



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

Метод построения поисковых индексов в реляционной базе данных на основе глубоких нейронных сетей

Студент: Маслова Марина Дмитриевна ИУ7-83Б
Руководитель: Оленев Антон Александрович

Москва, 2023



**Метод построения поисковых индексов
в реляционной базе данных
на основе глубоких нейронных сетей**

Студент: Маслова Марина Дмитриевна ИУ7-83Б
Руководитель: Озиев Антон Александрович

Здравствуйте, уважаемые члены комиссии! ...?

С развитием технологий становится возможным применение новых подходов к решению классических задач.

Так, последнее время стали проводиться исследования по применению методов машинного обучения к построению поисковых индексов.

Поэтому было решено внести небольшой вклад в развитие данного направления исследований и разработать...

Цель и задачи

Цель: разработка метода построения поисковых индексов в реляционной базе данных на основе глубоких нейронных сетей.

Задачи:

- рассмотреть и сравнить известные методы построения индексов;
- привести описание построения индексов с помощью нейронных сетей;
- разработать метод построения индексов в реляционной базе данных на основе глубоких нейронных сетей;
- разработать программное обеспечение, реализующее данный метод;
- провести исследование (по времени и памяти) операций поиска и вставки с использованием индекса, построенного разработанным методом, при различных объемах данных.

Цель и задачи

Цель и задачи

Цель: разработка метода построения поисковых индексов в реляционной базе данных на основе глубоких нейронных сетей.

Задачи:

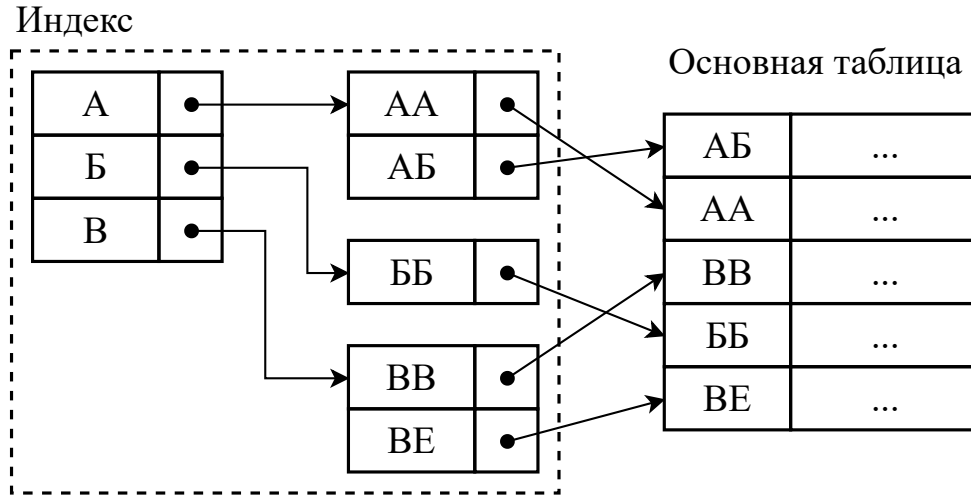
- рассмотреть и сравнить известные методы построения индексов;
- привести описание построения индексов с помощью нейронных сетей;
- разработать метод построения индексов в реляционной базе данных на основе глубоких нейронных сетей;
- разработать программное обеспечение, реализующее данный метод;
- провести исследование (по времени и памяти) операций поиска и вставки с использованием индекса, построенного разработанным методом, при различных объемах данных.

метод построения поисковых индексов в реляционной базе данных на основе глубоких нейронных сетей, что и стало целью данной работы.

Для достижения цели необходимо было проанализировать классические методы построения индексов, описать построение индексов с использованием нейронных сетей, разработать метод и реализующее его программное обеспечение, с помощью которого провести исследование.

Классические структуры индексов

Индекс — это некоторая структура, обеспечивающая быстрый поиск записей в базе данных за счет определяет соответствие ключа поиска конкретной записи с положением этой записи.



Основные типы структур:

- В-деревья;
- хеш-таблицы;
- битовые карты.

Классические структуры индексов

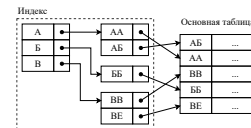
Как известно индекс представляет собой структуру, строящуюся в дополнение к существующим данным и обеспечивающую быстрый поиск записей в базе данных.

Обычно за основу индекса берут структуру, обеспечивающую выполнение той или иной подзадачи поиска:

- b-деревья для поиска в диапазоне;
- хеш-таблицы для поиска единичных ключей;
- битовые карты, на которых, основан фильтр Блума, — для проверки существования ключа.

Классические структуры индексов

Индекс — это некоторая структура, обеспечивающая быстрый поиск записей в базе данных за счет определяет соответствие ключа поиска конкретной записи с положением этой записи.

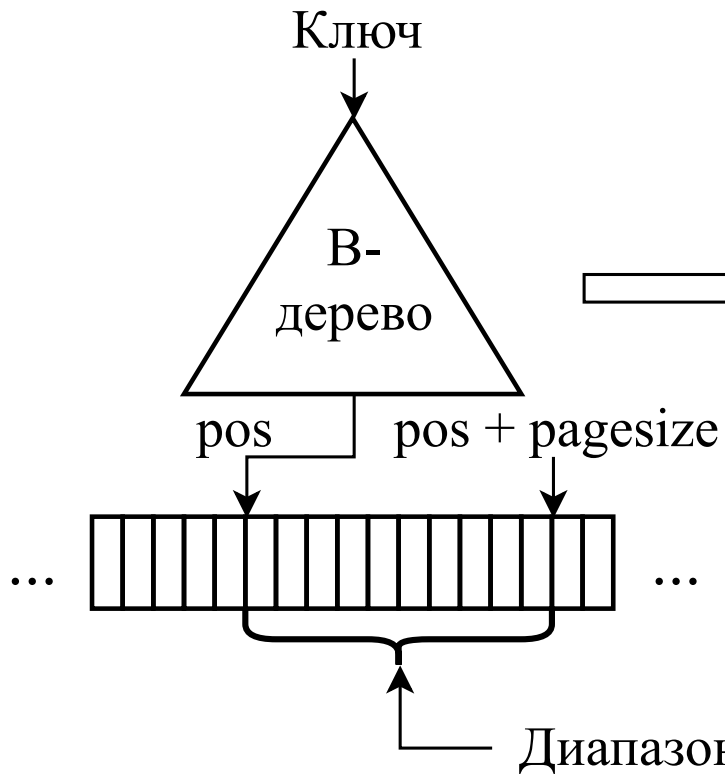


Основные типы структур:

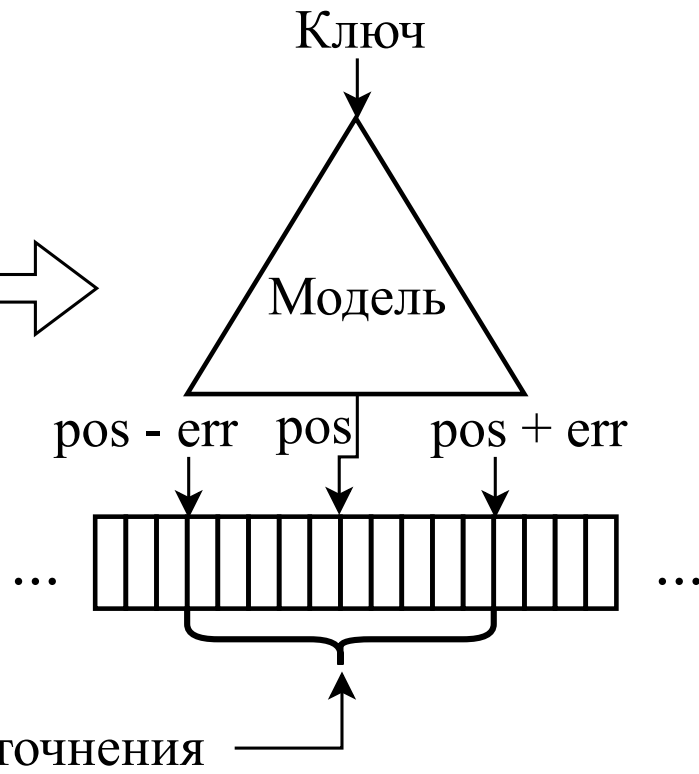
- B-деревья;
- хеш-таблицы;
- битовые карты.

