**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

###### Центр непрерывного образования

###### Факультета компьютерных наук

**ИТОГОВЫЙ ПРОЕКТ**

Решение задачи регрессии для многокомпонентных временных рядов для детектирование режимов работы насосного оборудования.

|  |  |
| --- | --- |
|  | Выполнил:  Сетдеков Кирилл Раильевич |
|  |  |
|  | Руководитель:  Абдуракипов Сергей Сергеевич |

Москва 2021

Оглавление

[Введение 2](#__RefHeading___Toc3502_3228052382)

[Обзор литературы 2](#__RefHeading___Toc3504_3228052382)

[Методы 6](#__RefHeading___Toc3516_3228052382)

[Эксперименты 7](#__RefHeading___Toc3518_3228052382)

[Описание данных 7](#__RefHeading___Toc724_1315981873)

[Заключение 8](#__RefHeading___Toc3520_3228052382)

[Приложения 9](#__RefHeading___Toc3522_3228052382)

[Приложение 1. Гиперпараметры для моделей 9](#__RefHeading___Toc726_1315981873)

[Список литературы 9](#__RefHeading___Toc3524_3228052382)

# Введение

Введение — нужно объяснить тему работы “человеческим языком”, поговорить об актуальности темы. Здесь же даётся формальная постановка задачи.

# Обзор литературы

нужно показать, что вы изучили наработки по вашей теме, обрисовать текущее состояние области

На основе сравнения методов по MAE на наборе данных ETTh2 (720), можно выделить 4 продвинутых подходов к прогнозированию временных рядом. Перечисляю их в порядке снижения качества прогнозирования:

| Ранг | Модель | MAE | MSE | Статья | Год |
| --- | --- | --- | --- | --- | --- |
| 1 | [**SCINet** (Одномерная)](https://paperswithcode.com/paper/time-series-is-a-special-sequence-forecasting) | 0.399 | 0.249 | [Time Series is a Special Sequence: Forecasting with Sample Convolution and Interaction](https://paperswithcode.com/paper/time-series-is-a-special-sequence-forecasting) | 2021 |
| 2 | [**QuerySelector**](https://paperswithcode.com/paper/long-term-series-forecasting-with-query) | 0.4130 | 0.2585 | [Long-term series forecasting with Query Selector - efficient model of sparse attention](https://paperswithcode.com/paper/long-term-series-forecasting-with-query) | 2021 |
| 3 | [**Informer**](https://paperswithcode.com/paper/informer-beyond-efficient-transformer-for) | 0.431 | 0.277 | [Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting](https://paperswithcode.com/paper/informer-beyond-efficient-transformer-for) | 2020 |
| 4 | [**Transformer**](https://paperswithcode.com/paper/long-term-series-forecasting-with-query) | 0.4340 | 0.2853 | [Long-term series forecasting with Query Selector - efficient model of sparse attention](https://paperswithcode.com/paper/long-term-series-forecasting-with-query) | 2021 |
| 5 | [**SCINet** (Многомерная)](https://paperswithcode.com/paper/time-series-is-a-special-sequence-forecasting) | 0.761 | 1.074 | [Time Series is a Special Sequence: Forecasting with Sample Convolution and Interaction](https://paperswithcode.com/paper/time-series-is-a-special-sequence-forecasting) | 2021 |

1 и 5 место, отличаются тем, используется ли многомерная реализация или одномерная реализация модели. Подробнее остановимся на 4 подходах и трех статьях, где они описываются:

