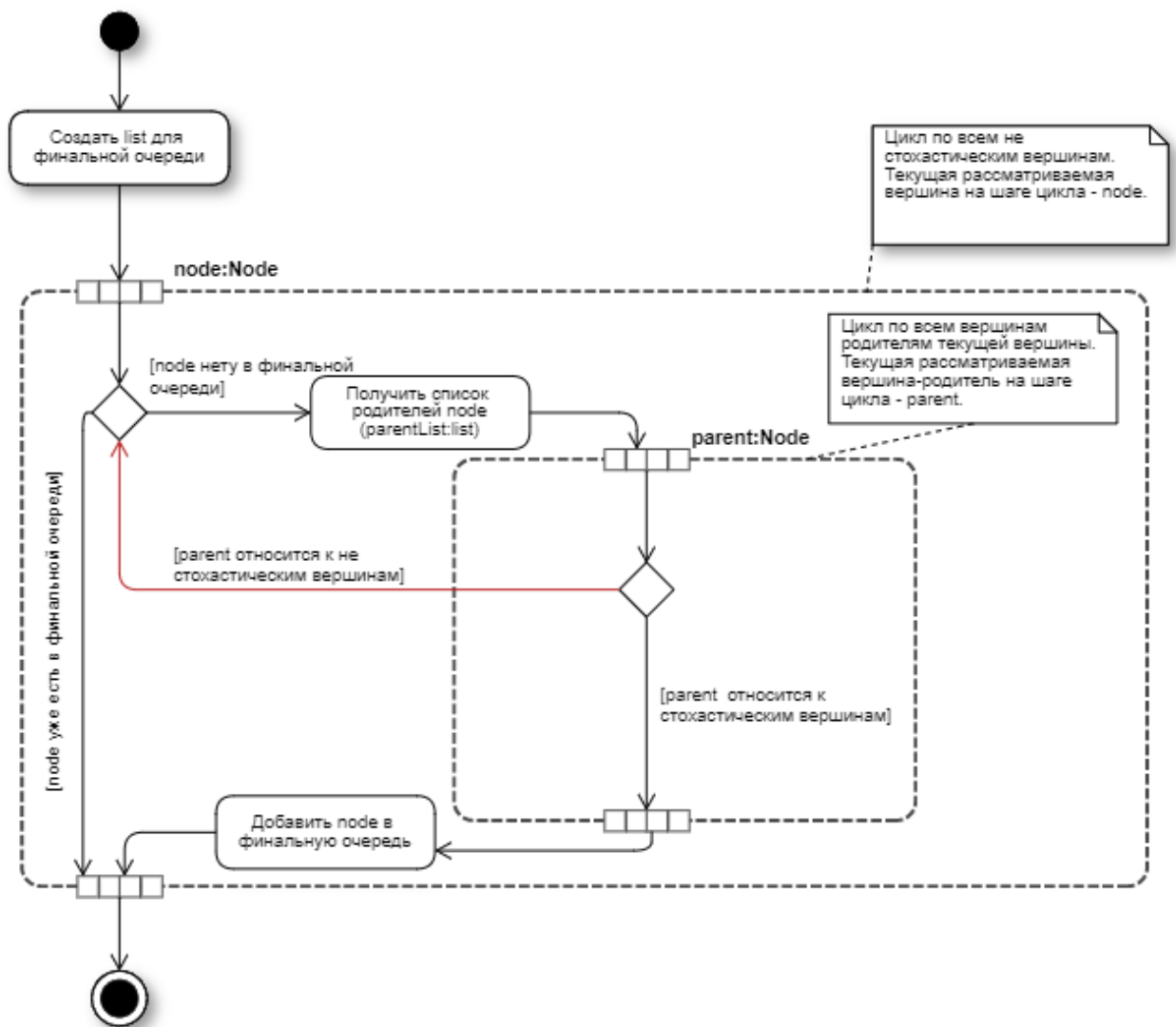


Рисунок 18 – Алгоритм построения очереди независимых вершин

Предварительная обработка данных позволяет осуществлять обход всех вершин сети в упорядоченном порядке, что позволяет избежать рекурсии и устранить необходимость в подсчете предыдущих итераций. Это приводит к

ускорению выполнения алгоритма прямого вероятностного вывода и сокращению количества вершин в сети на треть, поскольку все вершины, отвечающие за подсчет предыдущих итераций, больше не используются.

Для того чтобы перейти к распараллеливанию вычислений требуется разбить очередь на группы независимых вершин, данная процедура реализуется, используя алгоритм поиска в ширину. Диаграмма построения очереди показана на Рисунке 19.

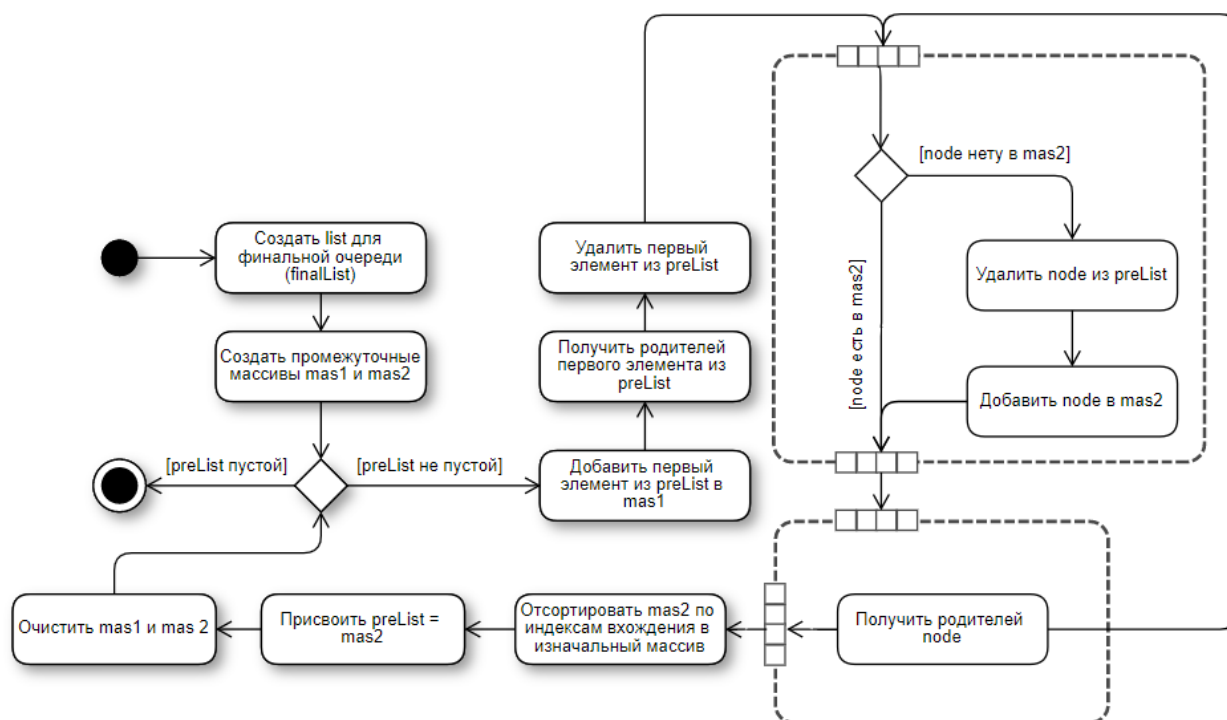


Рисунок 19 – Алгоритм построения списка из групп независимых вершин

2.4.3 Параллелизация вычислений

Выполнив подготовку данных мы можем перейти к параллелизации вычислений, алгоритм параллелизации показан на Рисунке 20.