Обученные индексы

Индекс на основе В-дерева

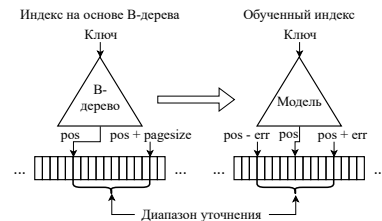


Обученный индекс



Обученные индексы

Обученные индексы



Однако каждая из этих структур может быть на более высоком уровне представлена в виде модели, сопоставляющей ключ с позицией искомой записи в некотором массиве.

Таким образом, классическая структура может быть заменена моделью машинного обучения, предсказывающей позицию по ключу. Индексы с такой моделью называют обученными.

Они предсказывают положение с ошибкой, что требует добавления второго этапа для уточнения позиции диапазоне, заданным этой ошибкой.

Сравнение методов построения индексов

Метод		В-дерево	Хеш-таблица	Фильтр Блума	Обученные индексы
Временная сложность	поиска	$O(\log N)$	$O(1) / O(N)$	$O(k)$	$O(1) / O(N)$
	вставки	$O(\log N)$	$O(1) / O(N)$	$O(k)$	(*)
Память		Высокая	Средняя	Низкая	Средняя
Поиск в диапазоне		+	-	-	+
Поиск единичного ключа		+	+	-	+
Проверка существования		+	+	+	+

(*) — вставка в обученный индекс требует переобучения, сложность которого зависит от архитектуры используемой модели машинного обучения.

Сравнение методов построения индексов

Сравнение методов построения индексов

Метод		В-дерево	Хеш-таблица	Фильтр Блума	Обученные индексы
Временная сложность	поиска	$O(\log N)$	$O(1) / O(N)$	$O(k)$	$O(1) / O(N)$
	вставки	$O(\log N)$	$O(1) / O(N)$	$O(k)$	(*)
Память		Высокая	Средняя	Низкая	Средняя
Поиск в диапазоне		+	-	-	+
Поиск единичного ключа		+	+	-	+
Проверка существования		+	+	+	+

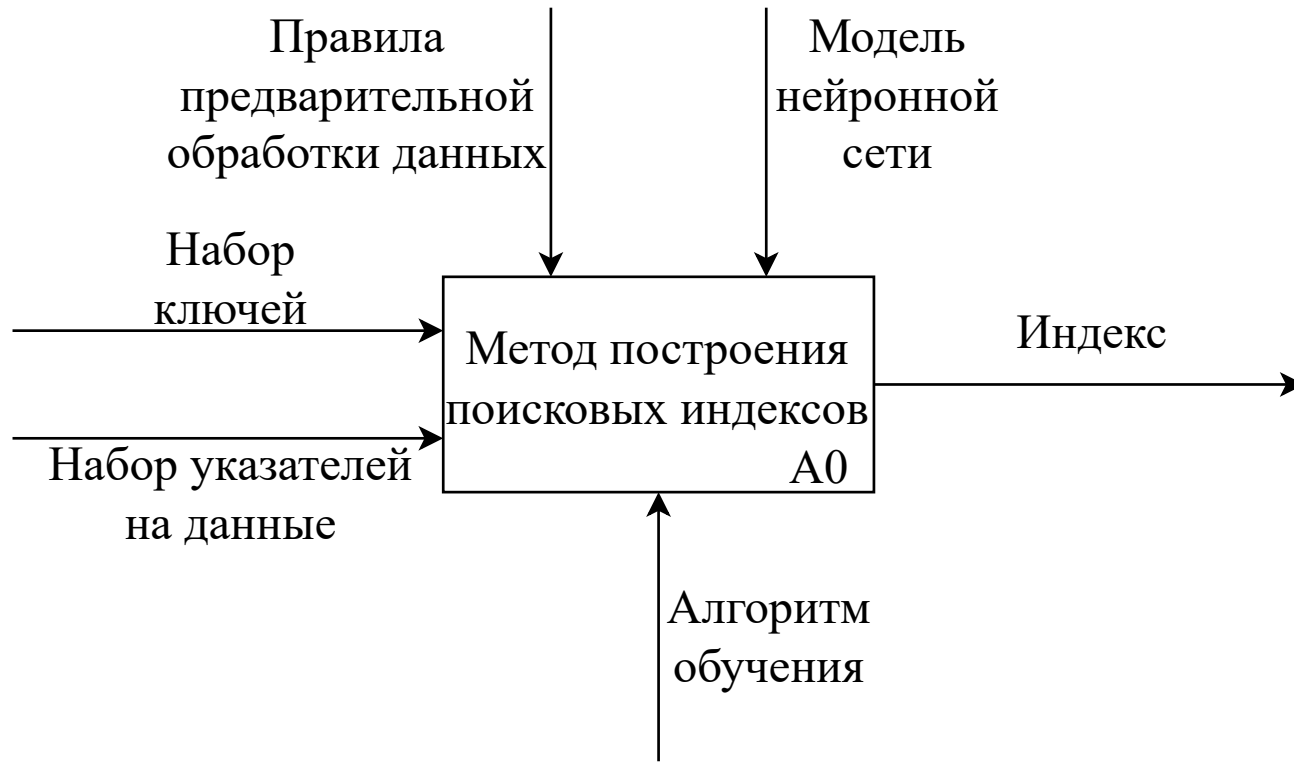
(*) — вставка в обученный индекс требует переобучения, сложность которого зависит от архитектуры используемой модели машинного обучения.

Таким образом, замена классических структур на модель машинного обучения может позволить в лучшем случае осуществлять поиск со сложностью $O(1)$ и в среднем и худшем — $O(N)$ при необходимости просмотра диапазона

При этом обученные индексы применяются ко всем подзадачам поиска и затрачивают постоянное количество памяти на поддержание структуры. Но имеют недостаток в виде необходимости переобучения модели при вставке.

Подробное сравнение с классическими индексами Вы можете видеть на слайде.

Постановка задачи



Ограничение: ключи — целые уникальные числа.

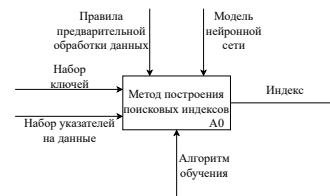
Постановка задачи

Для построения обученного индекса из реляционной базы данных получают набор ключей и соответствующих указателей на записи таблицы, подающихся на вход разработанного метода. На ключи накладывается ограничение по целочисленности и уникальности.

