

«Нейронные сети в моей публикации»

Биология, химия, медицина

Курсовая работа

*Multi-stage model for predicting retention times of metabolites in liquid chromatography using deep learning methods*

*Многослойная модель для предсказания времен удерживания метаболитов в условиях жидкостной хроматографии при помощи методов машинного обучения*

*Автор:* Строева Екатерина Николаевна (ВМК МГУ имени М.В. Ломоносова, <https://istina.msu.ru/workers/396763216/>)

*Научный консультант на курсе MSU.AI:* Немченко Виктор

*Научный консультант:* Матюшин Дмитрий Дмитриевич  
(ФГБУН Институт физической химии и электрохимии им. А.Н. Фрумкина РАН, <https://istina.msu.ru/workers/96443959/>)

# Введение

Предсказание времен удерживания в условиях жидкостной хроматографии является важной задачей, поскольку точное предсказание времени удерживания поможет исследователям и аналитикам оптимизировать условия проведения химических опытов и получения достоверных результатов, улучшить разделение соединений в смесях и ускорить процесс определения состава образцов. Это позволяет более эффективно проводить аналитические исследования в различных областях, включая фармацевтику, пищевую промышленность, окружающую среду и другие.

Точные предсказания времени удерживания также могут помочь в улучшении качества аналитических данных и повышении точности и надежности результатов исследований, а также сократить затраты на проводимые эксперименты.

Времена удерживания сильно зависят от конкретных экспериментальных условий (состав элюента, тип хроматографической колонки и прочие), однако для данных конкретных условий задача сводится к предсказанию одного единственного числа по структуре молекулы.

Существует ряд публично доступных наборов данных по временам удерживания метаболитов в условиях жидкостной хроматографии. Внутри одного набора данных все значения обычно получены при одних и тех же условиях. Однако, для разных наборов данных эти условия различаются. Причём во всех доступных наборах данных нет информации об условиях проведения экспериментов, от которых больше всего зависит результат.

Возникает проблема качественного предсказания времени удерживания для набора данных с небольшим количеством молекул, содержащим до 20 записей, вне зависимости от условий проведения наблюдений.

## Цель

Создать модель, основанную на методах машинного обучения, способную предсказывать времена удерживания для произвольных условий для небольшого набора данных (5-20 записей) для фиксированных заданных условий.

## Обзор существующих решений

В статье [1] рассматривается трёх-слойная схема, изображённая на рисунке 1:

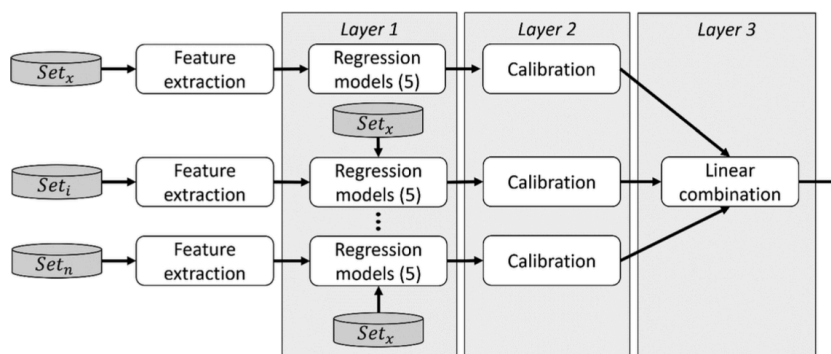


Рис 1. Блок-схема модели CALLC в статье [1]

Для первого этапа предсказаний используется комбинация из алгоритмов классического машинного обучения: XGBoost (GB), Support Vector Regression (SVR), Least Absolute Shrinkage and Selection Operator (LASSO), Adaptive Boosting (AB), и Bayesian Ridge Regression (BRR). Полученные предсказания подаются как вход для калибровочных функций на слой 2, и затем на слое 3 используется модель эластичной сети для получения регуляризованной линейной комбинации калиброванных прогнозов, полученных на основе разных экспериментальных установок и алгоритмов для одного и того же анализита. Архитектура имеет такие недостатки, как использование относительно небольшого набора данных и отсутствие глубоких нейронных сетей.

Идеи получили своё развитие в статье [2], где модели классического машинного обучения были заменены на полносвязные и свёрточные слои, что дало небольшой прирост точности.

Авторы работы [3] использовали графовую нейронную сеть (Graph Neural Network, GNN) и провели сравнительный анализ реализованной модели с такими алгоритмами, как Random Forest (RF), Bayesian Ridge Regression (BRR), а также простыми свёрточными и полносвязными искусственными нейронными сетями. В статье показано, что GNN показывает лучшую точность, и эту архитектуру можно использовать в качестве первого этапа (рис. 2) для разрабатываемой в рамках данного исследования архитектуры.

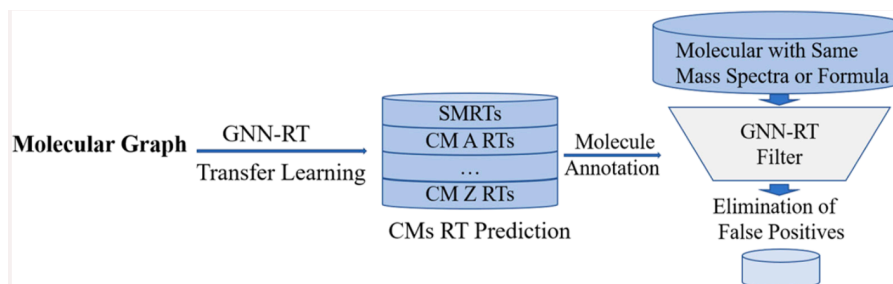


Рис 2. Блок-схема модели GNN-RT в статье [3]

В статье [4] достигнута очень хорошая точность первой ступени, основанной на GNN (рис. 3), но к недостаткам стоит отнести небольшой набор данных, используемый для обучения и тестирования модели.