1. Time Series is a Special Sequence: Forecasting with Sample Convolution and Interaction[[1]](#footnote-2) (использование свёрточных нейронных сетей и сравнение их с трансформерами) Данная статья проводит дальнейшее развитие актуальных архитектур для временных рядом (рекуррентные нейронные сети, модель транформеров, сверточные нейронные сети для временных рядов). Основная идея — модифицированный подход сверточной нейронной сети, где не только расширяется зона внимания от слоя к слою, но и проводится последовательность сжатие-свертка-смешивание. Это позволяет одновременно смотреть на параметры временного ряда разного масштаба, что помогает лучше извлекать признаки из временного ряда и приводит к росту предсказательной способности в модели SCINet, предложенной авторами, в сравнении с прошлыми архитектурами.   
   На данный момент один из передовых по точности алгоритм, имеющий как многомерный так и одномерный вариант.  
   Есть официальный репозиторий[[2]](#footnote-3) с кодом и реализацией на PyTorch.
2. Long-term series forecasting with Query Selector -- efficient model of sparse attention[[3]](#footnote-4) (Использование трансформеров и query selector модели)  
   В этой статье предлагается развития идеи трансформеров за счет построения детерминистического алгоритма построения разреженной матрицы внимания.  
   Есть реализация кода от авторов модели на PyTorch[[4]](#footnote-5).
3. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting[[5]](#footnote-6) (использование информеров)  
   Данная статья разбирает проблемы прогнозирования длительных временных рядов (LSTF) и отмечает проблемы использования архитектуры трансформеров для этих целей — сложность по времени и большие требования по памяти и ограничения, присущие всем архитектурам энкодер-декодер. В качестве альтернативы, авторы предлагают более экономичную архитектуру модели для прогнозирования длительных временных рядов «Информеры», которые обладают тремя основными характеристиками 1) механизм внимания, который имеет сложность по памяти и времени; 2) механизм внимания, который выделяет ключевые признаки, одновременно с делением пополам размера входов на каскадных слоях, что позволяет эффективно работать с чрезвычайно длинными временными рядами; 3) декодер генеративного типа, который строит прогноз за один шаг, а не итеративно, чем значительно ускоряет построение прогнозов.  
   В данной статье проводится сравнения с другими методами и показано превосходство предложенного подхода в разрезе прогнозирования длительных временных рядов.  
   Есть реализация кода от авторов модели на PyTorch[[6]](#footnote-7).

Эти статьи про методы, которые не входят в state-of-the-art, но я считаю кажется, их можно упомянуть:

1. Probabilistic Forecasting with Temporal Convolutional Neural Network[[7]](#footnote-8)  
   В статье описывается подход к созданию прогнозов для мультивариативных временных рядом на основе свёрточных нейронных сетей с использованием обучения признакам.  
   Есть официальная реализация в коде от авторов работы на keras[[8]](#footnote-9), что важнее — реализован внутри пакета Darts – современный и быстроразвивающийся пакет для работы с временными рядами[[9]](#footnote-10)
2. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting[[10]](#footnote-11)   
   Основной фокус на одномерных временных рядах — минус при анализе мультивариантных временных рядов. Предложен и реализован подход на основе глубокой нейронной сети со связями вперед и назад между слоями и набором множества полносвязных слоев внутри архитектуры сети.  
   Есть реализация от компании, связанной с автором статьи[[11]](#footnote-12)   
   Также реализован внутри пакета Darts
3. Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting.[[12]](#footnote-13)  
   На этой работе в первую очередь базируется работа Liu 2021 года (модель SCINet)  
   Основная идея — использование трансформеров и обучение поведению временного ряда на разных масштабах плюс выделение отдельной части модели под интерпретируемое представление внимания обученной сети, что позволяет и достигнуть лучших результатов в нескольких бенчмарках и отойти от реализации — черный ящик.   
   Есть реализация на tensorflow[[13]](#footnote-14), статья от исследователей в google research.

# Методы

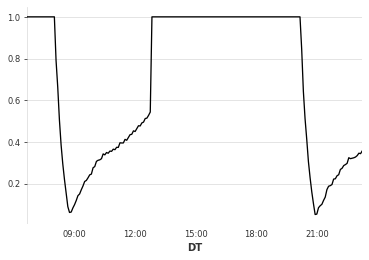
Методы — описание того, что вы предлагает сделать. Например, если вы предлагаете свой подход к генерации текстов с помощью нейросетей, то здесь описывается архитектура сети и то, как её нужно обучать.

# Эксперименты

## Описание данных

Для этой работы использованы данные, являющиеся сокращенным набором во времени и числу переменных набором данных, которые компания «Сургетнефтегаз» предоставляла в рамках открытого соревнования data science[[14]](#footnote-15). В работе использован набор данных по эксплуатации погружного оборудования фонда нефтяных скважин (17 скважин). Характеристики данных: 150 тысяч записей за два календарных месяца (июнь — июль 2019 года), выравненные по временной оси данные телеметрии с частотой дескретизации 5 минут. Для построения моделей использовали 15 независимых переменных, большая часть которых — телеметрия работы насосов (нагрузка двигателя, коэффициент мощности, давление в коллекторе узла учета, наработка насоса, расход жидкости и газа и т.д.), а также идентификатор номера насоса и колонка времени изменений. Пример динамики целевой переменной, «средняя скорость изменения давления на приеме насоса в ЧАС, МПа/час.» показан на рисунке ниже.

