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

ФАКУЛЬТЕТ «Робототехника и комплексная автоматизация»

КАФЕДРА «Системы автоматизированного проектирования (РК-6)»

РАСЧЁТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

по дисциплине «Модели и методы анализа проектных решений»

на тему

«Прогностическая способность глобальных нейросетевых моделей в случае нестационарных временных рядов»

Студент РК6-76Б (группа) _____ (подпись, дата) Новокшанов Е.А. (ФИО)

Руководитель КП _____ (подпись, дата) Соколов А.П. (ФИО)

Консультант _____ (подпись, дата) Першин А.Ю. (ФИО)

Москва, 2023

РЕФЕРАТ

курсовой проект: 23 с., 10 рис., 1 табл., 12 источн.

.

Работа посвящена проблеме прогнозирования нестационарных временных рядов нейросетевыми моделями и решению этой проблемы с помощью реализации трансформера и анализа его эффективности. В пояснительной записке приведено теоретическое описание рассмотренных методов, а также их практическое применение для решения выбранной проблемы и анализ результатов.

Тип работы: курсовой проект.

Тема работы: *«Прогностическая способность глобальных нейросетевых моделей в случае нестационарных временных рядов»*.

Объект исследования: простые трансформеры и их применения в задачах прогнозирования нестационарных временных рядов.

Основная задача, на решение которой направлена работа: исследование современных методов прогнозирования нестационарных временных рядов и проведение анализа их эффективности.

Цели работы: поиск источников литературы, обзор, определение объекта исследования и перспектив развития, реализация модели DLinear.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В современном мире объемы данных, генерируемых компьютерными системами, растут с каждым днем. Одной из задач, которые возникают перед исследователями и аналитиками данных, является прогнозирование значений временных рядов. Временные ряды широко применяются в финансовой аналитике, климатологии, экономике и других областях.

Однако прогнозирование временных рядов является нетривиальной задачей, особенно в случае нестационарных временных рядов. Нестационарность означает, что свойства ряда, такие как среднее значение и дисперсия, меняются со временем. Использование традиционных статистических методов, таких как ARIMA, может быть неэффективным в таких случаях.

В последние годы нейросетевые модели показали свою эффективность в прогнозировании временных рядов. Основное преимущество нейросетевых моделей состоит в их способности автоматически извлекать сложные зависимости в данных. Глобальные нейросетевые модели, такие как рекуррентные нейронные сети (RNN) и сверточные нейронные сети (CNN), позволяют учитывать долгосрочные зависимости в данных.

Однако последние исследования[1] говорят о том, что классические модели ML(machine learning) и DL(deep learning), а так же их ансамбли, во многих случаях справляются с задачей прогнозирования нестационарных временных рядов хуже статистических моделей. Для предсказания значений временного ряда было предложено [2] использовать трансформеры, которые раньше использовались только в задачах NLP(Natural Language Processing).

На основе обзора статей [3], [4], [1], [5], [2], [6], [7], [8], [9] и [10], объектом исследования были выбраны простейшие однослойные трансформеры и их применения в задачах прогнозирования нестационарных временных рядов.

Выводы

В данном обзоре литературы были рассмотрены различные модели трансформеров и других моделей ML и DL, а также их эффективность в контексте задач предсказания временных рядов. Был проведен анализ работы модели DLinear и метода скользящего окна. Также был проведен обзор литературы, рассматривающей нестационарные временные ряды и их свойства.

Был проведен анализ различных глобальных нейросетевых моделей, таких как Recurrent Neural Network(RNN) [4], Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU) и другие. Было показано, что эти модели успешно применяются для анализа стационарных временных рядов, но не всегда справляются с нестационарными данными.

Тем не менее, вопрос об отставании глобальных нейросетевых моделей от статистических методов требует более детального рассмотрения. В ряде случаев, статистические методы могут проявлять себя эффективно. Однако, именно адаптация и обучение на больших объемах данных при использовании нейросетей могут привести к лучшей способности моделей обрабатывать сложные, нелинейные и нестационарные зависимости.