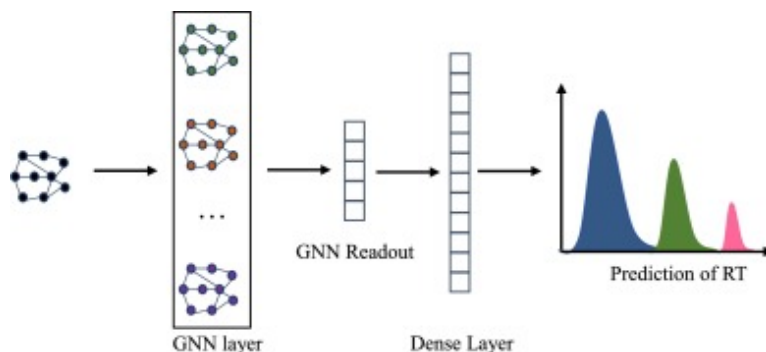


Рис 3. Блок-схема модели в статье [4]

В статье [5] вместо графовой нейронной сети используются архитектуры энкодера и декодера, что так же даёт улучшение в точности (рис. 4).

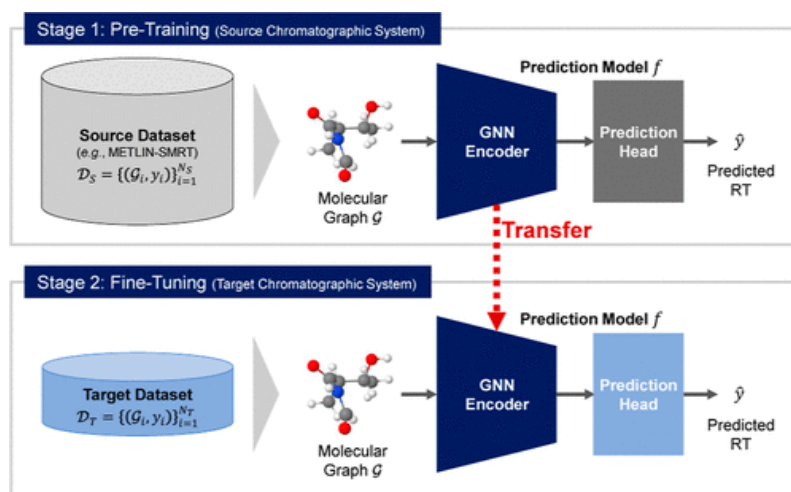


Рис 4. Блок-схема модели GNN-RT в статье [5]

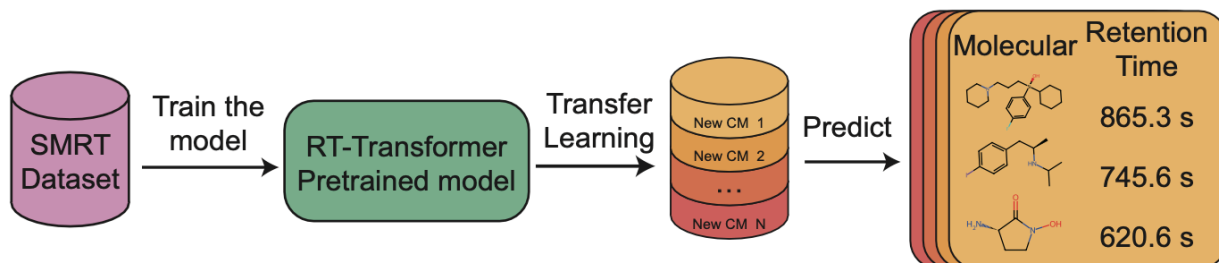


Рис 5. Блок-схема модели GNN-RT в статье [6]

Развитием является применение архитектуры трансформера, описанное в статье [6] (рис. 5), который был обучен на самом большом наборе данных METLIN SMRT на 80 038 молекул и многочисленные оценки показывают, что RT-Transformer обладает превосходной

масштабируемостью при прогнозировании времени удерживания для жидкостной хроматографии и повышает точность идентификации метаболитов.

Наконец, в статье [7] реализована уже трёх-ступенчатая схема, включающая искусственные нейронные сети, содержащие свёрточные и полносвязный слои (рис. 6), но задача описанного в работе исследования состоит в предсказании газовой хроматографии, можно взять её за основу для предсказаний времени удерживания в жидкостной хроматографии.