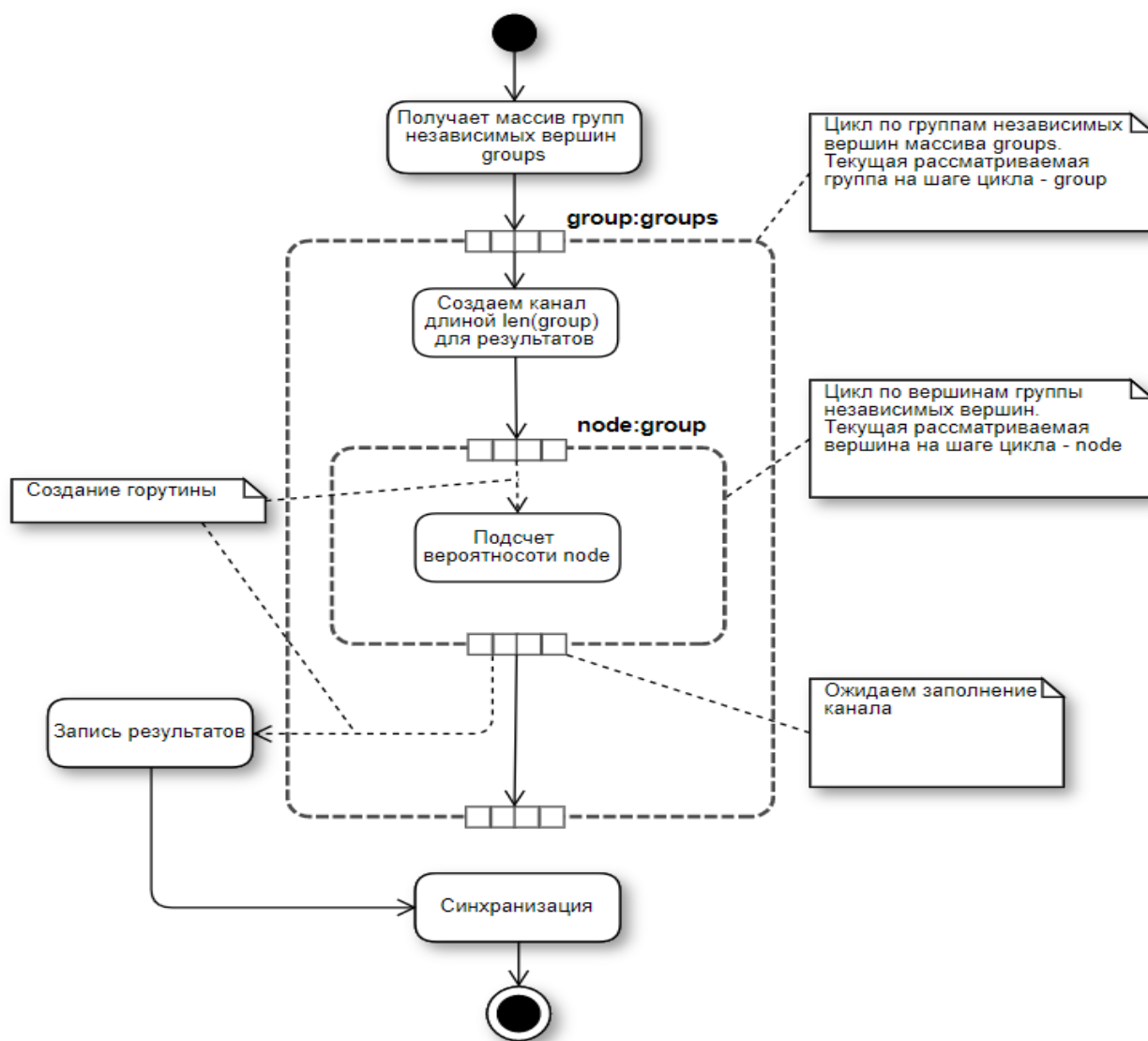


Рисунок 20- Алгоритм параллелизации вычислений

Изначально стартует цикл по списку групп независимых вершин, запуская цикл по каждой группе по порядку мы отправляем каждую вершину из группы на подсчёт в отдельные горутин. В это время горутин отвечающая за основной поток программы, будет ожидать окончания подсчёта всех вершин в группе, по завершению синхронизирует горутин и запишет результаты подсчёта в словарь, хранящий все результаты.

Произведя данные изменение получилось добиться максимально возможной оптимизации прямого вероятностного вывода.

Финальная версия полного алгоритма показана на Рисунке 21.