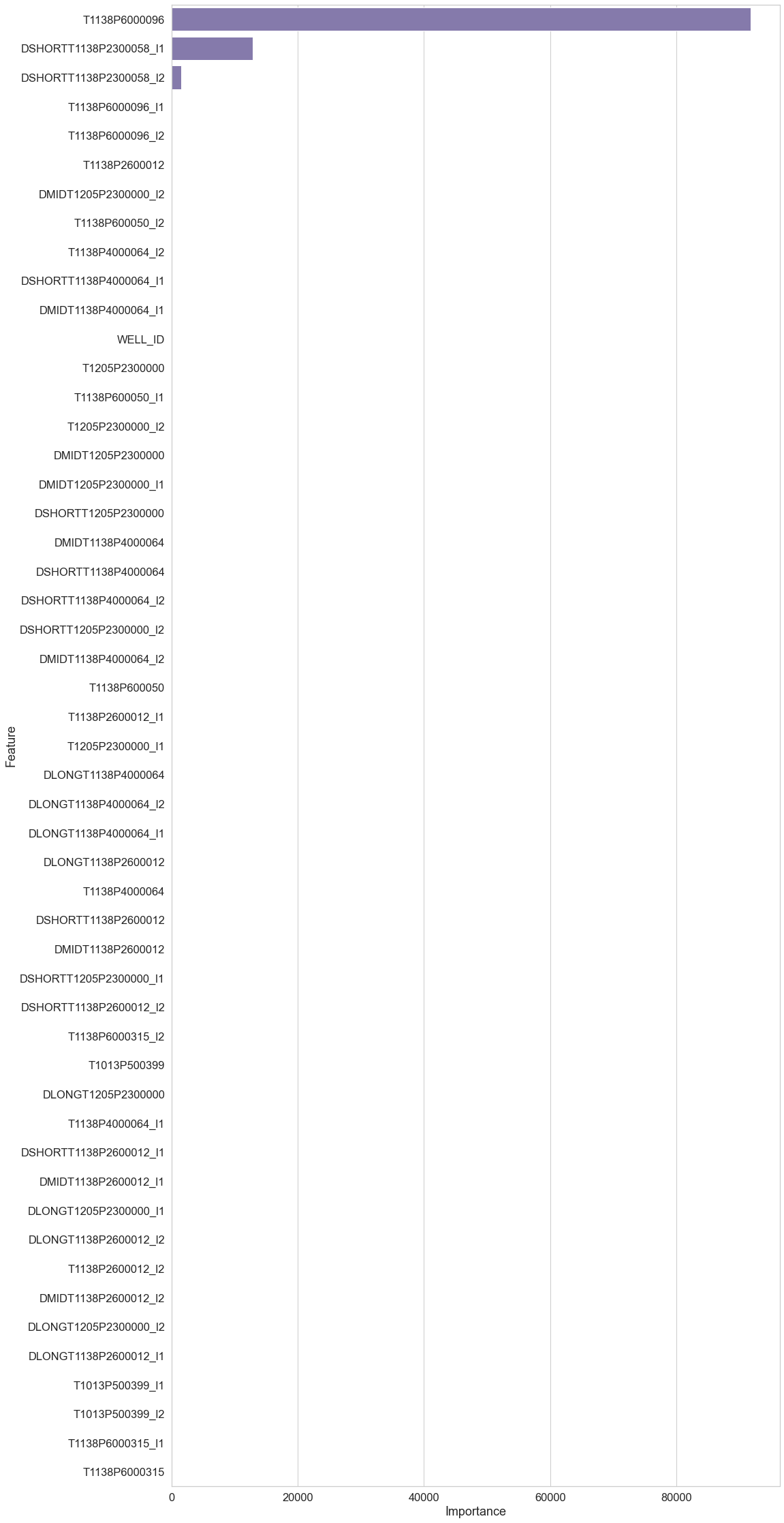
Динамика признаков визуально выглядела хаотичной и не связанной с целевой переменной, за исключением признаков «T1138P6000096 Наработка двигателя с момента последнего включения, сек» и «T1138P4000064': "Загрузка двигателя, %», для остальных переменных направление влияние и степень их взаимосвязей первого порядка была заранее не очевидна. В качестве дополнительного анализа я использовал вывод feature importance для auto ml модели.

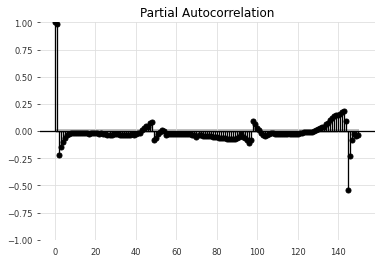
Рисунок 1: Нормированная к интервалу [0;1] динамика целевой переменной

Ниже пример значений переменных:

|  |  |
| --- | --- |
| *Рисунок 2: Время с момента включения насоса* | *Рисунок 3: Нагрузка насоса* |
| *Рисунок 4: Средняя скорость изменения давления в коллекторе ИУ* | *Рисунок 5: Целевая переменная* |

На рисунке выше мы видим, что только 1 переменная (время включения насоса) и значение целевой переменной с задержкой 1 и 2 шага оказались значимые в AutoML модели. Можно предположить, что взаимосвязи между переменными нелинейные и отложенные во времени, что и требует применения современных методов машинного обучения, заточенных под многомерные временные ряды.

