Санкт-Петербургский государственный университет
Кафедра системного программирования

Разработка инфраструктуры для обучения с учителем искусственной нейронной сети, выбирающей пути при символьном исполнении

Анна Артуровна Чистякова, группа 21.Б09-мм

Научный руководитель: к.ф.-м.н. С. В. Григорьев, доцент кафедры системного программирования

Санкт-Петербург
2024

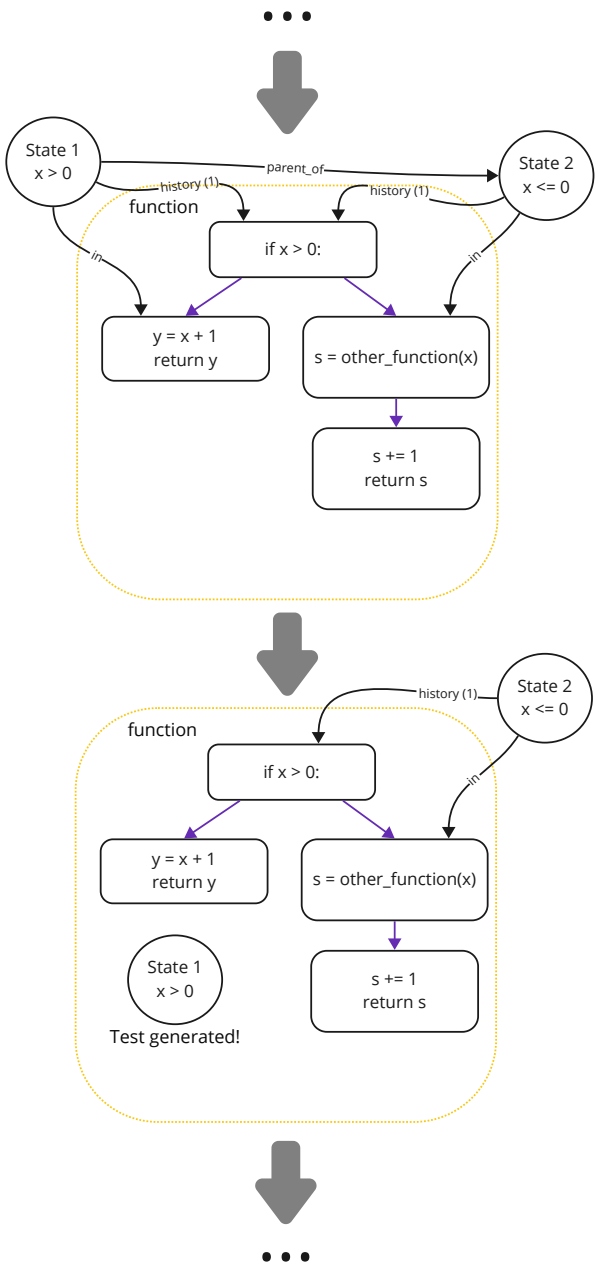
Введение

- Растет необходимость тестировать сложные системы
- Символьное исполнение помогает не пропустить входы, на которых программа работает неправильно
- Проблема символьного исполнения — большое количество состояний

Символьное исполнение

```
def function(x: int):
    if x > 0:
        y = x + 1
        return y
    else:
        s = other_function(x)
        s += 1
        return s

def other_function(x: int):
    if x == 5:
        return 0
    else:
        return x
```



Существующие решения

- Алгоритмы на графах (DFS, BFS)
- Различные эвристики (например, присваивание вероятности выбора пути на основе текущих свойств состояния)
- Подходы, основанные на машинном обучении
  - Learch[1] — линейная модель
  - Q-learning[2]

[1] "Learning to explore paths for symbolic execution", Jingxuan He, Gishor Sivanrupan, Petar Tsankov, Martin Vechev, 2021

[2] "Reinforcement learning guided symbolic execution", Wu Jie, Zhang Chengyu, Pu Geguang, 2020

Описание проекта

- Клиент
  - Предобработка данных
  - Обучение
  - Валидация
- Сервер
  - Символьное исполнение

Постановка задачи

Целью работы является исследование применимости графовых нейронных сетей к выбору путей при символьном исполнении
Задачи:

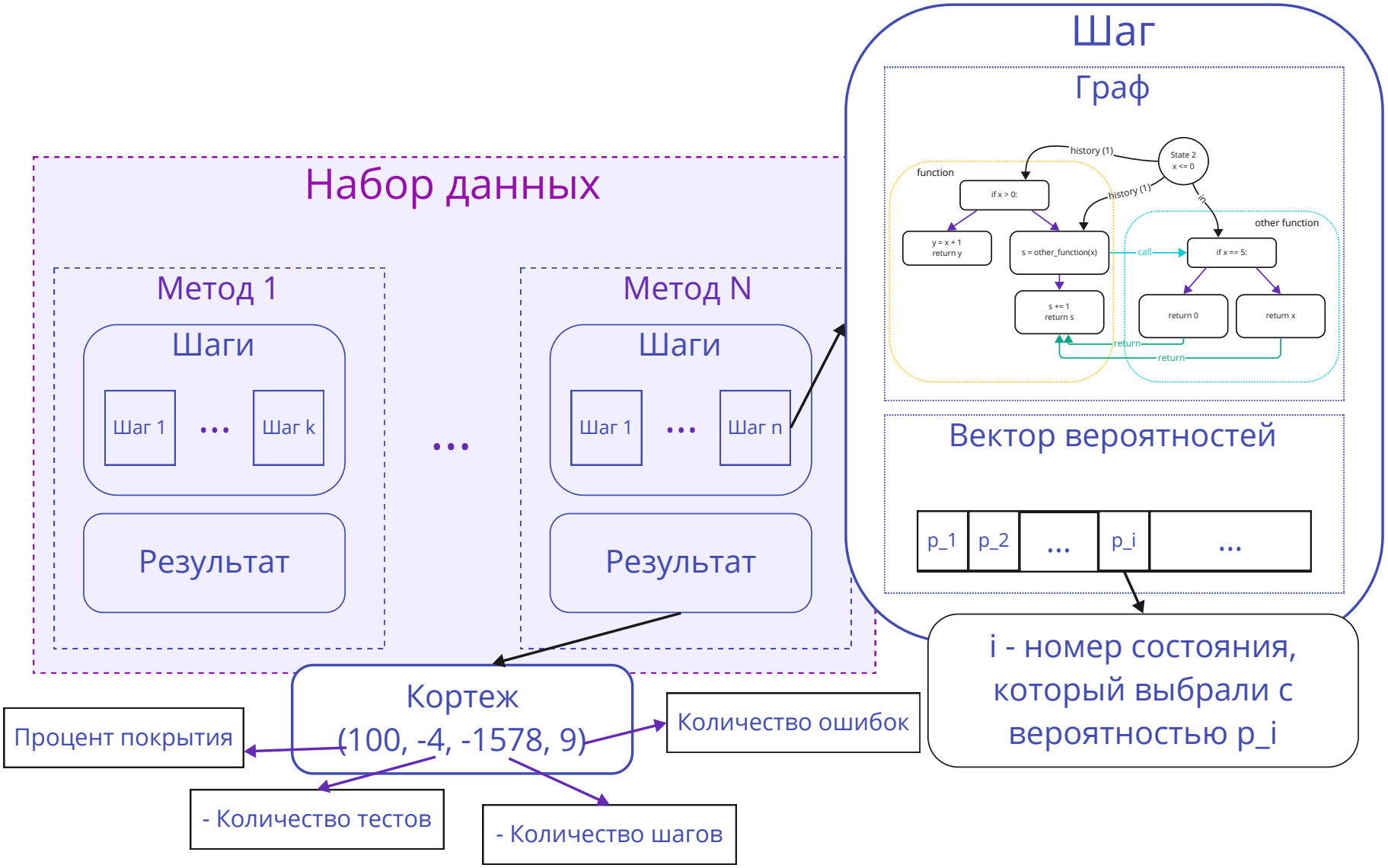
- Разработка инфраструктуры для обучения с учителем графовой нейронной сети
- Проведение экспериментов с различными гиперпараметрами с целью достижения максимально возможного процента тестового покрытия