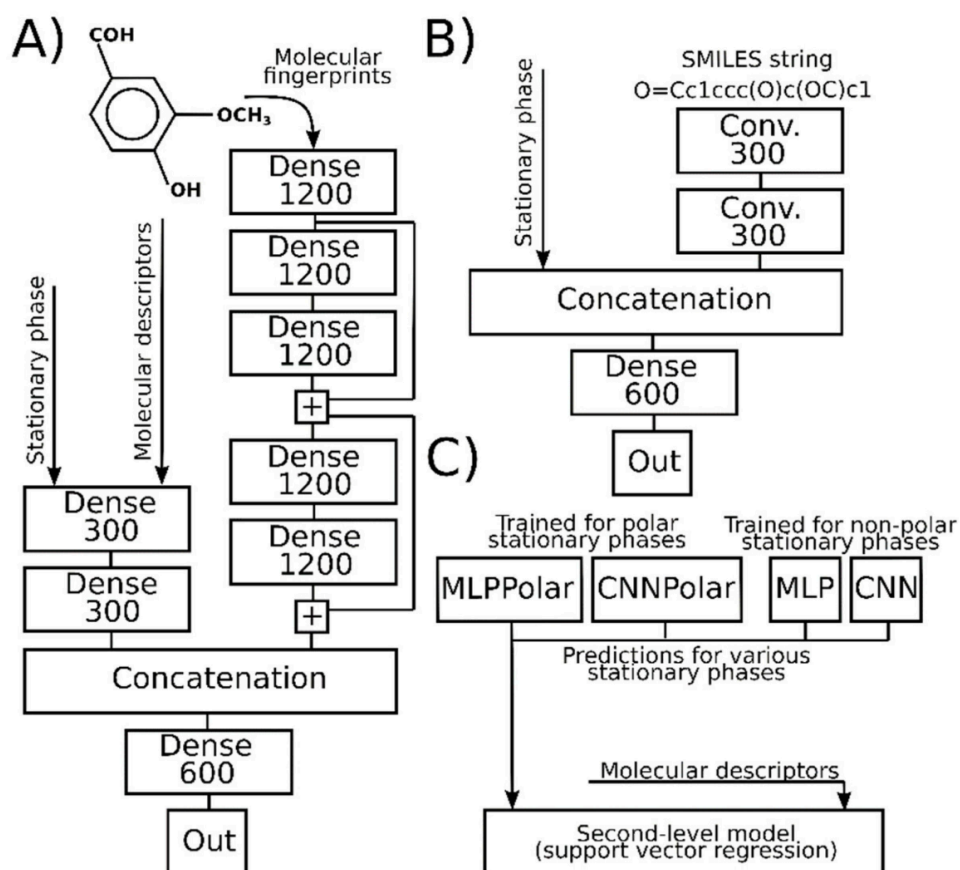


Рис 6. Блок-схема модели в статье [7]

## Методология и архитектура предполагаемой модели:

Предполагаемая модель будет состоять из трех или четырех ступеней:

- первая ступень будет обучена на самом большом имеющемся наборе данных (METLIN SMRT, более 80 тысяч записей) и представлять собой графовую нейронную сеть или ансамбль искусственных нейронных сетей;
- модели второго уровня предполагается обучить на нескольких наборах данных, содержащих в себе 100-2000 молекул. Для их обучения будет использовано или трансферное обучение искусственных нейронных сетей, предобученных с помощью METLIN SMRT, или регрессионные модели, работающие с табличными данными, одной из характеристик в которых будет время удерживания предсказанное для набора данных METLIN SMRT;
- финальное обучение и предсказание для целевой системы будет выполняться или с помощью модели 3-его уровня, связывающая предсказания, полученные при помощи моделей предыдущего уровня, с наблюдаемыми значениями.

Таким образом, архитектура будет состоять из 3-х уровней, на каждом уровне будет модель машинного обучения или искусственная нейронная сеть. Основная идея заключается в том, что на каждом следующем уровне следующая модель использует все меньший набор данных для обучения, но при этом базируется на результатах модели предыдущего уровня.

# Все проведённые эксперименты

## Метрики

Во всех перечисленных статьях используется один и тот же набор метрик: mean absolute error (MAE), mean percentage error (MPE), root mean square error (RMSE), median absolute error (MdAE), median percentage error (MdPE).

## Данные

{эксперименты расположены в файле [0] Data\_preparation\_METLIN\_and\_HILIC.ipynb}

Есть три варианта набора данных:

- 1) Самый большой набор данных METLIN SMRT на 80 038 экземпляров молекул (<https://www.nature.com/articles/s41467-019-13680-7>, [https://figshare.com/articles/dataset/The\\_METLIN\\_small\\_molecule\\_dataset\\_for\\_machine\\_learning-based\\_retention\\_time\\_prediction/8038913](https://figshare.com/articles/dataset/The_METLIN_small_molecule_dataset_for_machine_learning-based_retention_time_prediction/8038913))

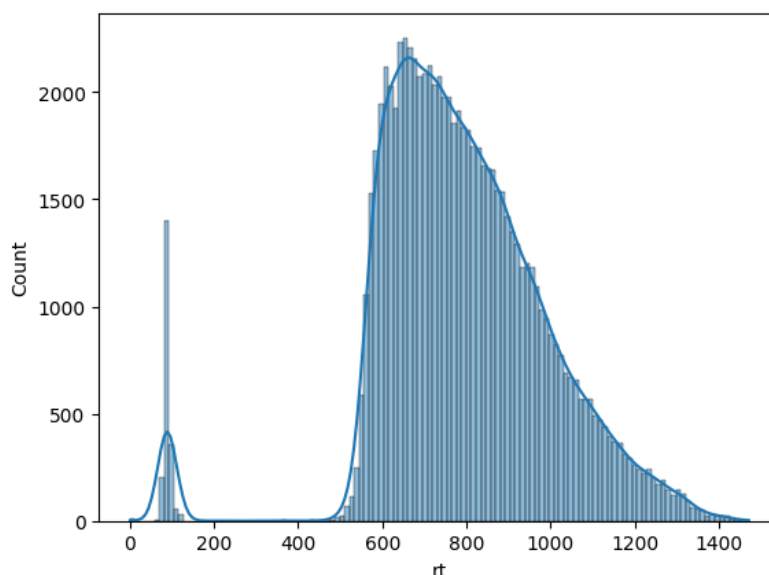


Рис 7. Распределение данных в наборе METLIN

Описание:

- inchi - необходимо перевести в SMILES
- rt дано в секундах, поэтому необходимо его перевести в минуты, так как, во-первых, нейронные сети лучше решают задачи с числами меньшего порядка, во-вторых, в других наборах данных колонка rt представлена в минутах.