Рисунок 6: Нормированная значимость признаков для модели LAMA

Рисунок 7: Частные автокрреляции для целевой переменной

Дополнительный анализ в части целевой переменной — анализ частных автокорреляций. Приведу правее график PACF для целевой переменной для случайного насоса. По нему видно, что до 144 лага значения являются значимыми. Уровень 144 временных интервала я буду использовать для моделей как базовую настройку размера входящего окна. Тесты показали, что более короткие значения приводят к снижению качества модели, а более высоки — увеличивают сложность моделей и увеличивают шанс переобучения.

## **Предобработка данных**

Для AutoML модели из пакета LAMA значения целевой переменной и признаков не нормализовывались, и идентификаторы насосов не кодировались отдельно — модель умеет работать с категориальными переменными. После того как модель показала плохой результат, я добавил значение целевой переменной и всех признаков с лагом 1 и 2.

Для всех остальных моделей использовался следующий поток обработки данных:

1. Идентификаторы насосов кодировались с помощью LabelBinarizer в отдельные столбцы с 1 и 0 значениями;
2. Датафрейм из предыдущего шага был преобразован в список из датафреймов, где для каждого были наблюдения только по 1 насосу (это связано с ограничением пакета darts);
3. Для модели Time fusion transformer дополнительно:
   1. Создавалась колонка time\_idx – индексы по времени, идущие от 0 с шагом 1 для каждого момента времени, который есть в обучающей выборке;
   2. Не проводилось разделение одного датафрейма на список из датафреймов, так как пакет pytorch forecasting умеет работать с множественными наблюдениями на 1 момент времени;
   3. Был подготовлен dataset и dataloaded в формате pytorch.

## Построение моделей

На основе обучающих выборок, для целевой переменной «Средняя скорость изменения давления на приеме насоса в ЧАС, МПа/час» я построил 4 модели, описанных выше, которые я отбирал на основе баланса новизны и простоты применения:

* TCN;
* N-Beats;
* Transformer;
* Time Fusion Transformer.

Как базовую линию я использовал модель LAMA, которая является AutoML моделью, поддерживает автоматический подбор гиперпараметров и обучает ансамбль из бустингов в рамках фиксированного времени на обучение.

Общий подход к построению и тестированию моделей:

* В моделях использовались все независимые переменные;
* Прогноз строился на 1 шаг вперед;
* Для сравнения модели использовались 4 метрики:
  + SMAPE (основная метрика для ранжирования моделей) — симметричная абсолютная ошибка в процентах. Выбрана основной на основе недостатка MAPE – при равенстве истинного значения 0, MAPE выдает неопределенные значения, так как происходит деление на 0. Специфика данных — в целевой переменной пристуствуют длительные периоды нулевых значений.
  + MAPE – средняя абсолютная ошибка в процентах;
  + MAE – средняя абсолютная ошибка;
  + R2 – коэффициент детерминации.
* Гиперпараметры были установлены значениями по умолчанию, или подобраны вручную для максимизации SMAPE на валидационных выборках;
* Обучение модели шло на 70% первых наблюдений из обучающей выборки, на 30% подбирались гиперпараметры, потом модель обучалась на всей обучающей выборке;
* Валидация и финальные метрики для результов модели проводились на двух отложенных выборках для валидации «valid1.csv» и «valid2.csv». Прогноз моделей на двух выборках и целевая переменная конкатенировались по двум валидацоинным выборкам и на этих данных рассчитаны метрики.