Используемые инструменты

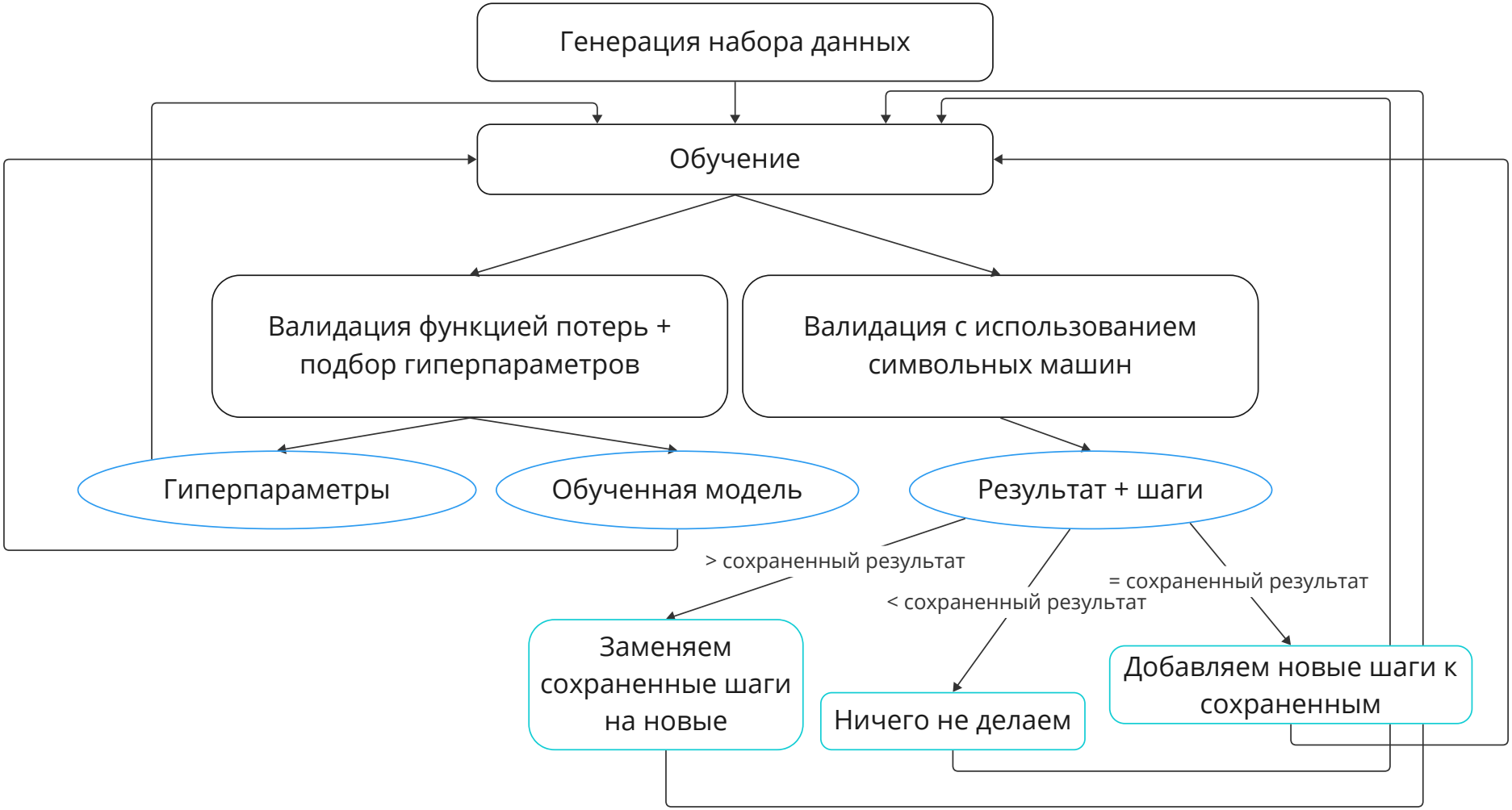
- Библиотека PyTorch Geometric
- Библиотека optuna для подбора гиперпараметров
- VSharp[3] в качестве символьной машины

[3] https://github.com/VSharp-team/VSharp (дата доступа: 2024-01-07)