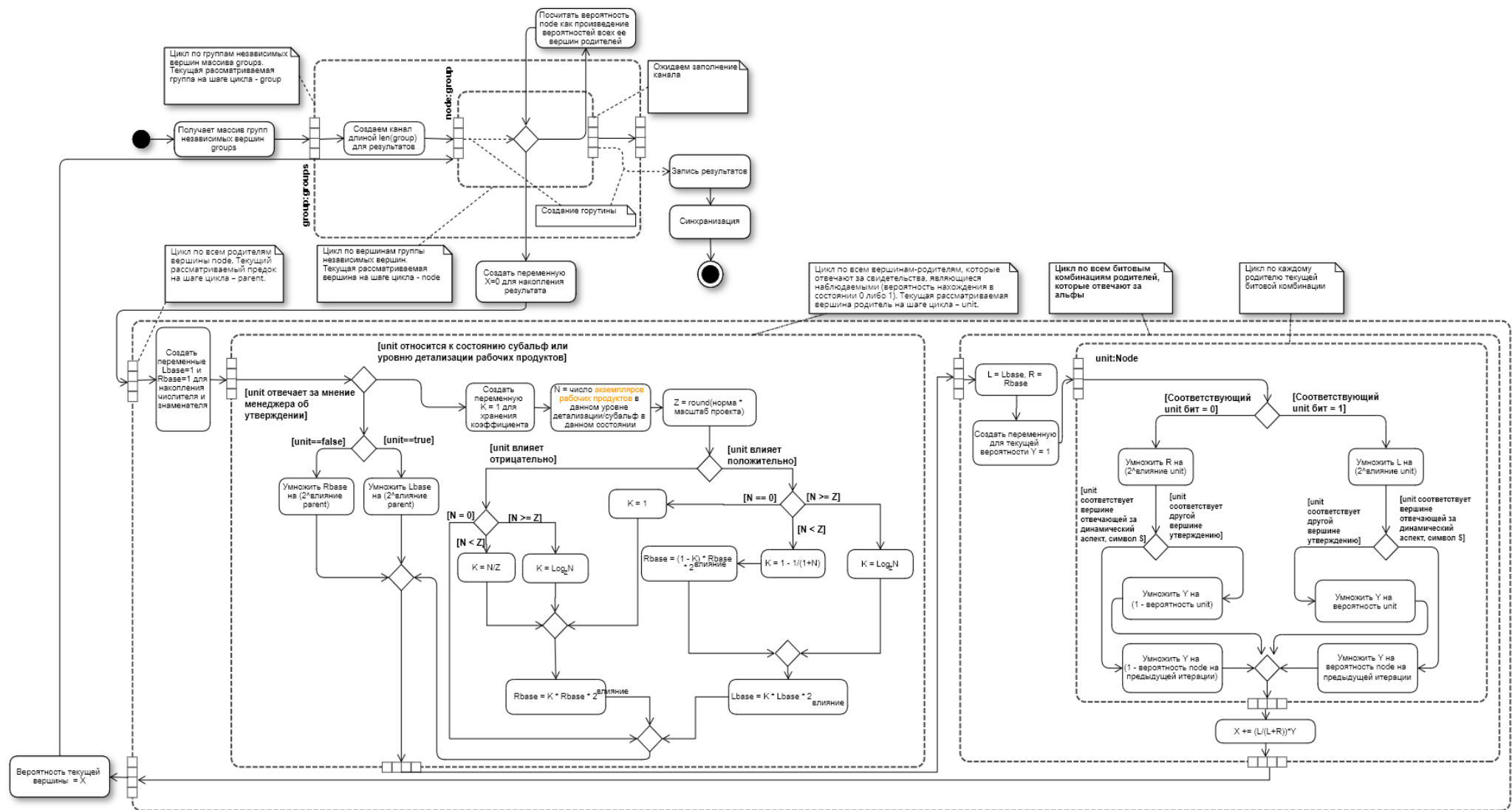


Рисунок 21- Финальная версия алгоритма прямого вероятностного вывода

3 ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

Система была декомпозирована на отдельные компоненты согласно научному подходу. Логика вычислений, ранее реализованная в виде плагина для платформы Redmine, была вынесена в отдельный сервис, реализованный на языке программирования Golang. Такое решение обеспечивает возможность более оперативного внесения изменений в алгоритм. Более того, выделение отдельного сервиса расширяет возможности для параллельных вычислений, что положительно сказывается на скорости выполнения расчетов.

1. Разработанный плагин для платформы Redmine представляет собой программное средство, предназначенное для выполнения операций создания, чтения, обновления и удаления данных, связанных с состоянием проекта. Эти данные включают информацию о состоянии чекбоксов и наличии альф в рамках проекта. Для хранения всех собранных данных используется база данных MySQL. Кроме того, плагин обеспечивает отображение результатов вычислений, осуществляемых другими компонентами системы.
2. Ключевым компонентом системы является "main", реализованный на языке программирования Golang. Данный компонент предоставляет программный интерфейс (API), который обеспечивает взаимодействие между вычислительной частью системы и плагином, используемым в Redmine. Реализация компонента "main" позволяет эффективно обмениваться данными и передавать управление между различными модулями системы, обеспечивая их согласованное функционирование.
3. Компонент "calculate_module" представляет собой программный модуль, разработанный с целью проведения вычислительных операций, направленных на прямой вероятностный вывод, с целью определения вероятности возникновения ошибки со стороны

менеджера. Задачей данного компонента является выполнение математических и статистических расчетов, основанных на доступных данных и моделях, для определения точности и надежности действий менеджера. Это позволяет системе предоставить объективную оценку вероятности возникновения ошибки и принять соответствующие решения на основе этих результатов.

4. Компонент "api_neo4j" представляет собой программный модуль, разработанный для обеспечения взаимодействия между системой и графовой базой данных Neo4j Graph Database. Данный компонент отвечает за управление операциями чтения, записи и обновления данных, связанных с структурой байесовской сети, которая используется в системе. Благодаря использованию Neo4j Graph Database в качестве хранилища данных, система может эффективно обрабатывать сложные связи и отношения между элементами байесовской сети, обеспечивая высокую производительность и гибкость при работе с ними.
5. Компонент "proto_grpc" является авто сгенерированным программным модулем, который выполняет функцию определения методов, используемых в протоколе gRPC (Google Remote Procedure Call) [20]. Данный компонент отвечает за создание и описание интерфейсов и операций, которые могут быть вызваны и обработаны с использованием протокола gRPC. Определение методов в компоненте "proto_grpc" позволяет системе установить стандартизированное взаимодействие и обмен данными между клиентскими и серверными компонентами посредством gRPC, обеспечивая эффективную передачу информации и выполнение удаленных процедур.

Компоненты и их взаимодействие представлены на Рисунке 22.