## по добавлению учета номера насоса

\* tcn показал улучшение метрик smape на валидации 0.05238308359709533 -> 0.0444367612098486

\* transformers показали небольшое метрик smape на валидации 0.07035859564973221-> 0.07002090974577367

\* nbeats ухудшение smape на валидации 0.13431373200428287 -> 0.17577817752305092

# Заключение

# Приложения

## Приложение 1. Гиперпараметры для моделей

### LAMA

Финальная модель обучалась методом TabularUtilizedAutoML, который подбирает гиперпараметры и строит ансамбли из моделей отталкиваясь от ограничения по времени. Я использовал:

* timeout (ограничение по времени) – 1000 секунд
* cpu\_limit (число ядер) – 10
* cv (кросс-валидация) — 5

### TCN

Финальная модель TCNModel из пакета Darts была обучена со следующими параметрами:

* n\_epochs (число эпох) — 100;
* input\_chunk\_length (длина входящего окна) — 144;
* output\_chunk\_length (длина прогноза) — 1;
* dropout (вероятность dropout на сверточных слоях) — 0.01;
* dilation\_base (основание для степенной функции, которая отвечает за расширение рецептивного поля на каждом слое) 2;
* weight\_norm (использовать нормализацию весов) True;
* kernel\_size (размер ядра сверки) – 5;
* num\_filters (число каналов на каждом слое сверки) — 3.

### N-Beats

Финальная модель NBEATSModel из пакета Darts была обучена со следующими параметрами:

* n\_epochs (число эпох) — 100;
* input\_chunk\_length (длина входящего окна) — 144;
* output\_chunk\_length (длина прогноза) — 1;
* generic\_architecture=True,
* num\_stacks (число слоев трендов и сезонностей) — 10,
* num\_blocks (число блоков в одном слое) — 1,
* num\_layers (число полносвязных слоев в каждом блоке каждого стэка)— 4,
* layer\_widths (число нейронов в полносвязных слоях) — 512,
* nr\_epochs\_val\_period (сколько эпох ждать до начала теста) —1,
* batch\_size (размер батча) —800.

### Transformer

Финальная модель TransformerModel из пакета Darts была обучена со следующими параметрами:

* n\_epochs (число эпох) — 100;
* input\_chunk\_length (длина входящего окна) — 144;
* output\_chunk\_length (длина прогноза) — 1;
* dropout (вероятность dropout на сверточных слоях) — 0.1;
* batch\_size (размер батча) — 32;
* d\_model (ожидаемое число фичей) — 32;
* nhead (число голов на входе механизма внимания) — 16;
* num\_encoder\_layers (число слоев энкодера) — 3;
* num\_decoder\_layers (число слоев декодера) —3;
* dim\_feedforward — 512;
* функция активации — "relu";
* остальные параметры — стандартные по-умолчанию

Time Fusion Transformer

Финальная модель TemporalFusionTransformer из пакета pytorch\_forecasting была обучена со следующими параметрами:

* learning\_rate (коэффициент скорости обучения) — 0,1789, подобран на обучающей выборке;
* размер скрытого слоя — 32;
* dropout (вероятность dropout на сверточных слоях) — 0.1;
* остальные параметры — стандартные по-умолчанию.

# Список литературы

1. Time Series is a Special Sequence: Forecasting with Sample Convolution and Interaction Minhao Liu, Ailing Zeng, Zhijian Xu, Qiuxia Lai, Qiang Xu https://arxiv.org/abs/2106.09305 [↑](#footnote-ref-2)
2. https://github.com/cure-lab/SCINet [↑](#footnote-ref-3)
3. Long-term series forecasting with Query Selector -- efficient model of sparse attention Jacek Klimek, Jakub Klimek, Witold Kraskiewicz, Mateusz Topolewski https://arxiv.org/abs/2107.08687v2 [↑](#footnote-ref-4)
4. https://github.com/moraieu/query-selector [↑](#footnote-ref-5)
5. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, Wancai Zhang https://arxiv.org/abs/2012.07436v3 [↑](#footnote-ref-6)
6. <https://github.com/zhouhaoyi/Informer2020> — самый популярный по активности редакторов и наблюдателей репозиторий из этого обзора [↑](#footnote-ref-7)
7. Probabilistic Forecasting with Temporal Convolutional Neural Network Yitian Chen, Yanfei Kang, Yixiong Chen, Zizhuo Wang https://arxiv.org/abs/1906.04397v3 [↑](#footnote-ref-8)
8. https://github.com/ashishpatel26/tcn-keras-Examples [↑](#footnote-ref-9)
9. https://github.com/unit8co/darts [↑](#footnote-ref-10)
10. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, Yoshua Bengio [↑](#footnote-ref-11)
11. https://github.com/ElementAI/N-BEATS [↑](#footnote-ref-12)
12. Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting Bryan Lim, Sercan O. Arik, Nicolas Loeff, Tomas Pfister https://arxiv.org/abs/1912.09363v3 [↑](#footnote-ref-13)
13. https://github.com/google-research/google-research/tree/master/tft [↑](#footnote-ref-14)
14. On increasing the efficiency of electric submersible pumps by using big data processing technologies, S Abdurakipov [↑](#footnote-ref-15)