Также было выявлено, что даже простые модели трансформеров могут хорошо себя показывать при прогнозировании нестационарных временных рядов.

Поэтому применение трансформеров в случае нестационарных временных рядов остается открытой проблемой. Дальнейшие исследования должны быть направлены на разработку более эффективных методов анализа и прогнозирования нестационарных данных.

1 Постановка задачи

1.1 Концептуальная постановка задачи

Целью данной работы является разработка модели трансформера DLinear и анализ результатов прогнозирования нестационарных временных рядов. Реализация обучения модели с помощью скользящего окна. После реализации данной модели следует провести тестирование на различных наборах данных, с разными параметрами модели, и проанализировать метрики. На вход модель получает матрицу данных размером $X \in \mathbb{R}^{L \times C}$, где L - размер окна скользящего среднего и C - количество признаков. На выходе мы получаем предсказание временного на T шагов, что представляет из себя матрицу $\hat{X} \in \mathbb{R}^{T \times C}$.

Таким образом, будут решены следующие задачи:

- разработка модели DLinear;
- реализация обучения с помощью механизма rolling window;
- разработанная и обученная модель будет протестирована на ETT датасете;
- результаты будут проанализированы на основе метрик MAPE, MAE и MSE и сделать выводы.

2 Теоретическая часть

2.1 Метрики

Для сравнения точности предсказания временного ряда разными моделями используются различные метрики:

- MAPE;
- MAE;
- MSE.

По данным метрикам проводился анализ эффективности в статье[1].

Средняя абсолютная процентная ошибка (MAPE) рассчитывается по следующей формуле:

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|X_t - \hat{X}_t|}{|X_t|}; \tag{2.1}$$

где X_t - фактическое значение, $\hat{X}_t$ - значение полученное в результате предсказания.

Средняя абсолютная ошибка (MAE) рассчитывается по следующей формуле:

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|X_t - \hat{X}_t|; \tag{2.2}$$

где X_t - фактическое значение, $\hat{X}_t$ - значение полученное в результате предсказания. MAE рассчитывается как среднее абсолютных разностей между наблюдаемым и предсказанным значениями. Она является линейной оценкой, а это значит, что все ошибки в среднем взвешены одинаково.

Среднеквадратичная ошибка (MSE) рассчитывается по следующей формуле:

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(X_t - \hat{X}_t)^2; \tag{2.3}$$

где X_t - фактическое значение, $\hat{X}_t$ - значение полученное в результате предсказания. MSE больше отражает влияние больших отклонений.

2.1.1 Модель Transformer

Трансформеры — относительно новый тип нейросетей, направленный на решение последовательностей с легкой обработкой дальнодействующих зависимостей. На сегодня это самая продвинутая техника в области обработки естественной речи (NLP). Однако недавно их также начали применять в задачах прогнозирования временных рядов.

Одним из ключевых преимуществ трансформеров является их способность улавливать долгосрочные зависимости в данных. Эта особенность может быть особенно полезной при работе с нестационарными временными рядами, где изменения в структуре данных могут происходить на различных временных шкалах. В отличие от рекуррентных нейронных сетей, трансформеры обладают возможностью обрабатывать последовательности независимо друг от друга, что способствует более эффективному извлечению сложных закономерностей.

Поднимаются вопросы [2] о целесообразности использования трансформеров для предсказания временных рядов, так как трансформеры не учитывают порядок входных данных.

Схема трансформера представлена далее на рисунке 1.