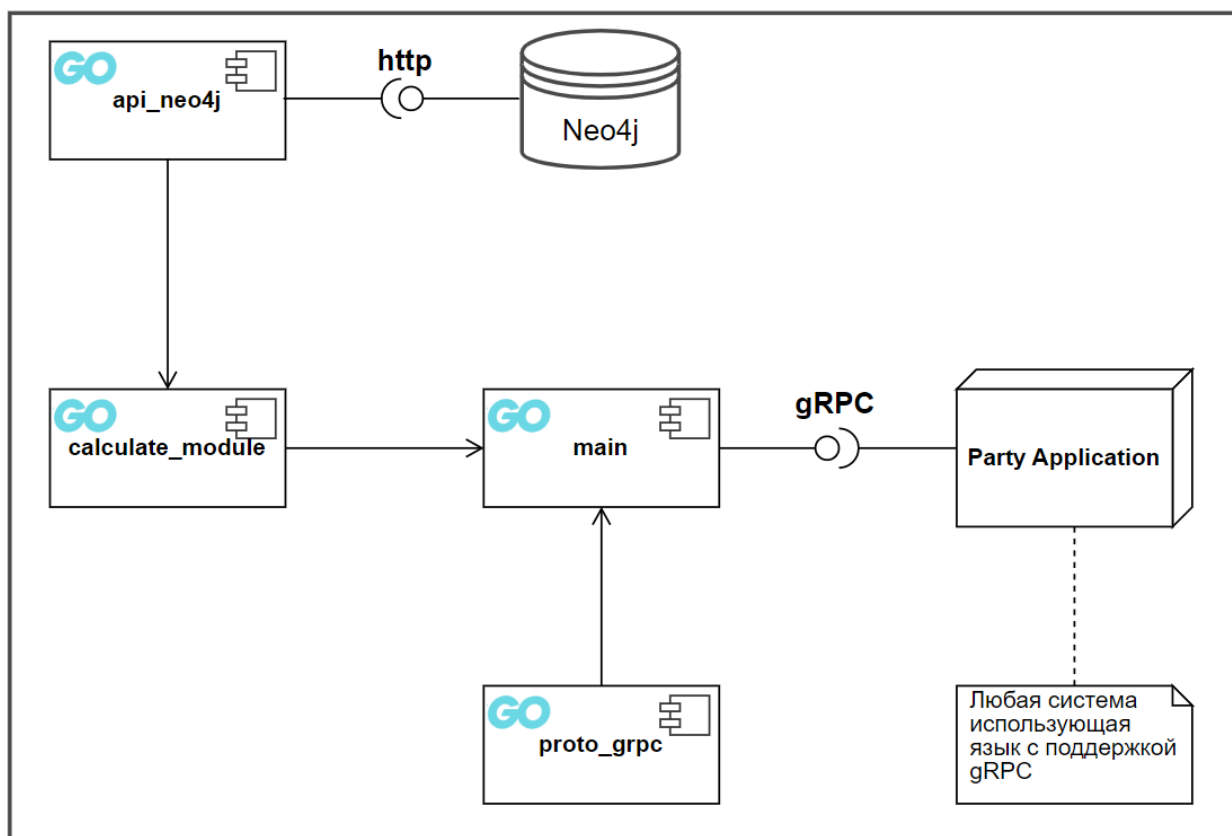


Рисунок 22 – Компоненты системы

3.1 Реализация взаимодействия с Neo4j

Первоначальная реализация взаимодействия с базой данных была разработана на языке программирования Python. Весь алгоритм, включая API, был написан с использованием библиотеки `py2neo`, которая предоставляет функциональность для работы с базой данных Neo4j.

Библиотека `py2neo` предоставляет как отдельные методы, предназначенные для выполнения операций с Neo4j, так и возможность выполнения пользовательских запросов к базе данных.

Однако, в последующих разработках, взаимодействие с базой данных Neo4j было переписано на язык программирования Golang. Это решение позволило улучшить производительность и эффективность взаимодействия с базой данных, а также обеспечило большую гибкость при разработке и поддержке системы.

3.2 Взаимодействие с внешней системой

Реализация взаимодействия с внешней системой осуществляется с применением протокола удаленного вызова процедур (Remote Procedure Calls, RPC) через систему gRPC, разработанную компанией Google. Данная система обеспечивает передачу данных и вызов удаленных процедур между системой принятия решений и визуальным представлением. В качестве сетевого транспорта для gRPC используется протокол HTTP/2, который обладает улучшенными характеристиками производительности по сравнению с HTTP 1.1, используемый в REST API. Это позволяет гарантировать более высокую скорость передачи данных и более эффективное взаимодействие между компонентами системы.

По умолчанию, система удаленного вызова процедур (gRPC) использует Protobuf (Protocol Buffers) [21] в качестве языка определения интерфейса (Interface Definition Language, IDL) для описания интерфейса службы и структуры сообщений, содержащих полезную нагрузку. Protobuf обеспечивает удобную и эффективную сериализацию данных для передачи между клиентом и сервером. Однако, при необходимости, имеется возможность использовать альтернативные инструменты для определения интерфейса и структур данных в gRPC. Такой подход позволяет разработчикам выбирать наиболее подходящий инструментарий, учитывая особенности и требования своего проекта.

Клиентские и серверные компоненты gRPC обладают способностью работать и взаимодействовать друг с другом в разнообразных средах и могут быть реализованы на любом из поддерживаемых языков программирования, предоставляемых gRPC. Это означает, что разработчики имеют возможность выбирать оптимальный язык программирования в соответствии с требованиями проекта и их собственными предпочтениями. Гибкость языковой поддержки в gRPC позволяет разрабатывать клиентские и серверные компоненты на популярных языках, таких как Java, C++, Python, Go, и других,

обеспечивая широкий спектр вариантов для разработки и интеграции систем, снижая преграды и повышая эффективность разработки.

С целью обеспечения взаимодействия между модулем расчетов и плагином для Redmine, был разработан gRPC-сервер на языке программирования Golang. Данный сервер реализует функциональность, позволяющую передавать запросы и данные между модулем расчетов и плагином для Redmine. В свою очередь, клиент, подключенный к плагину для Redmine, взаимодействует с gRPC-сервером для отправки запросов и получения результатов расчетов. Такое архитектурное решение обеспечивает эффективное и надежное взаимодействие между компонентами системы, обеспечивая передачу данных и управление операциями расчетов с высокой степенью надежности и гибкости.

С целью достижения оптимальности было принято решение описать взаимодействие между системами в виде единой функции. Данная функция принимает в качестве входных данных аргументы, передаваемые от Redmine, и возвращает словарь, содержащий результаты расчетов. Такой подход к организации взаимодействия позволяет объединить и структурировать операции, связанные с обменом информацией между системами, в одной функции. Это способствует упрощению кода и обеспечивает более эффективную обработку и возврат результатов расчетов, предоставляя компактный и удобный способ взаимодействия между системами.

На Листинге 2 показан protofile для системы поддержки принятия решений.

```
syntax = "proto3";

message Request{
    string map = 1;
    int32 iter = 2;
    double threshold = 3;
    string key = 4;
}

message Response{
    string res = 1;
}

service CalculatingService{
    rpc calculate (Request) returns
    (Response) {}
}
```

Листинг 2 – protofile

Из данного файла будет осуществляться генерация двух отдельных файлов, предназначенных для взаимодействия с использованием протокола gRPC - один файл для серверной стороны, а другой - для клиентской. Учитывая тот факт, что функционал серверной части реализован на языке программирования Golang, а плагин для Redmine написан на языке Rube, каждый из сгенерированных файлов будет соответствовать соответствующему языку программирования. Такое разделение файлов позволяет эффективно организовать взаимодействие между сервером и клиентом, обеспечивая соответствующие возможности для каждой стороны и облегчая разработку и интеграцию системы.

Давайте рассмотрим составные части protobuf-файла более детально:

- Request: это тип входных данных, который принимает 4 значения:
 - Map: строковая переменная, в которую записывается JSON-представление состояния вершин, установленное менеджером;
 - Iter: целочисленная переменная, содержащая номер итерации, которую необходимо выполнить;

- Threshold: переменная с плавающей точкой, представляющая границы точности, которые требуются для уверенности в правильности решения менеджера;
- Key: строковая переменная, содержащая безопасный ключ для доступа к API.
- Response: это тип выходных данных, который возвращает 3 значения:
 - Res: строковая переменная, содержащая JSON-представление словаря с парами {Имя вершины: Вероятность этой вершины быть истинной}.
- CalculatingService: это описание сервиса, включающего одну функцию "calculate", которая вызывает метод прямого вероятностного вывода и возвращает соответствующие значения.