Возникла проблема, что данные представляют собой бимодальное распределение (рис. 7). Поэтому было принято решение разделить этот набор данных на два других набора по значению времени удерживания: от 0 до 8 минуты (2068 молекул) от 8 до 25 минут (77889 молекул).

- 2) Набор данных HILIC (<https://pubs.acs.org/doi/abs/10.1021/acs.analchem.9b05765#>) состоит из 880 молекул, уже приведённых к виду SMILES, время удерживания от 0 до 14 минут.

3) Набор молекул hilic\_oxana, собранный экспериментально в лаборатории ИФХЭ РАН, состоящий из 18 молекул, значение времени удерживания - от 3 до 16 минут.

Совместное распределение всех наборов данных представлено на рисунке 8:

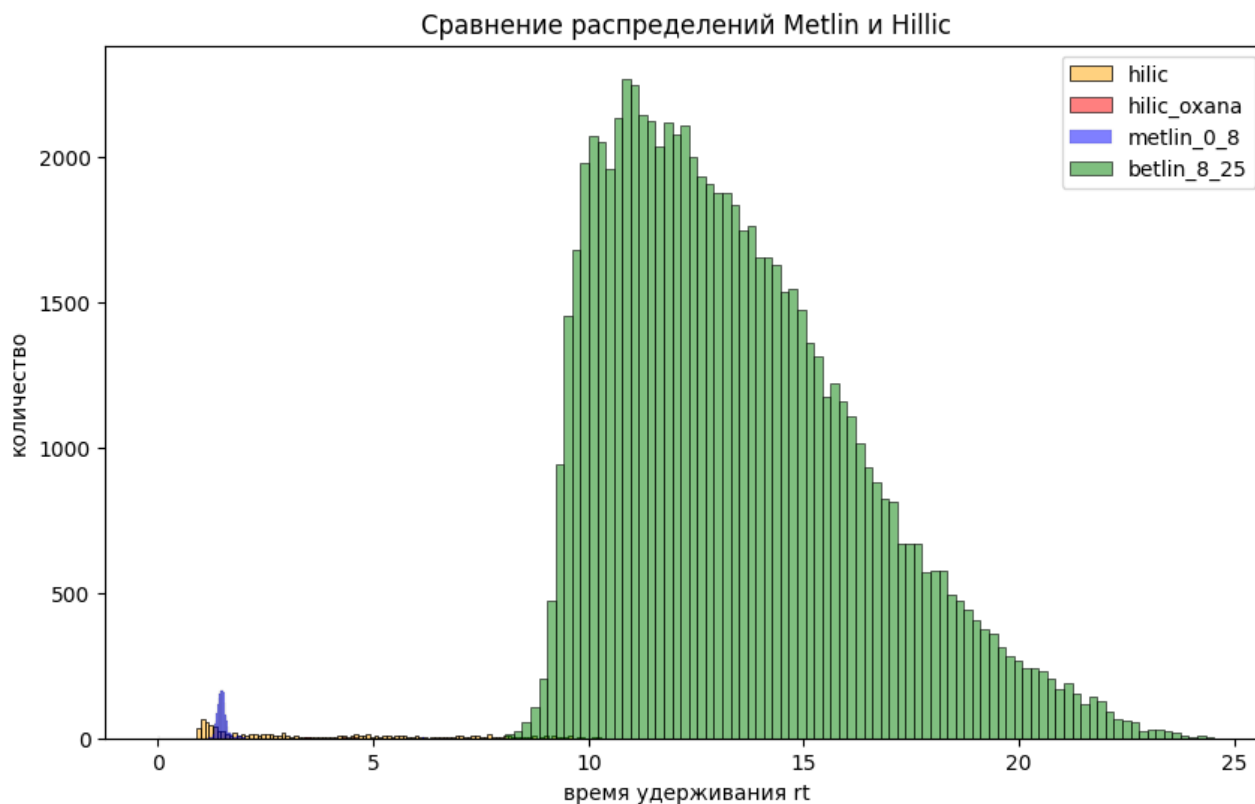


Рис 8. Распределение данных в наборах METLIN, HILIC, hilic\_oxana

Также было принято решение перевести все значения времён удерживания в секунды для увеличения точности.

## Эксперименты с данными

{эксперименты расположены в файле Failed\_experiments\_with\_data.ipynb}

Во время проведения экспериментов по дообучению нейронных сетей, было выявлено, что связка <<обучение на metlin\_8\_25 - дообучение на hilic - тестирование на hilic\_oxana>> показывает точность MAE на 2 меньше, чем <<обучение на metlin\_8\_25 - дообучение на metlin\_0\_8 - тестирование на hilic\_oxana>> (3 и 5 соответственно).

При сравнении наборов данных hilic и metlin были выявлены следующие различия:

- количество уникальных символов отличается
  - METLIN: 36 ['#', '(', ')', '+', '-', '/', '1', '2', '3', '4', '5', '6', '7', '8', '9', '=', '@', 'B', 'C', 'F', 'H', 'I', 'N', 'O', 'P', 'S', '[', '\\', ']', 'c', 'i', 'l', 'n', 'o', 'r', 's']
  - HILIC: 29 ['#', '(', ')', '+', '-', '1', '2', '3', '4', '5', '6', '7', '8', '9', '=', 'B', 'C', 'F', 'H', 'I', 'N', 'O', 'P', 'S', '[', ']', 'i', 'l', 'r']
- количество знаков после запятой в hilic до 6 знаков, в metlin - до 16 знаков
- элементы, состоящие из двух букв, которые встречаются в metlin {'Br', 'Cl', 'Si', 'No', 'Co', 'Cs'}, в hilic {'Br', 'Cl', 'Si'}



Возникла идея проведения следующих экспериментов с данными:

- Эксперимент 1: замена названий элементов, состоящих из 2 букв на символы {'Br': '\*', 'Cl': '&', 'Si': '%', 'Sc': '!', 'Cn': '?'}.
- Эксперимент 2: замена строчных букв с и n на прописные C и N.
- Эксперимент 3: сокращение количества знаков после запятой для значений времени удерживания в обоих наборах данных до 3.
- Эксперимент 4: все описанные преобразования.