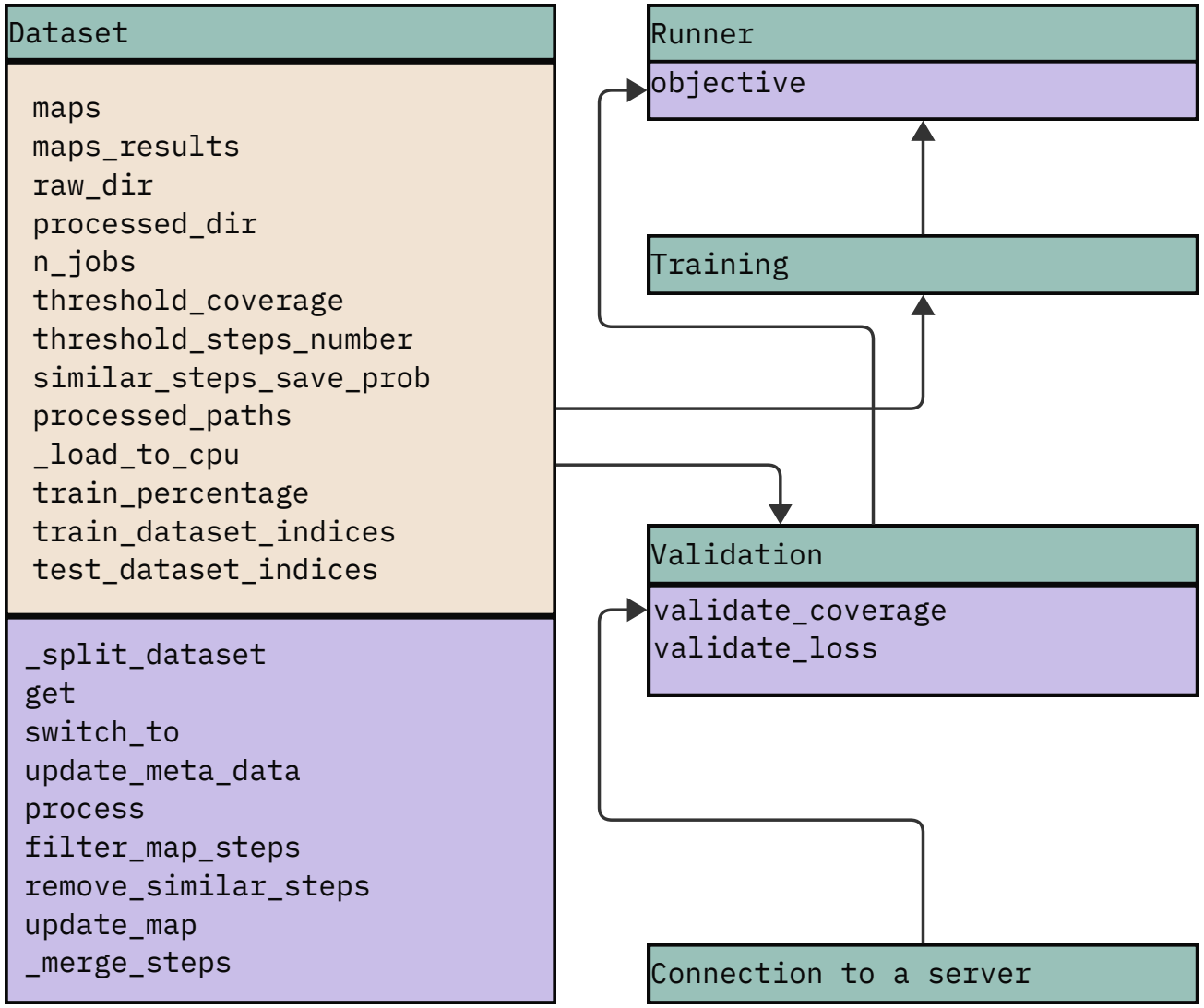
Набор данных



Описание процесса обучения



Архитектура

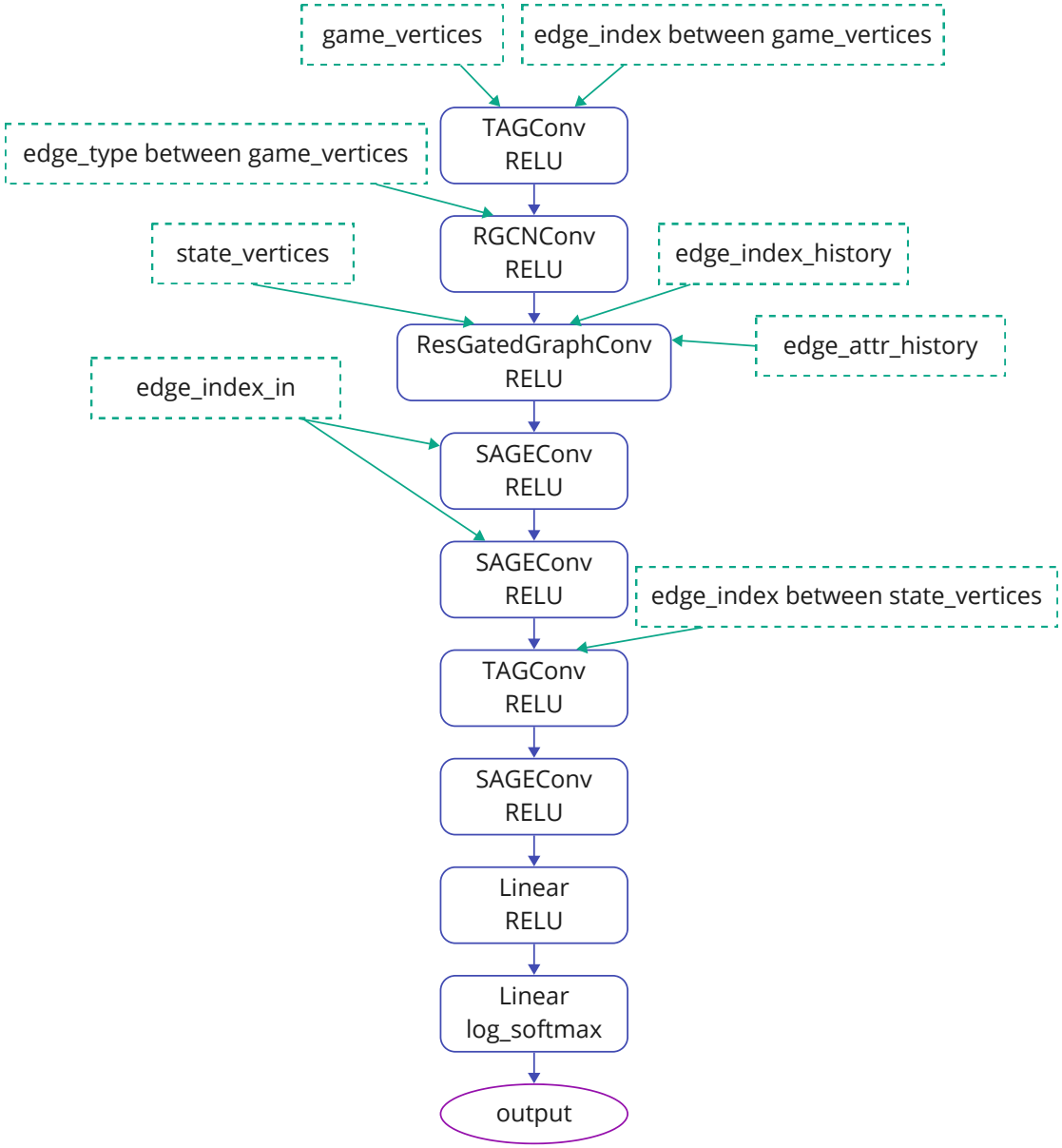


Эксперименты

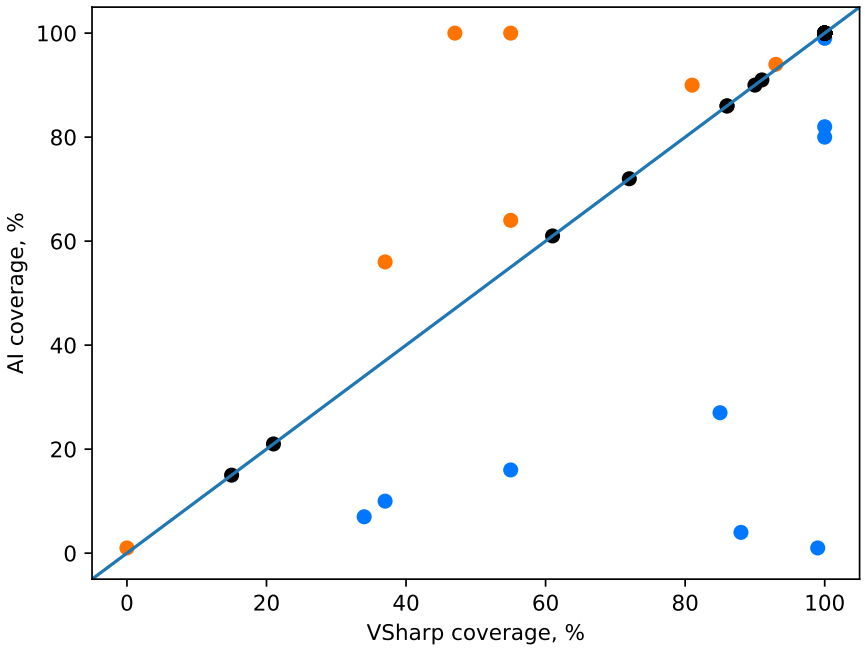
- Набор данных — 507 методов