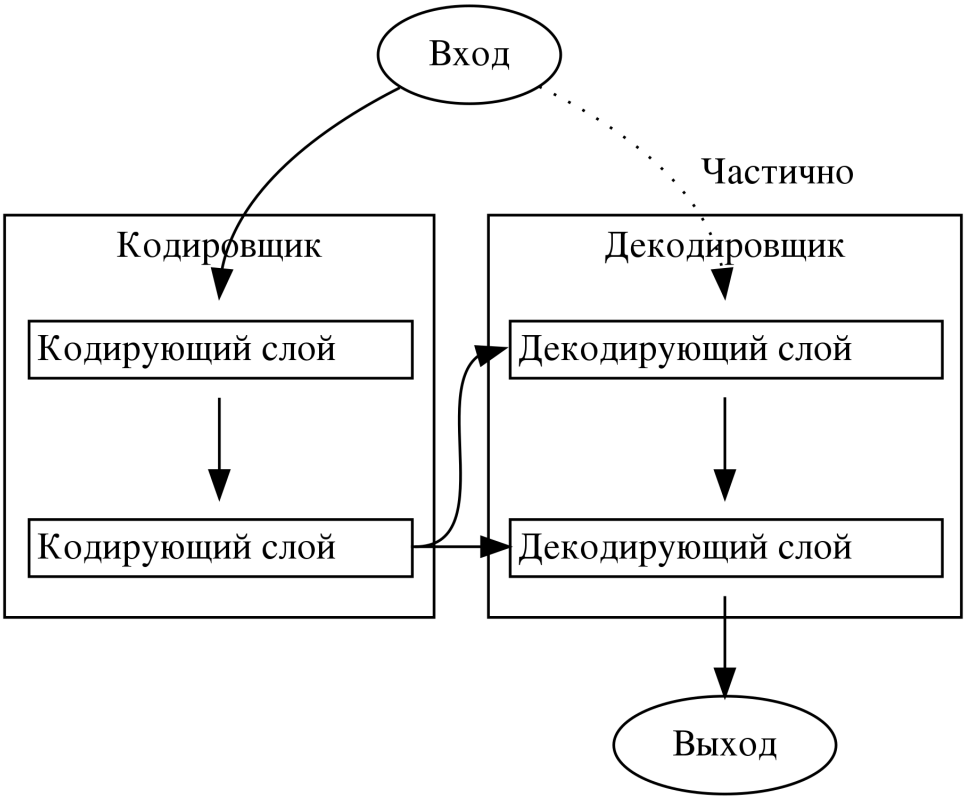

Рисунок 1. Устройство трансформера

2.1.2 Модель DLinear

В статье [1] рассматривается простейший однослойный трансформер DLinear. DLinear представляет собой простую сеть прямого распространения. Математическая интерпретация модели DLinear имеет следующий вид:

$$\hat{X} = W_s X_s + W_t X_t \in \mathbb{R}^{T \times C}; \tag{2.4}$$

где входные данные $X \in \mathbb{R}^{L \times C}$ раскладываются на сезонную и трендовую составляющие. W_s и W_t - линейные слои сезонности и тренда соответственно. L и T - размер окна скользящего среднего и количество предсказанных шагов. Необходимо программно реализовать данные методы. $\hat{X}$ - предсказанное значение, которые мы получили, основываясь на последовательности данных $X_1, X_2, X_3, \ldots, X_L$. Сначала исходные данные разбиваются на трендовую и сезонную составляющие с помощью скользящего среднего. Затем каждая составляющая проходит через два однослойных линейных слоя (2.4), т.е. суммируются два признака, чтобы получить окончательный прогноз.