**Результаты:** все сделанные операции над данными не принесли заметного повышения точности. Поэтому для повышения точности следует менять параметры и архитектуру искусственной нейронной сети.

## Эксперименты с классическими алгоритмами машинного обучения

{эксперименты расположены в файле [1] Classic mashine learning algorithms.ipynb}

### Основы работы с библиотекой RDKit и концепцией SMILES:

- *описание проведенных исследований*

Для того, чтобы разобраться с главной библиотекой для обработки химических соединений RDKit и научиться работать со SMILES была выбрана статья Retip: retention time prediction for compound annotation in untargeted metabolomics. Paolo Bonini, Tobias Kind, Hiroshi Tsugawa, Dinesh Kumar Barupal, and Oliver Fiehn (Anal. Chem., Just Accepted Manuscript. DOI: 10.1021/acs.analchem.9b05765) и повторены применения алгоритмов XGBoost, MLPRegressor, LightGBM, RandomForestRegressor, BRNN на наборе данных, прикрепленных к статье.

- *полученные результаты*

	HILIC	HILIC Positive	Plasma Positive	All data	Metlin 0-8	Metlin 8-25
<b>XGBoost co StandartScaler</b>	0.716	0.688	0.573	0.731	0.402	0.742
<b>MLP co StandartScaler</b>	13.499	25.965	13.485	9.994	15.954	15.715
<b>Random Forest co StandartScaler</b>	0.687	0.688	0.410	0.421	0.656	0.427
<b>LightGBM co StandartScaler</b>	0.734	0.391	0.659	0.659	0.471	0.736

Как видно из таблицы, алгоритмы XGBoost и RandomForest показали лучшие результаты из всех.

- *Оценка новизны и актуальности полученных результатов*

Результаты получились лучше, чем полученные авторами статьи, но поскольку статья была опубликована 4 года назад, их нельзя назвать актуальными, поскольку за 4 года появилось множество новых архитектур и новых подходов к решению задачи предсказания времени удерживания.

### **Проверка гипотезы о связи между различными предсказанными значениями из наборов данных, полученных в разных лабораториях при разных условиях проведения опытов.**

– *описание проведенных исследований*

Набор данных *hilic\_oxana.csv* был получен опытным путём в лаборатории РАН. По SMILES обученными методами машинного обучения из предыдущего пункта были предсказаны значения времени удерживания.

– *полученные результаты*

Обученные модели машинного обучения из предыдущего пункта не смогли хорошо предсказать полученные опытным путём данные. Не удалось даже понять линейную или нелинейную зависимость между полученными значениями.

На рисунке 9 показаны полученные различными алгоритмами предсказания для каждого из 18 объектов набора данных *hilic\_oxana.csv*.

На рисунке 10а показаны реальные значения для каждого из 18 объектов набора данных *hilic\_oxana.csv* и усреднённые значения, полученные при помощи всех алгоритмов машинного обучения.

На рисунке 10б показана зависимость между усреднёнными значениями, расположенными вдоль горизонтальной оси и реальными значениями, расположенными вдоль вертикальной оси.

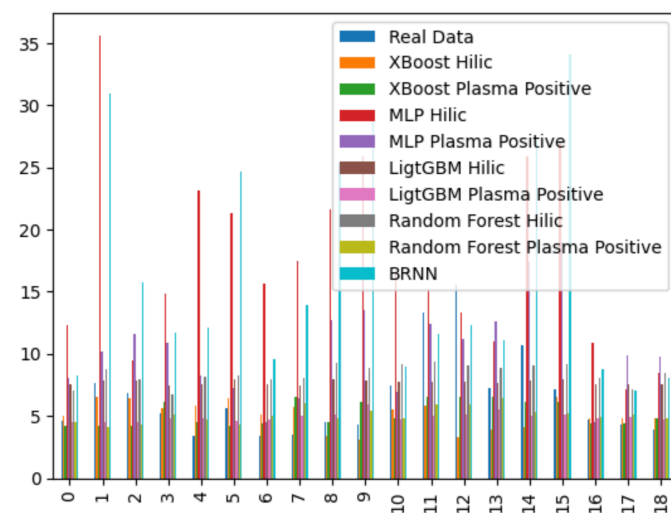


Рис. 9. Сопоставление значений, полученных при помощи различных алгоритмов машинного обучения со значениями, полученными экспериментальным путём