Результатом работы метода является индекс, представляющий собой структуру, состоящей из:

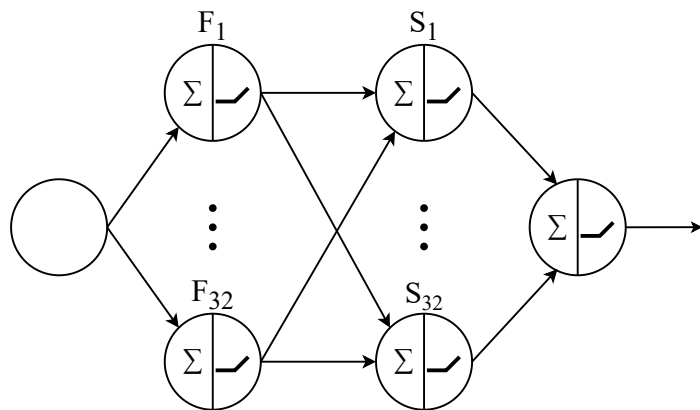
- обученной модели глубокой нейронной сети;
- массивов ключей и указателей;
- абсолютной ошибки, задающей диапазон поиска.

Постановка задачи



Ограничение: ключи — целые уникальные числа.

Архитектура нейронной сети

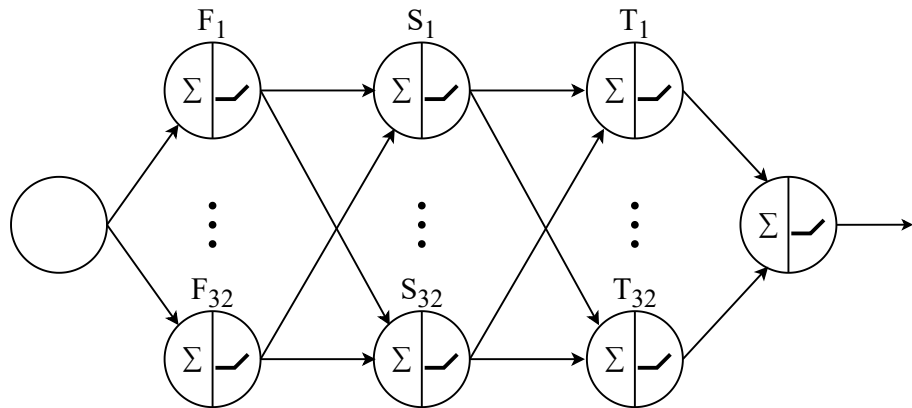


1 вход: нормализованный ключ K .
 1 выход: значение функции распределения $F(K)$.
 2-3 скрытых слоя по 32 нейрона.
 Функция активации: ReLU.

Определение значения функции распределения по позиции ключа:

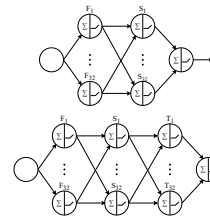
$$F(K) = \frac{p}{N},$$

где p — искомая позиция;
 K — ключ поиска;
 $F(K)$ — функция распределения;
 N — количество ключей.



Архитектура нейронной сети

Архитектура нейронной сети



1 вход: нормализованный ключ K .
1 выход: значение функции распределения $F(K)$.
2-3 скрытых слоя по 32 нейрона.
Функция активации: ReLU.

Определение значения функции распределения по позиции ключа:

$$F(K) = \frac{p}{N},$$

где p — искомая позиция;
 K — ключ поиска;
 $F(K)$ — функция распределения;
 N — количество ключей.

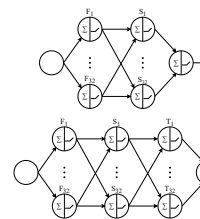
Для работы метода также должны быть заданы правила предварительной обработки, определяющиеся архитектурой глубокой нейронной сети, принимаемой в качестве модели обученного индекса.

На основе предыдущих исследований для моделей нейронной сети были выбраны параметры, которые представлены на слайде.

Ключевым моментом для обучения нейронной сети является нормализация входных и выходных значений, на которых обучается и работает модель.

Архитектура нейронной сети

Архитектура нейронной сети



1 вход: нормализованный ключ K .
 1 выход: значение функции распределения $F(K)$.
 2-3 скрытых слоя по 32 нейрона.
 Функция активации: ReLU.

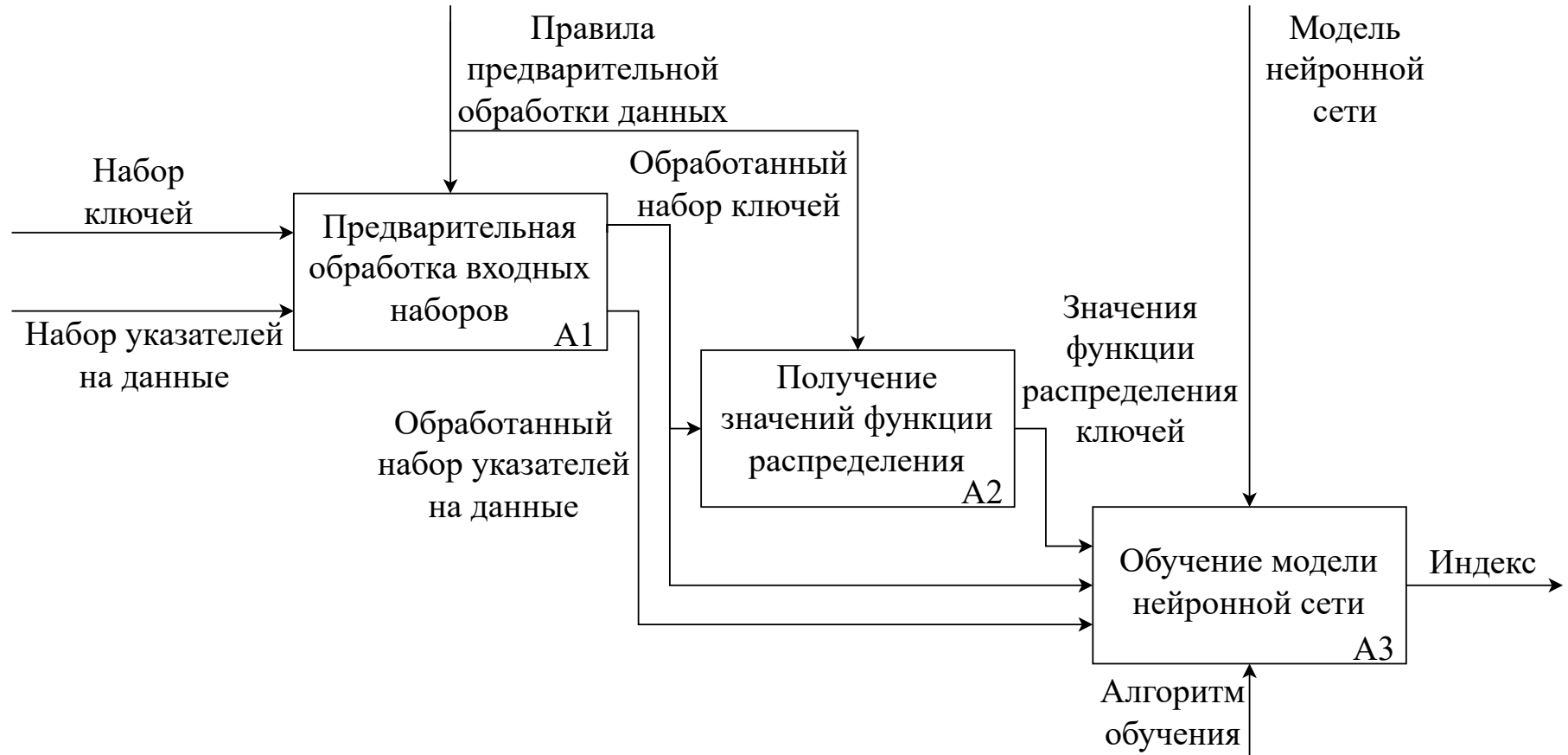
Определение значения функции распределения по позиции ключа:

$$F(K) = \frac{p}{N},$$

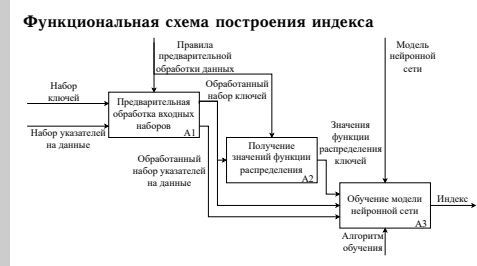
где p — искомая позиция;
 K — ключ поиска;
 $F(K)$ — функция распределения;
 N — количество ключей.