2.1.3 Скользящее окно

Метод скользящего окна включает постепенное добавление одного значения из тестового набора в обучающий набор за раз, обучение новой модели на обновленном обучающем наборе и использование модели для прогнозирования следующего значения в тестовом наборе. Этот процесс повторяется до тех пор, пока не будет предсказан весь набор тестов. На протяжении всего процесса прогнозирования используется окно одинакового размера. Подробнее об этом методе рассказывается в статье [5]. Схема работы прогнозирования с помощью скользящего окна представлена на рисунке 2

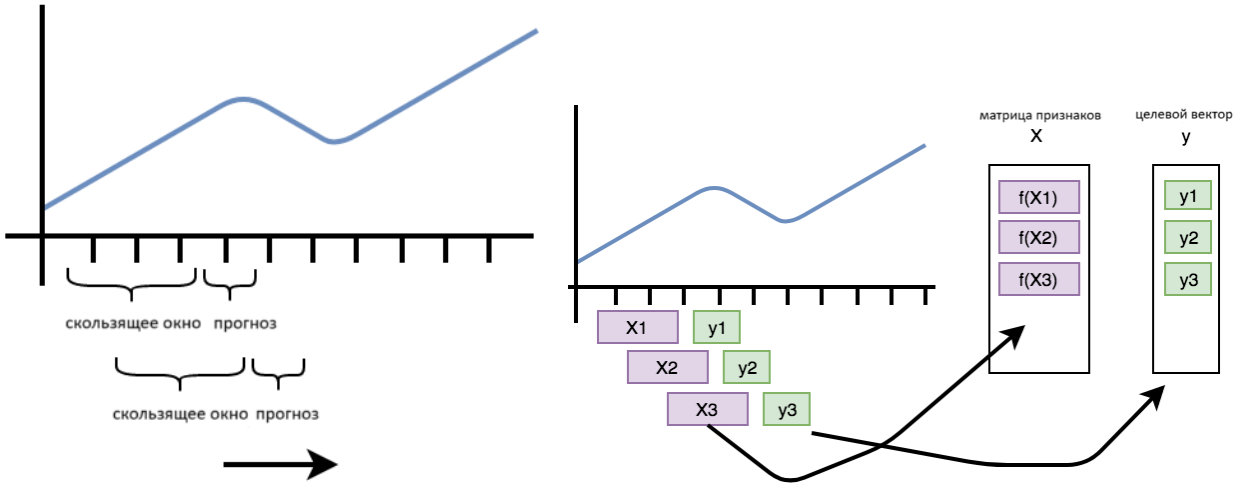

Рисунок 2. Схема работы предсказания с помощью скользящего окна [11].

3 Программная реализация

3.1 Архитектура

Для реализации был выбран язык программирования Python. Python - это мощный и популярный язык программирования и анализа данных, который поддерживает широкий спектр библиотек и инструментов для работы с данными. В данной работе был выбран Python, так как его используют библиотеки NumPy, Pandas и PyTorch, которые предоставляют надежные и эффективные инструменты для работы с данными. Кроме того, Python имеет обширное сообщество разработчиков и научных сотрудников, что позволяет легко найти ресурсы, информацию и поддержку при работе с этим языком.

PyTorch - это кроссплатформенная научная компьютерная библиотека с открытым исходным кодом, разработанная Facebook's AI Research (FAIR). Она основана на языке программирования Python и позволяет использовать вычислительные мощности GPU. PyTorch обеспечивает гибкость и скорость при создании и обучении нейронных сетей, благодаря своему динамическому вычислительному графу.

В данной работе рассматривается проблема прогнозирования временных рядов, которое является одним из ключевых аспектов искусственного интеллекта и машинного обучения.

Для прогнозирования временных рядов используется глубокое обучение и PyTorch, библиотека с открытым исходным кодом для создания и обучения глубоких нейронных сетей.

Процесс создания модели прогнозирования временных рядов с использованием PyTorch состоит из следующих шагов:

1. Подготовка данных;
2. Создание сети;
3. Обучение сети;
4. Проверка и валидация;
5. Прогнозирование;
6. Визуализация.

Для более удобной и последовательной разработки использовался Jupyter Notebook. Jupyter Notebook - это среда разработки, где сразу можно видеть результат выполнения кода и его отдельных фрагментов. Отличие от традиционной среды разработки в том, что код можно разбить на куски и выполнять их в произвольном порядке.