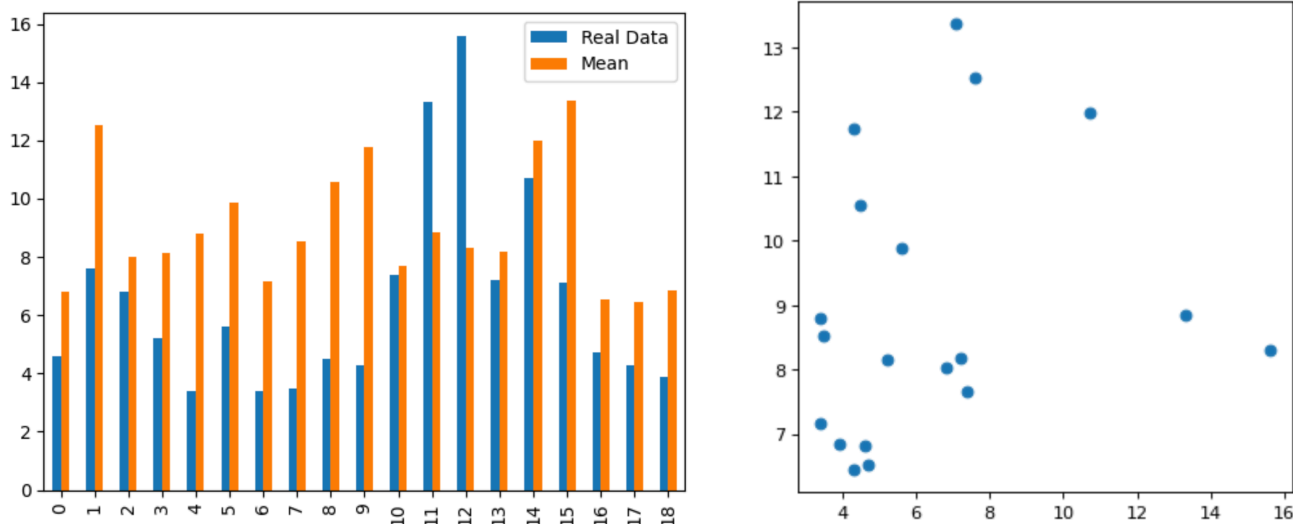


Рис. 10. а) слева - сопоставление усреднённых значений, полученных при помощи различных алгоритмов машинного обучения со значениями, полученными экспериментальным путём. б) справа - вдоль горизонтальной оси расположены усреднённые значения, вдоль вертикальной - реальные значения.

**Результаты:** Найти характер зависимости между различными предсказанными значениями из наборов данных, полученных в разных лабораториях при разных условиях проведения опытов, пока не получилось.

## Эксперименты по предсказанию значений при помощи свёрточной искусственной нейронной сети

{эксперименты расположены в файле [2] CNN\_transfer\_METLIN\_to\_HILIC\_Box\_Cox.ipynb}

Архитектура сети представлена в виде кода ниже:

```
nn.Conv1d(in_channels=37, out_channels=300, kernel_size=5),
nn.ReLU(),
nn.Conv1d(in_channels=300, out_channels=300, kernel_size=5),
nn.ReLU(),
nn.AvgPool1d(kernel_size=177, stride=1),
nn.Flatten(),
nn.Linear(300, 600),
nn.ReLU(),
nn.Linear(600, 1)
```

Использовалось 2 сверточных слоя, с функцией активации ReLU и усреднением полученных значений.

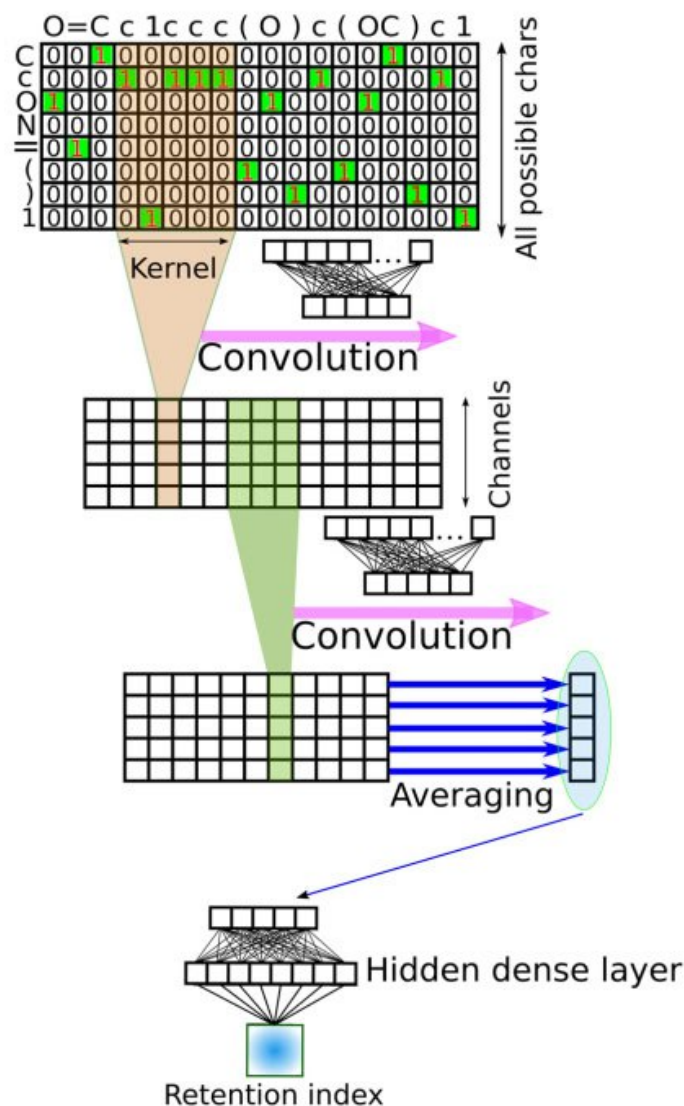


Рис 11. Принципы кодирования данных и усреднения полученных значений

Особенности (рис. 11):

- для правильной работы искусственной нейронной сети все молекулы были дополнены пробелами в конце строки до максимальной длины;
- для подачи данных в нейронную сеть было произведено кодирование one-hot-encoding;
- усреднение производилось вдоль одной полученной строки (её размерность после предыдущих преобразований составила 177 символов).

Функция потерь, а также основная метрика - MAE.