Значение ключа отображается в диапазон $[0, 1]$ с помощью минимакс нормализации. Выходные значения для обучения нормализуются путем вычисления эмпирической функции распределения ключей по их позициям в отсортированном массиве по формуле на слайде.

Функциональная схема построения индекса



└ Функциональная схема построения индекса

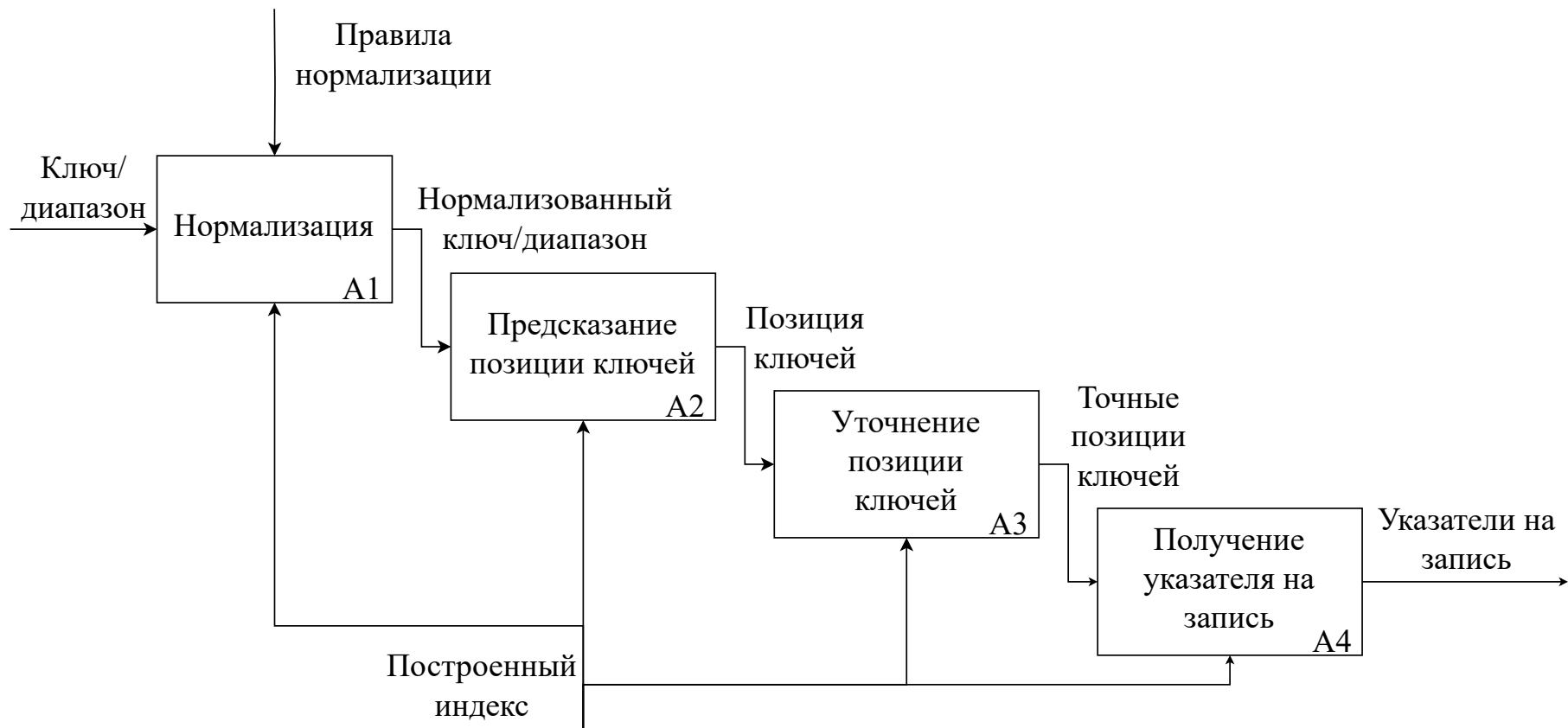


Сортировка набора ключей дает также возможность диапазонного поиска.

Поэтому на этапе метода предварительной обработки происходит согласованная сортировка массивов ключей и указателей с нормализацией ключей. На втором этапе вычисляются значения функции распределения.

Результаты первого и второго этапа передаются на вход этапа обучения модели глубокой нейронной сети, выдающий на выходе необходимый индекс.