- Функция потерь — расстояние Кульбака-Лейблера
- Оптимизатор — Adam
- Алгоритм для подбора гиперпараметров — Tree-Structured Parzen Estimator

Эксперименты (подбор гиперпараметров)

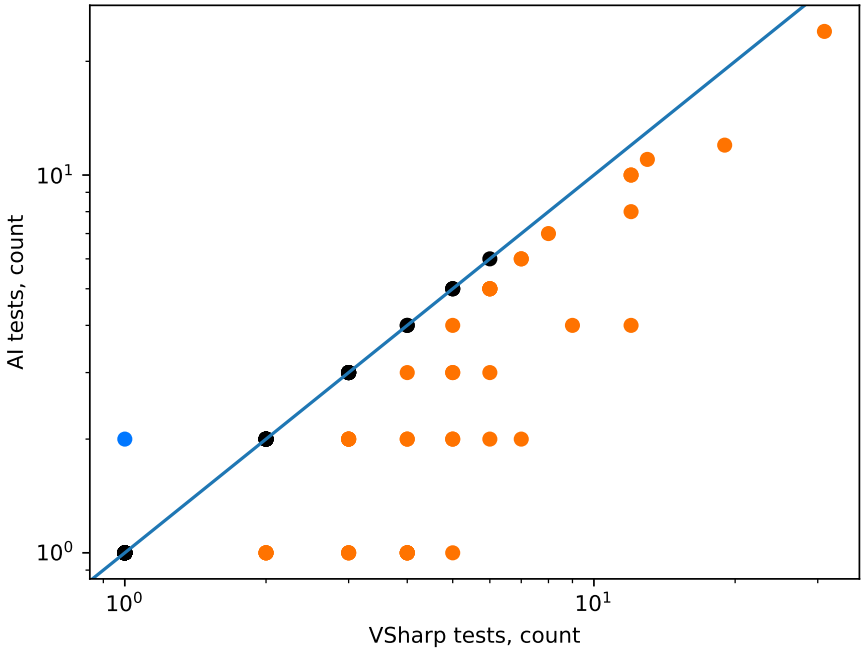
- $lr = 0.0003$
- $batch_size = 109$
- $num_hops_1 = 5$
- $num_hops_2 = 4$
- $num_of_features = 30$
- $hidden_channels = 110$



Результаты



AI (оранжевый) лучше: 7, VSharp (синий) лучше: 9,
одинаковый результат (черный): 113