Приведенные составные части protobuf-файла определяют структуру входных и выходных данных, а также описывают сервис и его функциональность, связанную с выполнением расчетов. Такая спецификация позволяет эффективно передавать и обрабатывать данные между компонентами системы, обеспечивая структурированное взаимодействие и обмен информацией.

3.3 Результаты

В результате применения разработанного инструмента была выполнена моделирование сети, содержащей 483 вершины и 1702 связи (максимальное количество родителей для одной вершины составляет 15). Для данной сети было проведено вычисление прямого вероятностного вывода в течение одной итерации, и это заняло 3,1 секунды. Таким образом, полученные результаты свидетельствуют о достигнутой производительности инструмента при обработке сети с такими характеристиками, что отражает его эффективность и способность выполнять вычисления с высокой скоростью.

На рисунке 23 представлен скриншот результатов вычисления прямого вероятностного вывода, полученных при использовании разработанного инструмента. Этот скриншот отображает полученные значения и результаты

расчетов, которые являются результатом применения инструмента к соответствующей сети и входным данным. Представление результатов визуально позволяет оценить эффективность инструмента и рассмотреть полученные выводы для дальнейшего анализа и интерпретации.

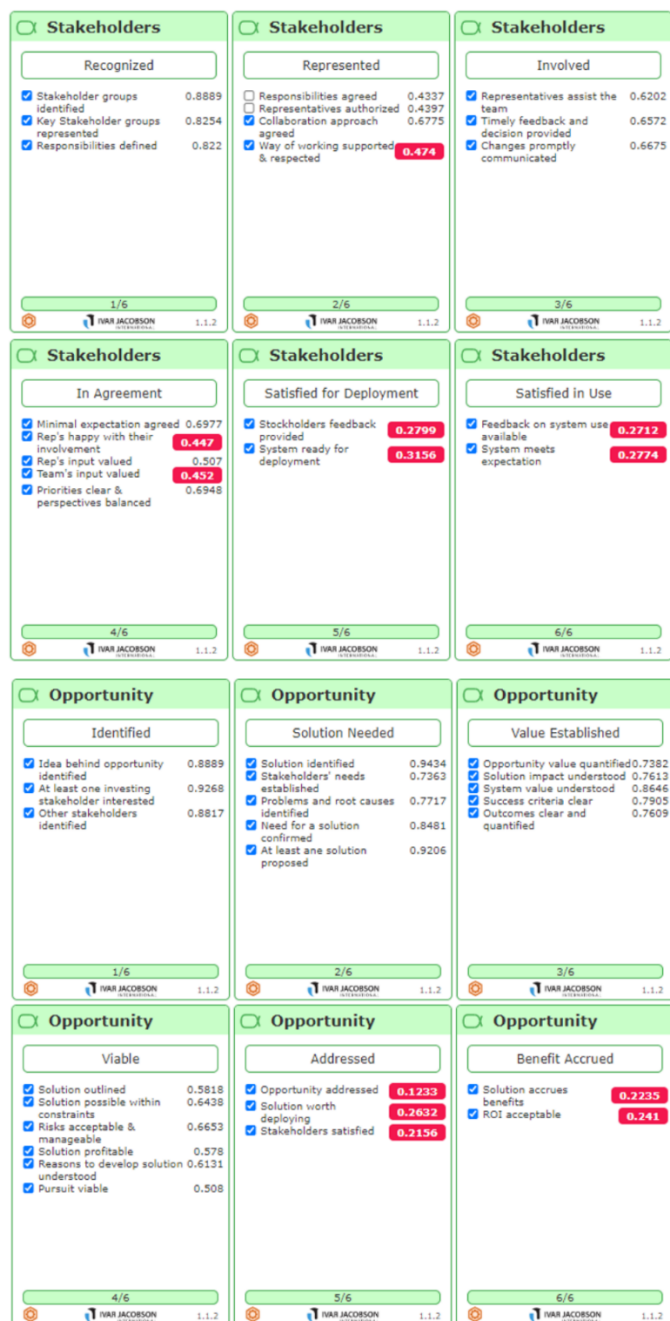


Рисунок 23 – Результаты выполнения прямого вероятностного вывода

ЗАКЛЮЧЕНИЕ

В данной исследовательской работе были представлены различные методы, примененные для реализации Байесовского подхода в системе поддержки принятия решений. Особое внимание было уделено методам оптимизации прямого вероятностного вывода, которые позволили достичь максимального улучшения производительности системы.

В контексте требования №1, применение графовой системы управления базами данных (СУБД) позволяет преодолеть ограничения на количество вершин в сети, что способствует удовлетворению данного требования. Кроме того, применение графовой СУБД снимает ограничение на количество родительских вершин (с точки зрения хранения) и обеспечивает быструю перестройку сети. Важным аспектом является интеграция системы, и для этого были использованы API и библиотеки для взаимодействия с Neo4j, предоставляемые разработчиком графовой СУБД.

Разработка сервера для взаимодействия с другими системами открывает возможность интеграции данного метода в различные программные системы, которые используют поддерживаемый gRPC язык программирования.

В рамках данной работы разработанная система была зарегистрирована [22] и введена в эксплуатацию.

В дальнейшем данная работа будет продолжена с целью расширения возможностей системы, включая предложение последующих итераций разработки программного обеспечения.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Змеев Д. О. Прототип системы поддержки принятия решений для управления проектами на основе стандарта OMG Essence и байесовских сетей: диссертация на соискание ученой степени кандидата технических наук: 05.13.11 / Змеев Денис Олегович; науч. Рук. Змеев О. А.; Томский гос. Ун-т. [Электронный ресурс] URL: <http://vital.lib.tsu.ru/vital/access/manager/Repository/koha:000894552/> (дата обращения: 10.02.2022).
2. Jacobson I., Ng P.–W., McMahon P. E., Spence I., Lidman S. The Essence of Software Engineering: Applying the SEMAT Kernel. / Addison–Wesley, 2013. 224 p. ISBN 978–0321885951.
3. Вероятностный вывод [Электронный ресурс]. // wikipedia URL: <https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B9%D0%B5%D1%81%> (дата обращения: 19.10.2022).
4. GeNIe Modeler: Complete Modeling Freedom // GeNIe Modeler URL: <https://www.bayesfusion.com/genie/> (дата обращения: 25.04.2023).
5. Redmine [Электронный ресурс]. URL: <https://www.redmine.org/> (дата обращения: 10.12.2021).
6. AgenaRisk: Bayesian Network Software for Risk Analysis and Decision Making [Электронный ресурс]. URL: <https://www.agenarisk.com> (дата обращения: 02.02.2022).
7. BayesFusion, LLC. Data Analytics, Modeling, Decision Support [Электронный ресурс]. URL: <https://www.bayesfusion.com> (дата обращения: 03.02.2022).
8. Netica Application [Электронный ресурс]. URL: <https://www.norsys.com/netica.html> (дата обращения: 03.03.2022).
9. BayesiaLab 10 [Электронный ресурс]. URL: <https://www.bayesia.com> (дата обращения: 10.02.2022).
10. BayesAML [Электронный ресурс]. URL: <https://www.hugin.com/bayesaml/> (дата обращения: 10.02.2022).