Функциональная схема поиска



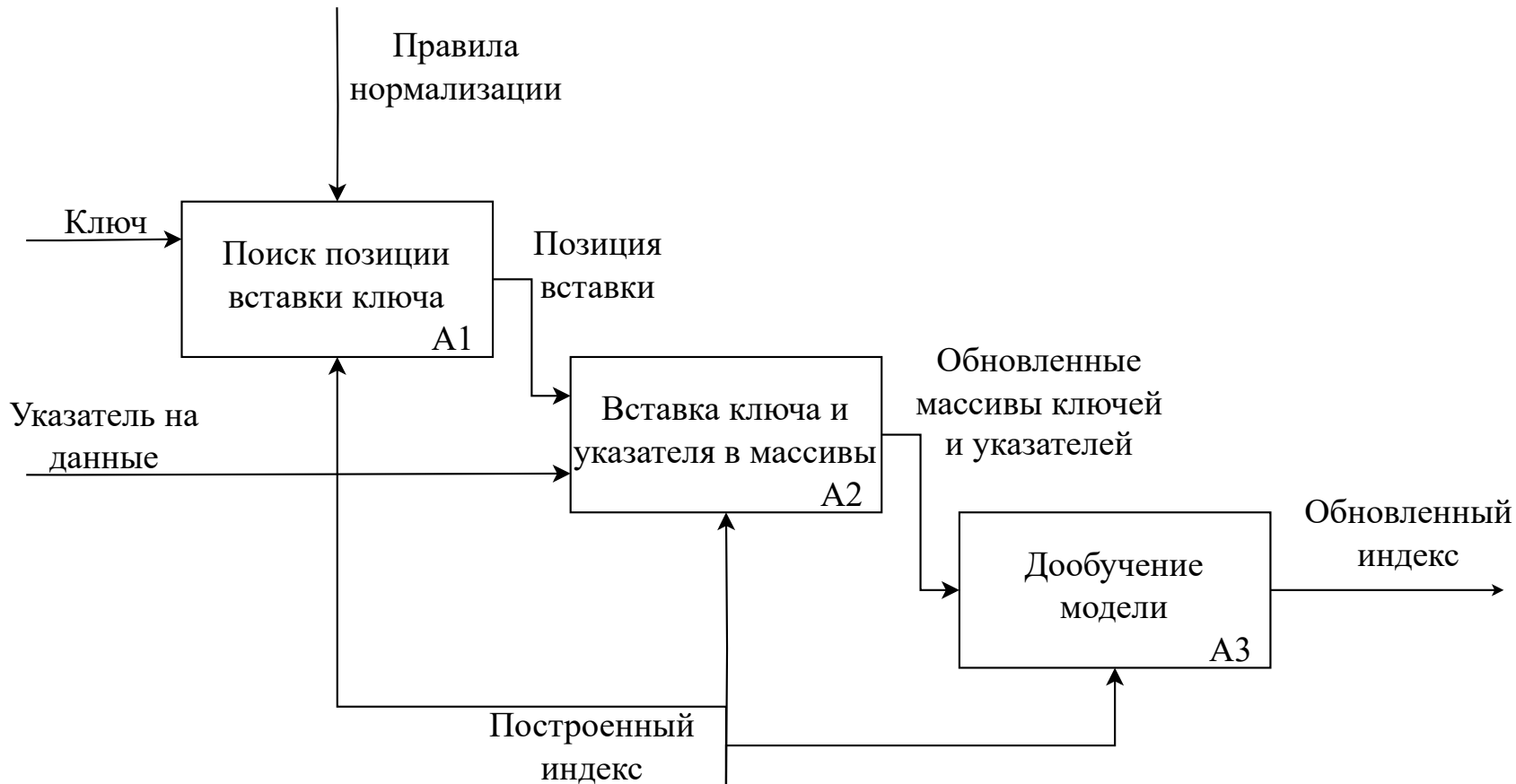
└ Функциональная схема поиска



Для оценки работы метода также разработаны алгоритмы поиска и вставки в построенном индексе.

Поиск включает в себя тот же этап нормализации ключа или границ диапазона ключей, этап получения предсказанной нейронной сетью позиции, этап ее уточнения, который осуществляется бинарным поиском с учетом отсортированности массива, и этап получения соответствующего уточненной позиции ключа указателя на данные.

Функциональная схема вставки

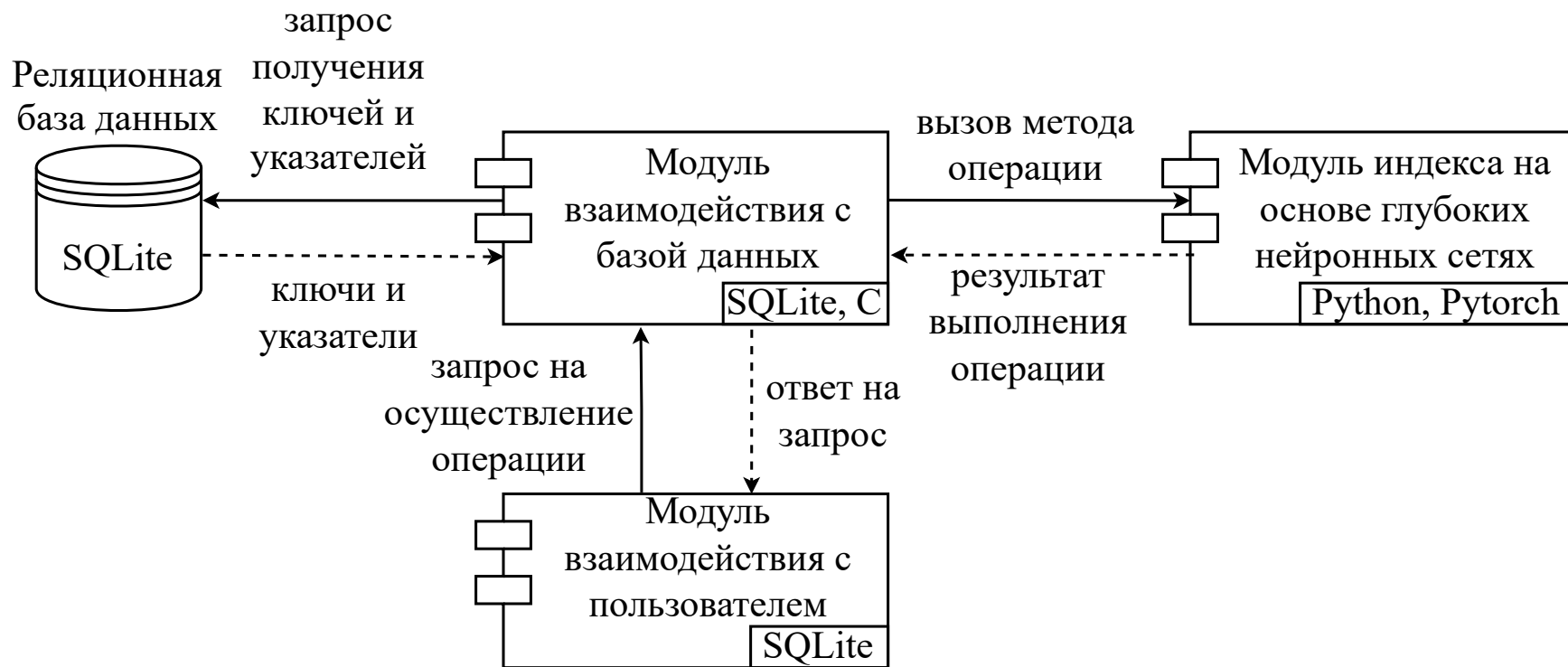


Функциональная схема вставки



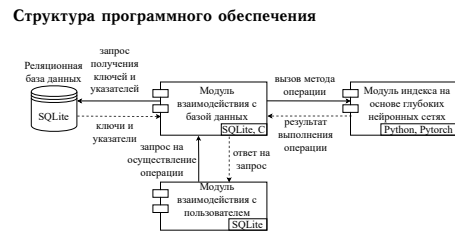
Вставка представляет собой комбинацию алгоритмов поиска и построения, то есть включает этап получения позиции для вставки только что описанным алгоритмом без получения указателя на данные, этап изменения массивов и этап обучения модели на новых данных, начиная с уже рассчитанных весов модели.