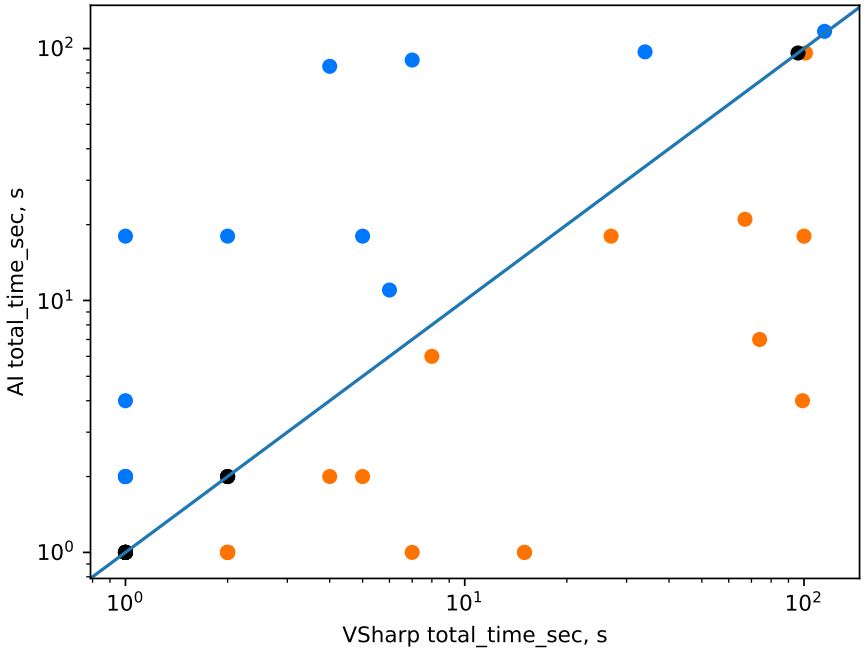
AI (оранжевый) лучше: 43, VSharp (синий) лучше: 1,
одинаковый результат (черный): 69

Результаты

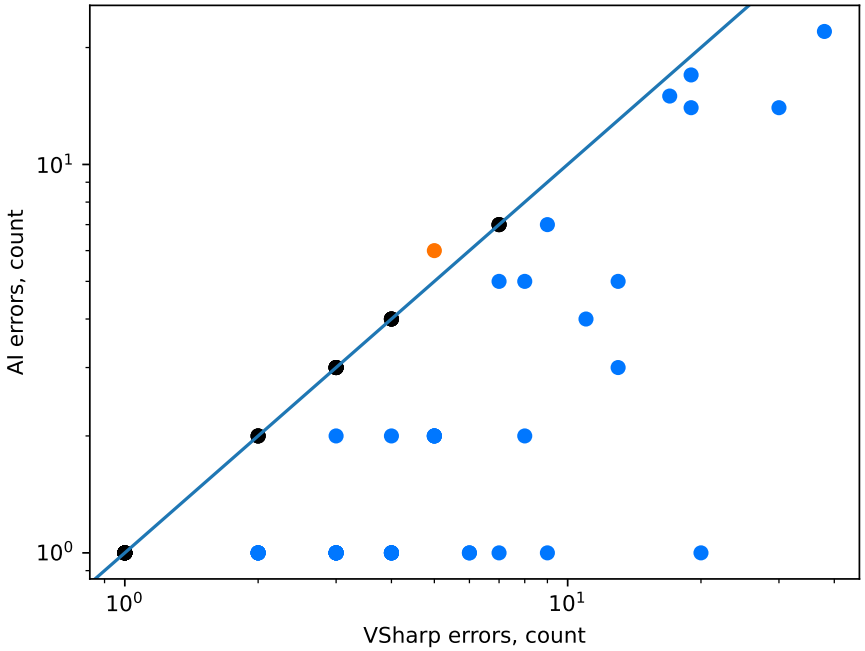
- Разработана инфраструктура для обучения с учителем графовой нейронной сети[4]
- Проведены эксперименты и проанализированы результаты
- Обучена модель, которая была интегрирована в V#

[4] `https://github.com/gsvgit/PySymGym/tree/dev` — ссылка на репозиторий проекта (дата доступа: 2024-06-06). Имя пользователя Anya497

Еще результаты



AI (оранжевый) лучше: 26, VSharp (синий) лучше: 23,
одинаковый результат (черный): 64



AI (оранжевый) лучше: 1, VSharp (синий) лучше: 50,
одинаковый результат (черный): 62