- 11.MSBNx. Bayesian Network Editor and ToolKit. [Электронный ресурс]. URL: <https://msbnx.azurewebsites.net> (дата обращения: 15.02.2022).
- 12.Bayes Server [Электронный ресурс]. URL: <https://www.bayesserver.com> (дата обращения: 15.02.2022).
- 13.Why is a Goroutine's stack infinite? // Dave Cheney URL: <https://dave.cheney.net/2013/06/02/why-is-a-goroutines-stack-infinite> (дата обращения: 19.04.2023).
- 14.Wet Grass [Электронный ресурс] // Bayes Net Library URL: <https://www.norsys.com/netlibrary/index.htm> (дата обращения: 21.10.2022).
- 15.Robinson I., Webber J., Eifrem E. Graph Databases. – O'Reilly Media, 2013. – 178 p. – ISBN 978–1449356262.
- 16.Neo4j Graph Database [Электронный ресурс]. URL: <https://neo4j.com/product/neo4j-graph-database/> (дата обращения: 15.02.2022).
- 17.Д. О. Змеев, О. А. Змеев, Л. С. Иванова Практика работы с антипаттернами для Essence Practice Library // "Программная инженерия". – 2022. – №7. – С. 311–321.
- 18.Bayes' rule: Log-odds form [Электронный ресурс] // Arbital URL: https://arbital.com/p/bayes_log_odds/ (дата обращения: 21.10.2022).
- 19.Goroutines in Go (Golang) // Golangbyexample URL: <https://golangbyexample.com/goroutines-golang/> (дата обращения: 01.05.2023).
- 20.Why gRPC? // grpc.io URL: <https://grpc.io> (дата обращения: 12.02.2023).
- 21.Protocol Buffers // protobuf.dev URL: <https://protobuf.dev> (дата обращения: 15.02.2023).
- 22.Свидетельство о государственной регистрации программы для ЭВМ № 2023612258. Программа для расчета вероятности ложноположительной ошибки менеджера с применением байесовских сетей / Иванова Л. С. (RU), Змеев Д. О. (RU), Фрейдин В. И. (RU); правообладатели: Иванова Л. С. (RU), Змеев Д. О. (RU), Фрейдин В. И. (RU). – Заявка № 2022683135; заявл. 28.11.2022; дата регистрации 01.02.2023. – 1 с.

Отчет о проверке на заимствования №1



Автор: Фрейдин В И
Проверяющий: Политова Анастасия Михайловна
Организация: Томский Государственный Университет

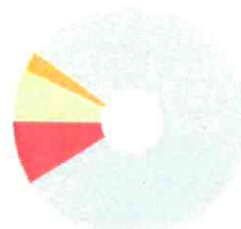
Отчет предоставлен сервисом «Антиплагиат» - <http://tsu.antiplagiat.ru>

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 56
Начало загрузки: 31.05.2023 09:03:29
Длительность загрузки: 00:00:20
Имя исходного файла:
Diplom_final_Freydin.docx
Название документа: Diplom_final_Freydin
Размер текста: 56 кБ
Тип документа: Выпускная квалификационная работа
Символов в тексте: 57534
Слов в тексте: 6769
Число предложений: 372

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Начало проверки: 31.05.2023 09:03:50
Длительность проверки: 00:01:48
Комментарии: не указано
Поиск с учетом редактирования: да
Проверенные разделы: титульный лист с. 1, содержание с. 2, основная часть с. 3-51, библиография с. 52-53
Модули поиска: ИПС Адилет, Библиография, Сводная коллекция ЭБС, Интернет Плюс*, Сводная коллекция РГБ, Цитирование, Переводные заимствования (RuEn), Переводные заимствования по eLIBRARY.RU (EnRu), Переводные заимствования по коллекции Гарант: аналитика, Переводные заимствования по коллекции Интернет в английском сегменте, Переводные заимствования по Интернету (EnRu), Переводные заимствования по коллекции Интернет в русском сегменте, Переводные заимствования издательства Wiley, eLIBRARY.RU, СПС ГАРАНТ: аналитика, СПС ГАРАНТ: нормативно-правовая документация, Медицина, Диссертации НББ, Коллекция НБУ, Перефразирования по eLIBRARY.RU, Перефразирования по СПС ГАРАНТ: аналитика, Перефразирования по Интернету, Перефразирования по Интернету (EN), Перефразированные заимствования по коллекции Интернет в английском сегменте, Перефразированные заимствования по коллекции Интернет в русском сегменте, Перефразирования по коллекции издательства Wiley, Патенты СССР, РФ, СНГ, СМИ России и СНГ, Шаблонные фразы, Модуль поиска "tsu", Кольцо вузов, Издательство Wiley, Переводные заимствования



СОВПАДЕНИЯ
8,53%

САМОЦИТИРОВАНИЯ
2,68%

ЦИТИРОВАНИЯ
7,05%

ОРИГИНАЛЬНОСТЬ
81,74%

Совпадения — фрагменты проверяемого текста, полностью или частично сходные с найденными источниками, за исключением фрагментов, которые система отнесла к цитированию или самоцитированию. Показатель «Совпадения» — это доля фрагментов проверяемого текста, отнесенных к совпадениям, в общем объеме текста.

Самоцитирования — фрагменты проверяемого текста, совпадающие или почти совпадающие с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа. Показатель «Самоцитирования» — это доля фрагментов текста, отнесенных к самоцитированию, в общем объеме текста.

Цитирования — фрагменты проверяемого текста, которые не являются авторскими, но которые система отнесла к корректно оформленным. К цитированиям относятся также шаблонные фразы; библиография; фрагменты текста, найденные модулем поиска «СПС Гарант: нормативно-правовая документация». Показатель «Цитирования» — это доля фрагментов проверяемого текста, отнесенных к цитированию, в общем объеме текста.

Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.

Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.

Оригинальный текст — фрагменты проверяемого текста, не обнаруженные ни в одном источнике и не отмеченные ни одним из модулей поиска. Показатель «Оригинальность» — это доля фрагментов проверяемого текста, отнесенных к оригинальному тексту, в общем объеме текста.