Структура программного обеспечения

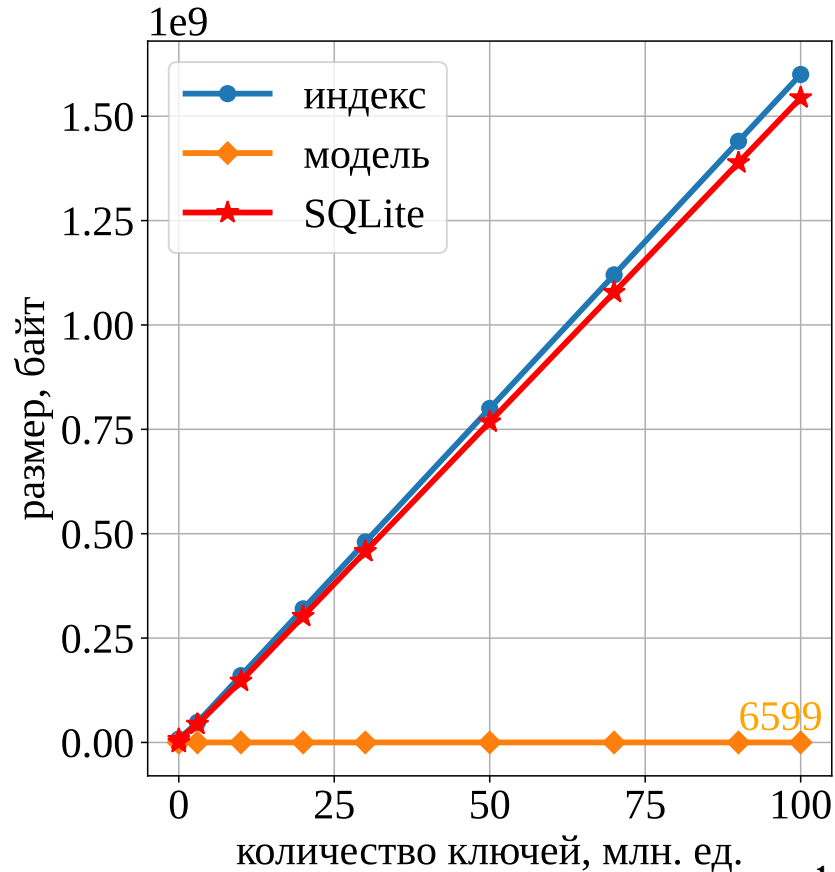
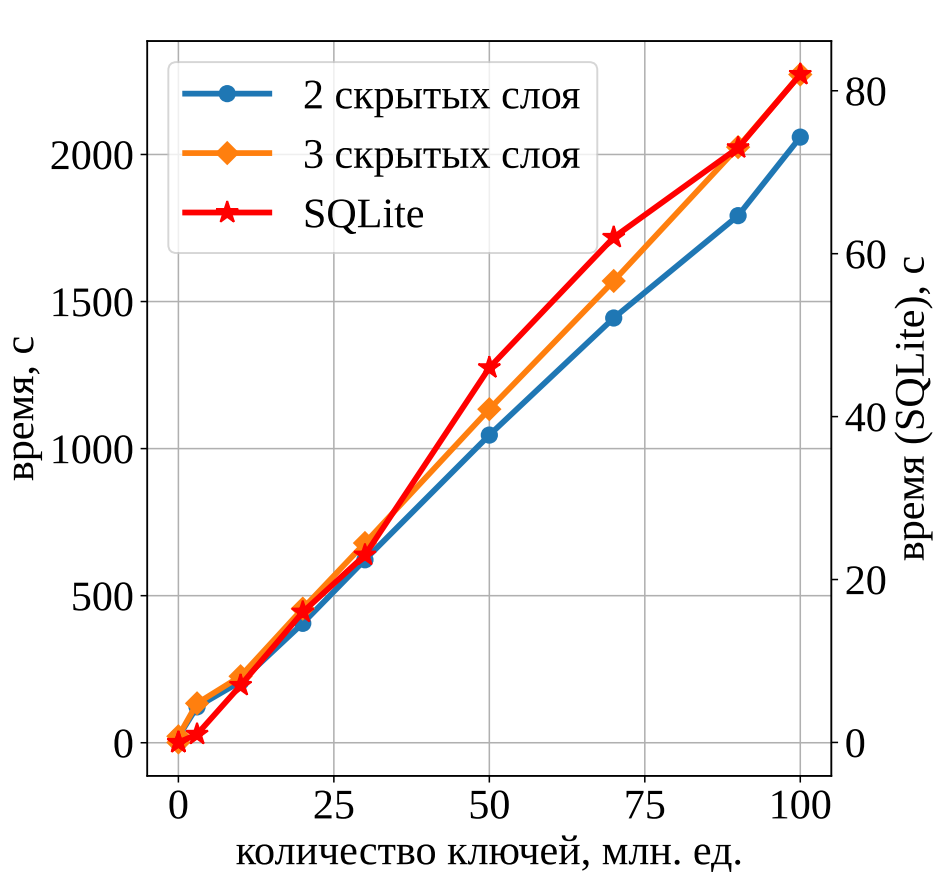


└ Структура программного обеспечения

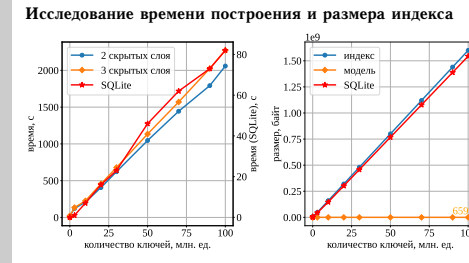
Для реализации метода поистроения индекса было разработано программное обеспечения, включающее модуль взаимодействия с реляционной базой данных, в качестве которой была выбрана SQLite, модуль реализации индекса на основе глубоких нейронных сетей и готовый модуль взаимодействия с пользователем, представленный интерфейсом SQLite.



Исследование времени построения и размера индекса



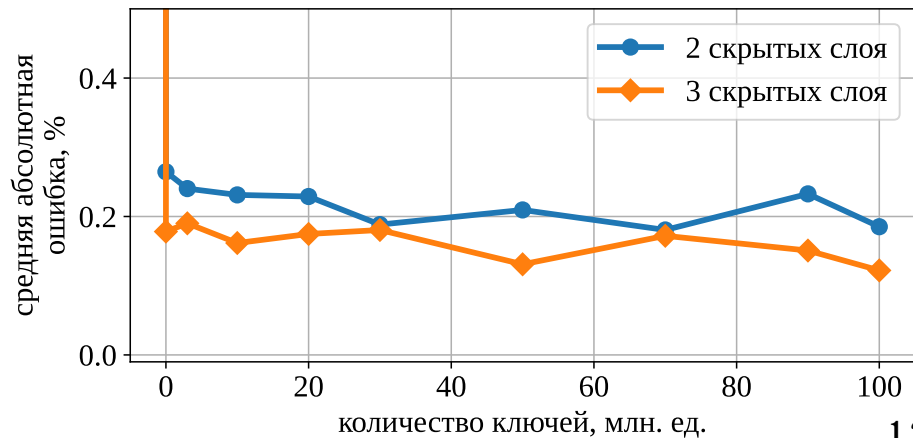
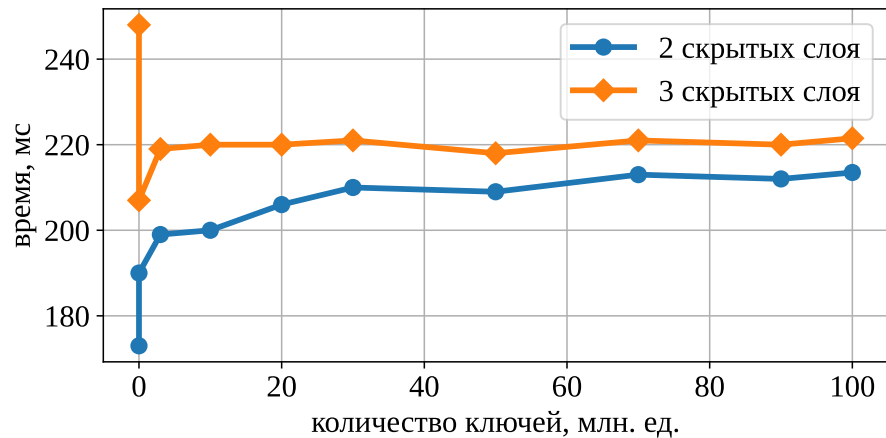
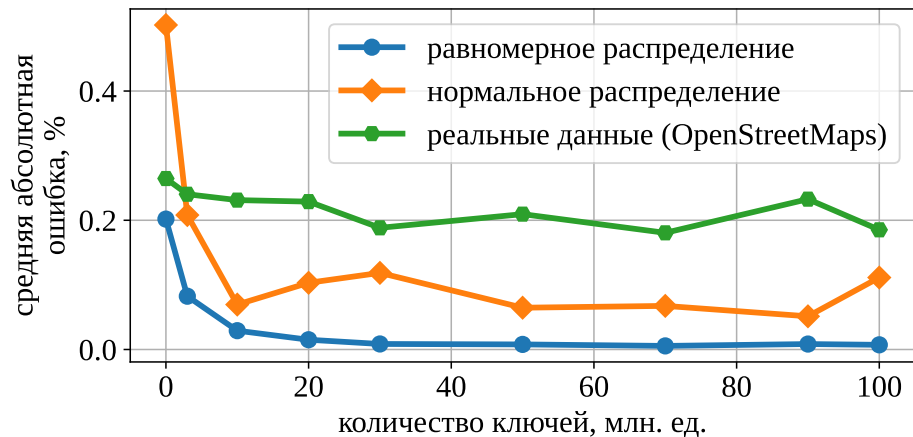
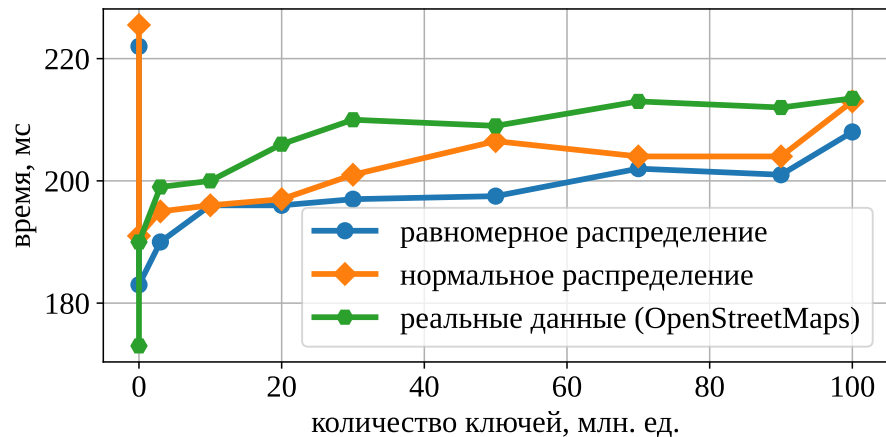
Исследование времени построения и размера