3.2 Реализация DLinear

Перед началом разработки своей модели был проведен обзор библиотеки GluonTS, которая специализируется на работе с временными рядами.

Модель DLinear была реализована на основе класса nn.Module. Листинг класса представлен далее на листинге 3.1.

Листинг 3.1. Программная реализация модели DLinear

```
1  class DLinearModel(nn.Module):
2      def __init__(self, input_size, output_size):
3          super(DLinearModel, self).__init__()
4          self.linear_seasonal = nn.Linear(input_size, output_size)
5          self.linear_trend = nn.Linear(input_size, output_size)
6          self.decomposition = DecompositionLayer(input_size)
7          self.bias = nn.Parameter(torch.zeros(1))
8
9      def forward(self, context):
10         seasonal, trend = self.decomposition(context)
11         seasonal_output = self.linear_seasonal(seasonal.reshape(1, 1, -1))
12         trend_output = self.linear_trend(trend.reshape(1, 1, -1))
13
14         return seasonal_output + trend_output
```

Для реализации декомпозиции входных данных был написан класс на основе того же модуля PyTorch, что и DLinear. Блок декомпозиции DLinear основан на блоке декомпозиции модели Autoformer [2]. Autoformer включает блок декомпозиции в качестве внутренней операции модели, как представлено в архитектуре Autoformer ниже. Как можно видеть, кодировщик и декодировщик используют блок разложения для агрегирования циклической части тренда и постепенного извлечения сезонной части из ряда. Концепция внутренней декомпозиции продемонстрировала свою полезность с момента публикации Autoformer.

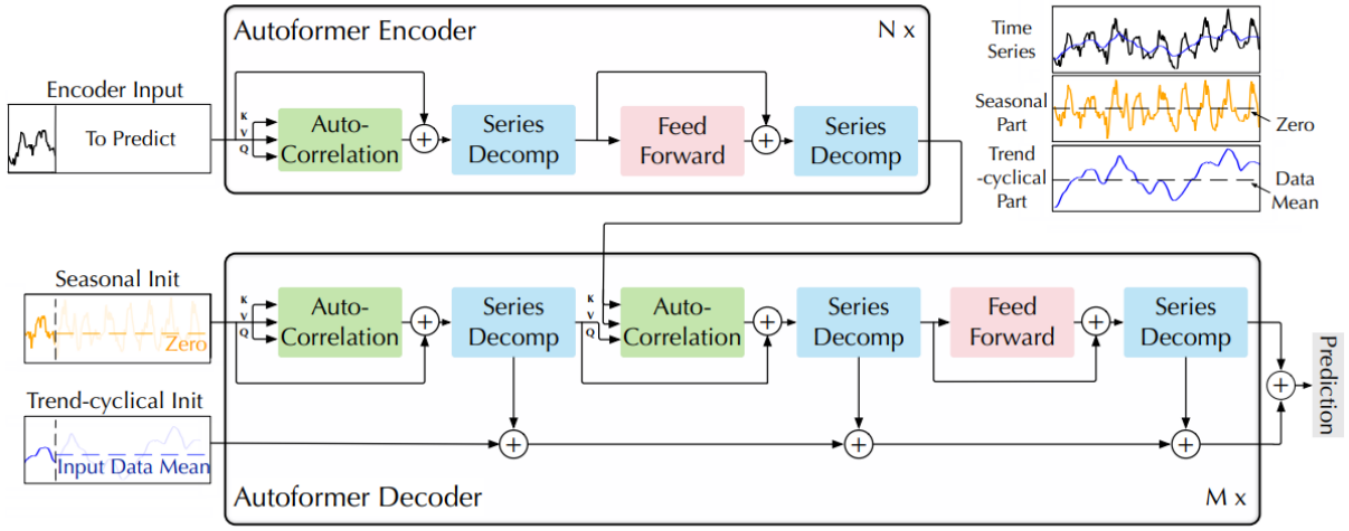

Рисунок 3. Архитектура Autoformer. Взято из статьи [3].