«Совпадения», «Цитирования», «Самоцитирования», «Оригинальность» являются отдельными показателями, отображаются в процентах и в сумме дают 100%, что соответствует полному тексту проверяемого документа.

Обращаем Ваше внимание, что система находит текстовые совпадения проверяемого документа с проиндексированными в системе источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности совпадений или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в тексте	Доля в отчете	Источник	Актуален на	Модуль поиска	Комментарии
[01]	5,83%	5,83%	не указано	29 Сен 2022	Библиография	
[02]	3,76%	0%	https://journal.tsur.ru/storage/157645/%D0%B6%D1... https://journal.tsur.ru	19 Дек 2022	Интернет Плюс*	
[03]	2,68%	2,68%	Разработка подсистемы работы с байесовскими сет... https://elibrary.ru	31 Дек 2022	eLIBRARY.RU	
[04]	2,18%	2,18%	Применение байесовского подхода в измерениях а... http://elibrary.ru	31 Авг 2017	Перефразирования по eLIBRARY.RU	
[05]	1,98%	0,6%	ВКР_Аносов А.В.	04 Июн 2021	Модуль поиска "tsu"	
[06]	1,98%	0%	ВКР_Аносов	03 Июн 2021	Модуль поиска "tsu"	
[07]	1,98%	0%	Vkr_anosov	28 Фев 2022	Модуль поиска "tsu"	
[08]	1,96%	0,84%	Змеев, Денис Олегович Прототип системы поддерж... http://dlib.rsl.ru	14 Ноя 2022	Сводная коллекция РГБ	
[09]	1,83%	0,83%	https://vital.lib.tsu.ru/vital/access/services/Download/... https://vital.lib.tsu.ru	07 Мая 2023	Интернет Плюс*	

[10]	1,51%	0%	Культура и искусство: поиски и открытия. Сборник ... http://bibliorossica.com	21 Янв 2020	Сводная коллекция ЭБС	
[11]	1,44%	0,3%	Лекция №6. БАЙЕСОВСКИЕ СЕТИ ДОВЕРИЯ КАК СРЕД... http://habarov.spb.ru	24 Ноя 2022	Интернет Плюс*	
[12]	1,44%	0%	Лекция №6. БАЙЕСОВСКИЕ СЕТИ ДОВЕРИЯ КАК СРЕД... http://habarov.spb.ru	26 Июн 2020	Интернет Плюс*	
[13]	1,41%	1,21%	не указано	29 Сен 2022	Шаблонные фразы	
[14]	1,37%	1,04%	26.05.05_Степанчук В.А._2022	17 Мар 2023	Кольцо вузов	
[15]	1,3%	0%	ahmetyanov_a_r_sistema-sbora-i-analiza-novostnyh-d...	19 Мая 2023	Кольцо вузов	
[16]	1,3%	1,09%	Что такое горутины и каков их размер? https://habr.com	11 Дек 2020	СМИ России и СНГ	
[17]	1,23%	0%	nazmutdinov_r_r_mobilnoe-prilojenie-dlya-organizacii-...	20 Мая 2023	Кольцо вузов	
[18]	1,06%	0%	2023BKP730301ЗИЯЕВ	25 Мая 2023	Кольцо вузов	
[19]	0,98%	0,39%	MAGISTERSKAYA_POLISHCHUK_022110- (1).docx	27 Мая 2023	Модуль поиска "tsu"	
[20]	0,86%	0%	Баландинские чтения. Т. XI сборник статей научных ... http://dlib.rsl.ru	12 Окт 2017	Сводная коллекция РГБ	
[21]	0,85%	0%	Международный терроризм: происхождение, эвол... http://studentlibrary.ru	19 Дек 2016	Медицина	
[22]	0,8%	0%	Лекция №6. БАЙЕСОВСКИЕ СЕТИ ДОВЕРИЯ КАК СРЕД... https://steptosleep.ru	24 Ноя 2022	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[23]	0,8%	0%	Лекция №6. БАЙЕСОВСКИЕ СЕТИ ДОВЕРИЯ КАК СРЕД... https://steptosleep.ru	24 Ноя 2022	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[24]	0,78%	0,78%	Скачать 56-й выпуск с http://wisesoft.spb.ru ... (4/8) http://wisesoft.spb.ru	01 Янв 2017	Перефразирования по Интернету	
[25]	0,75%	0%	https://csi.tsu.ru/sites/default/files/%D0%A5%D1%80%... https://csi.tsu.ru	25 Мая 2023	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[26]	0,75%	0%	https://csi.tsu.ru/sites/default/files/%D0%A5%D1%80%... https://csi.tsu.ru	25 Мая 2023	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[27]	0,72%	0%	ЭС_Отчёт8 http://studfiles.ru	07 Янв 2017	Перефразирования по Интернету	
[28]	0,67%	0%	МНД Аналитические обзоры 2021-2022	17 Янв 2023	Кольцо вузов	
[29]	0,5%	0%	VKR_-_Zhilin_Sergey.docx	18 Июн 2020	Модуль поиска "tsu"	Источник исключен. Причина: Маленький процент пересечения.
[30]	0,48%	0%	Замараева, Юлия Сергеевна Региональные культур... http://dlib.rsl.ru	08 Ноя 2022	Сводная коллекция РГБ	
[31]	0,48%	0%	Реферат: Применение Байесовых сетей - BestReferat... https://bestreferat.ru	24 Ноя 2022	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[32]	0,47%	0%	Койстинен_BKP_M.pdf	28 Мая 2021	Модуль поиска "tsu"	
[33]	0,47%	0,47%	https://www.theses.fr/2018NANT4019/abes https://theses.fr	14 Янв 2023	Переводные заимствования по коллекции Интернет в английском сегменте	
[34]	0,46%	0%	Планирование образовательного процесса в учрежд... http://dep.nlb.by	04 Июл 2017	Диссертации НББ	Источник исключен. Причина: Маленький процент пересечения.
[35]	0,43%	0%	Вагнер, Дмитрий Викторович Высокочастотные эле... http://dlib.rsl.ru	27 Дек 2019	Сводная коллекция РГБ	Источник исключен. Причина: Маленький процент пересечения.
[36]	0,41%	0%	"Научно-аналитический журнал ""Инновации и инв... https://book.ru	03 Июл 2017	Сводная коллекция ЭБС	Источник исключен. Причина: Маленький процент пересечения.
[37]	0,39%	0%	ВКР-ВЫШЕБАБА.docx	02 Июн 2022	Модуль поиска "tsu"	
[38]	0,35%	0%	Актуальные вопросы разработки системы управлен... http://ivo.garant.ru	20 Авг 2022	СПС ГАРАНТ: аналитика	Источник исключен. Причина: Маленький процент пересечения.
[39]	0,35%	0%	Стандарты доказывания: аналитическое и эмпирич... http://ivo.garant.ru	30 Ноя 2019	Перефразированные заимствования по коллекции Интернет в русском сегменте	Источник исключен. Причина: Маленький процент пересечения.
[40]	0,35%	0%	Метод байесовских сетей и ключевые аспекты байе... http://elibrary.ru	04 Янв 2019	Перефразирования по eLIBRARY.RU	Источник исключен. Причина: Маленький процент пересечения.
[41]	0,34%	0%	https://www.tsu.ru/upload/medialibrary/9ff/metodich... https://tsu.ru	12 Янв 2022	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[42]	0,33%	0%	Эмеев, Денис Олегович Прототип системы поддерж... http://dlib.rsl.ru	11 Окт 2022	Сводная коллекция РГБ	Источник исключен. Причина: Маленький процент пересечения.
[43]	0,33%	0%	https://esu.citis.ru/dissertation/2NVQSTL13CEGSBDO8... https://esu.citis.ru	20 Мар 2018	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[44]	0,31%	0%	https://vital.lib.tsu.ru/vital/access/services/Download/... https://vital.lib.tsu.ru	07 Мая 2023	Переводные заимствования по коллекции Интернет в русском сегменте	Источник исключен. Причина: Маленький процент пересечения.
[45]	0,3%	0%	Агаджанян, Альберт Грантович Метод и алгоритмы ... http://dlib.rsl.ru	21 Сен 2021	Сводная коллекция РГБ	Источник исключен. Причина: Маленький процент пересечения.
[46]	0,29%	0%	Вопросы и ответы для собеседования Go-разработч... https://habr.com	20 Июн 2022	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[47]	0,28%	0%	https://www.ugatu.su/media/uploads/MainSite/Scienc... https://ugatu.su	24 Сен 2022	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[48]	0,27%	0%	Что такое горутины и каков их размер? / Хабр https://habr.com	25 Мая 2023	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.