С использованием разработанного ПО было проведено исследование разработанного индекса в сравнении с реализацией В-дерева в SQLite, на моделях с 2 и 3 слоями, на трех различных распределениях ключей. По результатам исследования выявлена:

- линейная зависимость времени построения от количества ключей у обоих типов индексов, с увеличением времени построения на 10% при добавлении дополнительного слоя модели;
- линейный рост размеров обоих индексов при увеличении числа ключей.

Исследование времени поиска



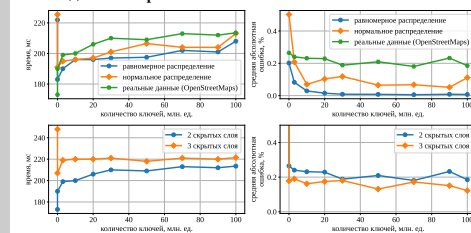
Исследование времени поиска

Также выявлено стремление отношения средней абсолютной ошибки модели для каждого распределения к некоторой постоянной, что обеспечивает логарифмическую сложность поиска.

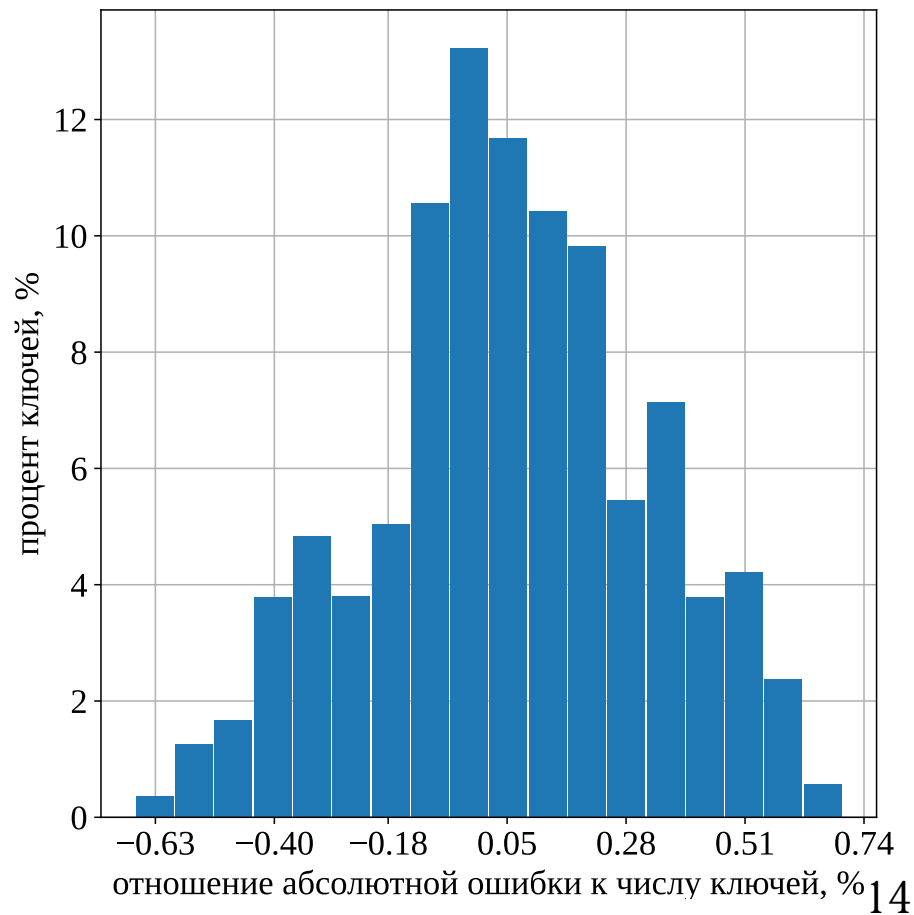
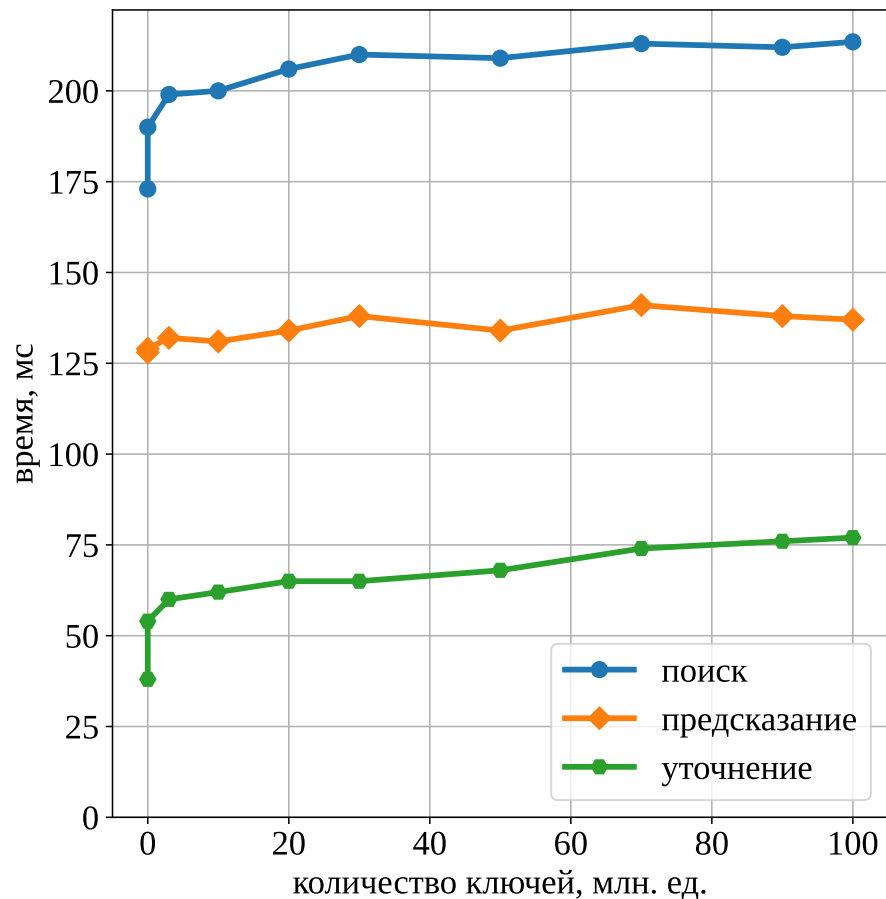
Выявлена меньшее значение ошибки на классических распределениях и как следствие меньшее время поиска.