Обучение нейронной сети производилось на наборе данных metlin\_8\_25, была достигнута точность 0.8 по метрике MAE (рис. 12).

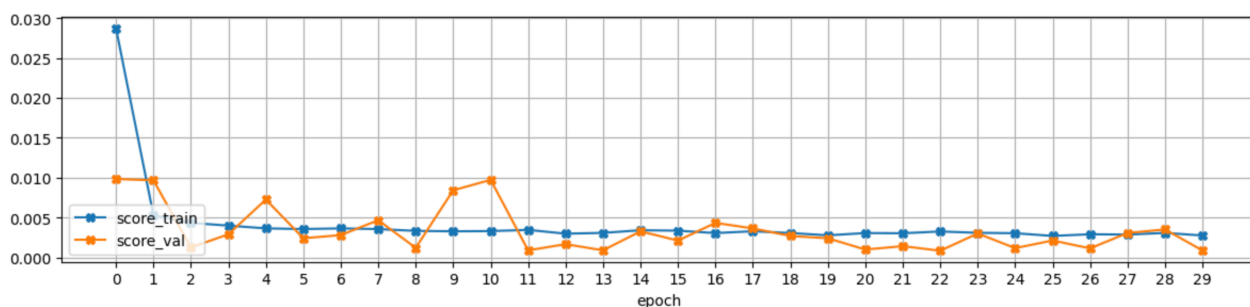


Рис 12. Значение точности MAE в зависимости от номера эпохи

#### Эксперимент 1

Дообучение на metlin\_0\_8 с последующим дообучением на hilic и тестированием на hilic\_oxana без замораживания весов и без добавления новых слоёв.

#### Эксперимент 2

Дообучение на metlin\_0\_8, потом hilic, тестирование на hilic\_oxana с изменением последнего слоя.

#### Эксперимент 3

Дообучение на hilic, потом на metlin\_0\_8, тестирование на hilic\_oxana с изменением последнего слоя

#### Эксперимент 4

Дообучение на metlin\_0\_8, потом на hilic, тестирование на hilic\_oxana с добавлением нового последнего слоя

#### Эксперимент 5

Дообучение на hilic, потом на metlin\_0\_8, тестирование на hilic\_oxana с добавлением нового последнего слоя

**Результаты и выводы: самый успешный пайплайн состоит из следующих шагов:**

- обучение нейронной сети на наборе данных metlin\_8\_25
- дообучение на metlin\_0\_8 (без замораживания весов и без добавления новых слоёв)
- тестированием на hilic\_oxana

Полученная последовательность действий приводит к наименьшему MAE на наборе hilic\_oxana - 3.8 (эксперимент 1) против 4.1-5.2 для других архитектур.

Но, поскольку, мы нашли одно пересечение в наборе данных hilic и hilic\_oxana, где время удерживания отличается существенно (что как раз связано с условиями проведения экспериментов, которые свои в каждой лаборатории): C1=CN=CN1 - 4.5255 (hilic) и 7.4 (hilic\_oxana), этот результат можно считать хорошим.

# Итоговая архитектура и самый успешный эксперимент

{эксперименты расположены в файле [3] CNN\_XGB\_RF\_METLIN\_HILIC\_SECOND.ipynb}

1) Для первой ступени модели была выбрана описанная выше архитектура свёрточной нейронной сети, но было добавлено преобразование Бокса-Кокса для усреднения значений предсказываемого значения времени удерживания.

Проводилось обучение на наборе данных metlin\_8\_25 с последующим дообучением на наборе данных metlin\_0\_8.

Получены предсказания на наборе данных hilic, которые были добавлены отдельным столбцом данному набору и переданы как обучающие на следующую ступень.

2) Второй ступенью был выбран ансамбль алгоритмов классического машинного обучения XGBoost и RandomForest, поскольку они показали лучшие метрики из всех алгоритмов машинного обучения. Также была проведена стандартизация данных при помощи StandartScaler.

Итоговое значение по метрике Mean absolute error после всех вышеперечисленных действий удалось снизить с 258.843 до 144.414.

Сравнение с результатами, полученными в статьях из списка литературы при решении схожих задач в газовой хроматографии:

Модель/статья	MAE	Размер тестового набора данных	MAE/размер набора данных
GNN-RT [5]	87	484 (PredRet)	0.17
Трёх-ступенчатая модель [7]	24.3	151 (DB624)	0.16
Наша модель CNN	147	18 (hilic_oxana)	8.2

Поскольку задача и набор данных hilic\_oxana по сути уникальный, нет возможности явно сравнить эффективность построенной нами модели. Но существует готовая библиотека DeepChem для предсказания данных в области химии.

Была выбрана готовая модель MultitaskRegressor из библиотеки DeepChem MultitaskRegressor из библиотеки DeepChem, обучена на наборе данных metlin\_8\_25 и наборе metlin\_0\_8 (с использованием преобразования Бокса-Кокса и без него) и получены предсказания для набора данных hilic\_oxana. Лучшие результаты представлены в таблице.