[49]	0,26%	0%	https://core.ac.uk/download/pdf/333610699.pdf https://core.ac.uk	12 Мар 2022	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[50]	0,25%	0%	Проектирование DSL на основе метамодели предме... https://docplayer.ru	13 Окт 2021	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[51]	0,24%	0%	Курсовая. Бельская.docx	01 Июн 2020	Модуль поиска "tsu"	Источник исключен. Причина: Маленький процент пересечения.
[52]	0,24%	0%	Казенова_ВКР.pdf	11 Июн 2020	Модуль поиска "tsu"	Источник исключен. Причина: Маленький процент пересечения.
[53]	0,23%	0%	Исследование и разработка эффективных композиц... http://diss.natlib.uz	29 Авг 2014	Коллекция НБУ	Источник исключен. Причина: Маленький процент пересечения.
[54]	0,21%	0%	Разработка и исследование свойств полимерных ко... http://diss.natlib.uz	29 Авг 2014	Коллекция НБУ	Источник исключен. Причина: Маленький процент пересечения.
[55]	0,2%	0%	Патентный анализ перспектив развития в России зе...	14 Дек 2022	СМИ России и СНГ	Источник исключен. Причина: Маленький процент пересечения.
[56]	0,2%	0%	НОВЫЕ ТЕХНОЛОГИИ FINTECH В ЮЖНОЙ КОРЕЕ: К-...	17 Авг 2019	СМИ России и СНГ	Источник исключен. Причина: Маленький процент пересечения.
[57]	0,2%	0%	jubanov_a_e_ispolzovanie-sistem-podderjki-prinyatiya-...	25 Мая 2023	Кольцо вузов	Источник исключен. Причина: Маленький процент пересечения.
[58]	0,2%	0%	Медико-социальные аспекты депрессивных расстр... http://emll.ru	21 Дек 2016	Медицина	Источник исключен. Причина: Маленький процент пересечения.
[59]	0,2%	0%	Апелляционное определение СК по гражданским де... http://ivo.garant.ru	08 Янв 2022	СПС ГАРАНТ: нормативно-правовая документация	Источник исключен. Причина: Маленький процент пересечения.
[60]	0,18%	0%	pdf https://vvsu.ru	25 Мая 2023	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[61]	0,18%	0%	Технология разработки программного обеспечени... https://esstu.ru	25 Мая 2023	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[62]	0,17%	0%	не указано http://dspace.nbuv.gov.ua	25 Мая 2023	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[63]	0,16%	0%	Сетевые анализаторы и анализаторы протоколов се... https://bibliofond.ru	25 Мая 2023	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[64]	0,16%	0%	Управление талантами и трансформация корпорат... https://e.lanbook.com	22 Янв 2020	Сводная коллекция ЭБС	Источник исключен. Причина: Маленький процент пересечения.
[65]	0,16%	0%	Адаптивные системы https://e.lanbook.com	22 Янв 2020	Сводная коллекция ЭБС	Источник исключен. Причина: Маленький процент пересечения.
[66]	0,16%	0%	Управление талантами и трансформация корпорат... http://biblioclub.ru	21 Янв 2020	Сводная коллекция ЭБС	Источник исключен. Причина: Маленький процент пересечения.
[67]	0,14%	0%	Язык и культура : сборник статей XXVII Междунаро... http://biblioclub.ru	21 Янв 2020	Сводная коллекция ЭБС	Источник исключен. Причина: Маленький процент пересечения.
[68]	0,14%	0%	Наш перевод статьи: Двадцать пять целей индустри... http://habrahabr.ru	21 Авг 2015	СМИ России и СНГ	Источник исключен. Причина: Маленький процент пересечения.
[69]	0,13%	0%	https://www.iacp.dvo.ru/resources/fileman/Uploads/di... https://iacp.dvo.ru	29 Дек 2022	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[70]	0,12%	0%	Разработка методологического подхода и техниче... http://diss.natlib.uz	21 Фев 2017	Коллекция НБУ	Источник исключен. Причина: Маленький процент пересечения.
[71]	0,11%	0%	Разработка протоколов для криптографической за... http://diss.natlib.uz	03 Июн 2017	Коллекция НБУ	Источник исключен. Причина: Маленький процент пересечения.
[72]	0,11%	0%	https://www.hse.ru/data/2022/02/21/1749376530/MIE... https://hse.ru	23 Сен 2022	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[73]	0,11%	0%	http://asu.ugatu.ac.ru/library/56025/42717fe80b8fb33ff... http://asu.ugatu.ac.ru	25 Мая 2023	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[74]	0,1%	0%	http://www.issp.ac.ru/ebooks/conf/youth_science_201... http://issp.ac.ru	24 Апр 2023	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[75]	0,1%	0%	https://www.iupr.ru/_files/ugd/b06fdc_20e71ae194464... https://iupr.ru	25 Фев 2023	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.