На представленных графиках отображено, что добавление третьего слоя к модели нейронной сети необавдано, так как сокращение времени бинарного поиска в меньшем диапазоне за счет большей точность перекрылось затратами на вычисления дополнительных коэффициентов

Исследование времени поиска

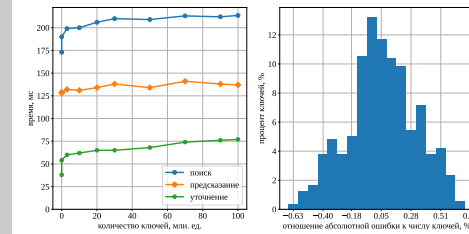


Исследование времени поиска



Исследование времени поиска

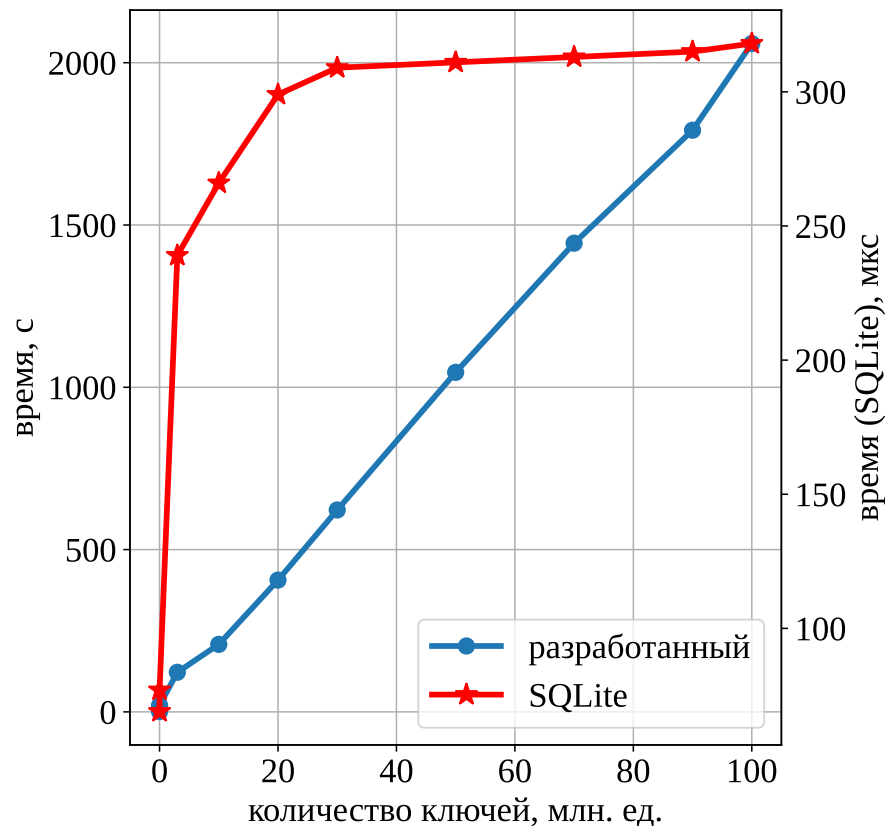
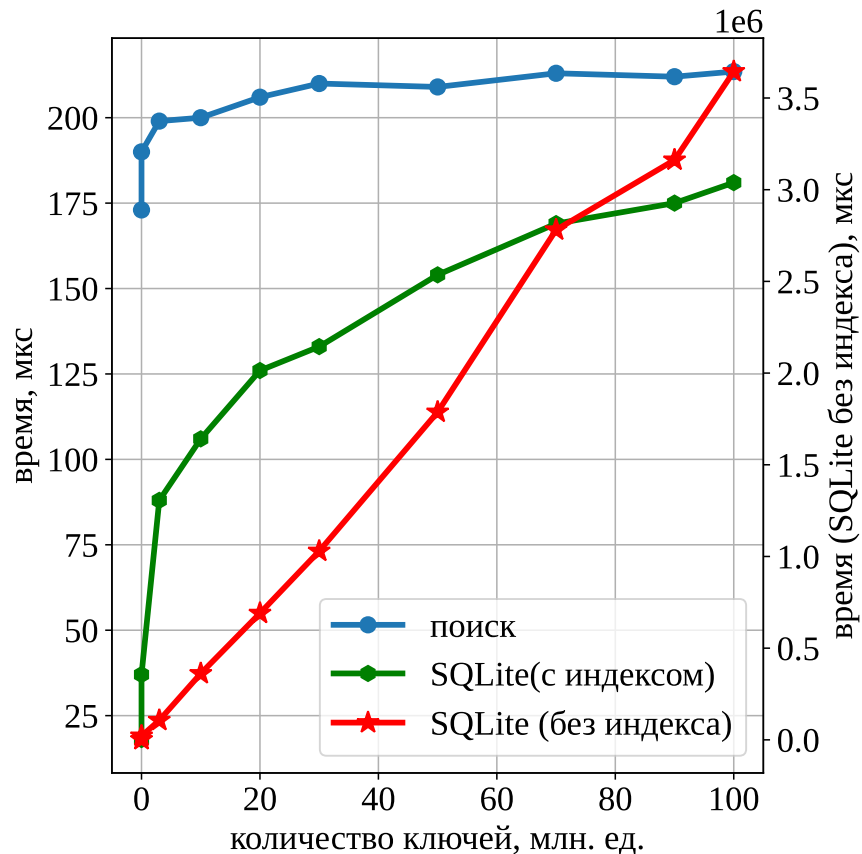
Исследование времени поиска



Исследование поиска по этапам дало постоянное время предсказания позиции, логарифмическую зависимость времени бинарного поиска от числа ключей, и как следствие логарифмическую сложность поиска с помощью разработанного индекса.

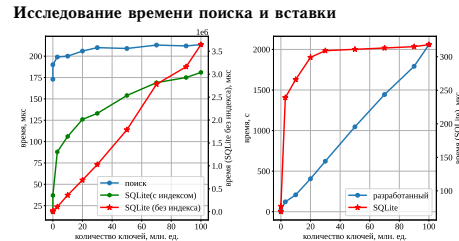
При этом линейный рост размера диапазона осуществления бинарного поиска происходит с коэффициентов, лежащим пределах от 0 до 0.6%, что представлено на гистограмме.

Исследование времени поиска и вставки



Исследование времени поиска и вставки

Исследование времени поиска и вставки разработанного индекса в сравнении с индексом SQLite показало одинаковые порядки значений времени поиска при логарифмических зависимостях у обоих индексов, и превышение в десятки раз времени вставки в разработанный индекс времени индекса SQLite, при линейном и логарифмическом росте соответственно.



Заключение

В ходе данной работы:

- проанализированы известные методы построения индексов;
- приведено описание построения индексов с помощью нейронных сетей;
- разработан метод построения индексов в реляционной базе данных на основе глубоких нейронных сетей;
- разработано программное обеспечение, реализующее данный метод;
- проведено исследование (по времени и памяти) операций поиска и вставки с использованием индекса, построенного разработанным методом, при различных объемах данных.

Поставленная цель достигнута.

Дальнейшее развитие

1. Оптимизация алгоритма вставки с учетом распределения ключей.
2. Добавление возможности построения индекса по ключам других типов данных.
3. Построение многомерных обученных индексов.