{эксперименты расположены в файле [4] MultitaskRegressor\_DeepChem.ipynb}

Сравнение с результатами, полученными при предсказании готовой моделью

	smiles	real rt	predicted rt our model	predicted rt model from DeepChem
0	C1=NC=NN1	276.0	361.90580489	556.6185
1	C1(=NC(=NN1)N)N	456.0	350.51210793	687.9944
2	C1=NN=CN1N	408.0	404.89112242	615.11884
3	CC1=NC=NN1	312.0	300.81737119	556.1645
4	C1=NNN=C1	204.0	314.93627073	556.6185
5	C1=NNC(=N1)N	336.0	361.90580489	687.9944
6	C1=CNN=C1	204.0	350.51210793	615.11884
7	CC1=CC=NN1	210.0	404.89112242	556.1645
8	CN1C=CC(=N1)N	270.0	300.81737119	556.6185
9	CCC1=CC(=NN1)N	258.0	314.93627073	687.9944
10	C1=CN=CN1	444.0	361.90580489	615.11884
11	CC1=NC=CN1	798.0	350.51210793	556.1645
12	CC1=NC=CN1C	936.0	404.89112242	556.6185
13	CCN1C=CN=C1	432.0	300.81737119	687.9944
14	CC1=NC=CN1CCCN	642.0	314.93627073	615.11884
15	CC1=NC(=NC(=N1)N)N	426.0	361.90580489	556.1645
16	C1=CC=NN=C1	282.0	350.51210793	556.6185
17	C1=CN=CN=C1	258.0	404.89112242	687.9944
18	C1=CN=CC=N1	234.0	300.81737119	615.11884

По приведённым значениям можно сделать вывод, что наша архитектура показала заметно лучшие результаты, чем готовая модель.

## Перспективы для дальнейшей работы

- 1) Можно заменить построенную в данной работе модель на уже обученную из библиотеки DeepChem по примеру репозитория <https://github.com/ncfrey/litmatter/blob/main/prototyping.ipynb> {эксперименты в файле [ \_notworked]\_LitDeepChem.ipynb}
- 2) Получить предсказания от готовой большой языковой модели ChemGPT из библиотеки HuggingFace по примеру репозитория <https://github.com/kjappelbaum/gptchem/blob/main/README.md> {эксперименты в файле [ \_not\_worked]\_GPTChem\_hilic\_oxana.ipynb}

## Заключение

Перспективы для успешного решения задачи по переносу данных обучения есть. Удалось применить полученные знания и добиться некоторых улучшений в решении поставленной задачи, также возникло понимание по продолжению дальнейшей работы.

## Описание приложенных файлов

- [0] Data\_preparation\_METLIN\_and\_HILIC.ipynb - загрузка и предобработка данных
- [1] Classic mashine learning algorithms.ipynb - эксперименты с алгоритмами классического машинного обучения
- [2] CNN\_transfer\_METLIN\_to\_HILIC\_Box\_Cox.ipynb - эксперименты со свёрточной нейронной сетью
- [3] CNN\_XGB\_RF\_METLIN\_HILIC\_SECOND.ipynb - основной файл с двух-ступенчатой моделью
- [4] MultitaskRegressor\_DeepChem.ipynb - эксперименты с готовой архитектурой из библиотеки DeepChem
- [\_notworked]\_LitDeepChem.ipynb - эсперименты с моделью LitDeepChem
- [\_not\_worked]\_GPTChem\_hilic\_oxana.ipynb - эксперименты с языковой моделью
- Failed\_experiments\_with\_data.ipynb - неудачные эксперименты с данными

## Список литературы

- [1] Generalized Calibration Across Liquid Chromatography Setups for Generic Prediction of Small-Molecule Retention Times. Robbin Bouwmeester, Lennart Martens, and Sven Degroeve Analytical Chemistry 2020 92 (9), 6571-6578, <https://pubs.acs.org/doi/abs/10.1021/acs.analchem.0c00233>
- [2] Elizaveta S. Fedorova, Dmitriy D. Matyushin, Ivan V. Plyushchenko, Andrey N. Stavrianidi, Aleksey K. Buryak. Deep learning for retention time prediction in reversed-phase liquid chromatography, Journal of Chromatography A, Volume 1664, 2022, 462792, ISSN 0021-9673, <https://www.sciencedirect.com/science/article/abs/pii/S0021967321009146>
- [3] Prediction of Liquid Chromatographic Retention Time with Graph Neural Networks to Assist in Small Molecule Identification. Qiong Yang, Hongchao Ji, Hongmei Lu, and Zhimin Zhang. Analytical Chemistry 2021 93 (4), 2200-2206, <https://pubs.acs.org/doi/abs/10.1021/acs.analchem.0c04071>
- [4] Qiyue Kang, Pengfei Fang, Shuai Zhang, Huachuan Qiu, Zhenzhong Lan. Deep graph convolutional network for small-molecule retention time prediction. Journal of Chromatography A, Volume 1711, 2023, 464439, ISSN 0021-9673, <https://www.sciencedirect.com/science/article/abs/pii/S0021967323006647>, <https://github.com/michaelwitting/RepoRT>
- [5] Retention Time Prediction through Learning from a Small Training Data Set with a Pretrained Graph Neural Network. Youngchun Kwon, Hyukju Kwon, Jongmin Han, Myeonginn Kang, Ji-Yeong Kim, Dongyeeb Shin, Youn-Suk Choi, and Seokho Kang. Analytical Chemistry 2023 95 (47), 17273-17283, <https://pubs.acs.org/doi/abs/10.1021/acs.analchem.3c03177>
- [6] Jun Xue, Bingyi Wang, Hongchao Ji, WeiHua Li, RT-Transformer: retention time prediction for metabolite annotation to assist in metabolite identification, Bioinformatics, Volume 40, Issue 3, March 2024, btae084, <https://academic.oup.com/bioinformatics/article/40/3/btae084/7613958>
- [7] Matyushin, D.D.; Sholokhova, A.Y.; Buryak, A.K. Deep Learning Based Prediction of Gas Chromatographic Retention Indices for a Wide Variety of Polar and Mid-Polar Liquid Stationary Phases. Int. J. Mol. Sci. 2021, 22, 9194, <https://www.mdpi.com/1422-0067/22/17/9194>