Формальное определение слоя декомпозиции: для входных данных $X \in \mathbb{R}^{L \times C}$ слой декомпозиции возвращает $X_{seasonal}$ и X_{trend}:

$$X_{trend} = AvgPool(Padding(X)); \tag{3.5}$$

$$X_{seasonal} = X - X_{trend}; \tag{3.6}$$

Padding в контексте PyTorch означает добавление дополнительных нулевых значений в начало и/или конец массива при обработке данных с использованием нейронных сетей. Это делается для того, чтобы размерности входных данных соответствовали требованиям модели. AvgPolling - метод усреднения полученных данных для повышения точности получаемых результатов.

Реализация данного слоя на языке Python с использованием PyTorch прдставлена далее на листинге 3.2.

Листинг 3.2. Программная реализация слоя декомпозиции временного ряда

```
1
2 import torch
3 from torch import nn
4
5 class DecompositionLayer(nn.Module):
6     """
7     Returns the trend and the seasonal parts of the time series.
8     """
9
10    def __init__(self, kernel_size):
11        super().__init__()
12        self.kernel_size = kernel_size
13        self.avg = nn.AvgPool1d(kernel_size=kernel_size, stride=1, padding=0) # moving
              average
14
15    def forward(self, x):
16        """Input shape: Batch x Time x EMBED_DIM"""
17        # padding on the both ends of time series
18        num_of_pads = (self.kernel_size - 1) // 2
19        front = x[:, 0:1, :].repeat(1, num_of_pads, 1)
20        end = x[:, -1:, :].repeat(1, num_of_pads, 1)
21        x_padded = torch.cat([front, x, end], dim=1)
22
23        # calculate the trend and seasonal part of the series
24        x_trend = self.avg(x_padded.permute(0, 2, 1)).permute(0, 2, 1)
25        x_seasonal = x - x_trend
26        return x_seasonal, x_trend
```

После получения X_{trend} и $X_{seasonal}$, мы пропускаем их через 2 линейных слоя, для сезонности и тренда соответственно и полученные матрицы складываем, что видно в методе forward класса DLinearModel.

Работа механизма работы DLinear показана в схеме 4, которая описана в статье [2].

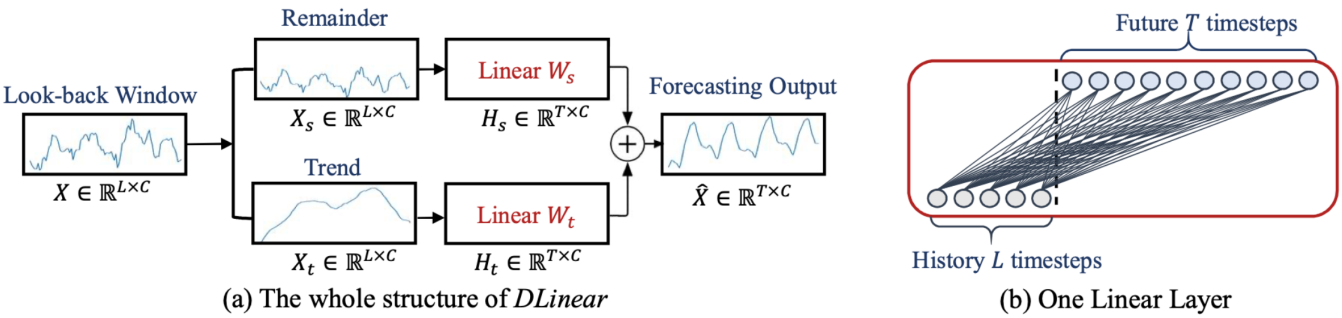

Рисунок 4. Структура DLinear. Взято из статьи [2].

3.3 Обучение с помощью скользящего окна

Механизм обучения модели rolling window был реализовал с помощью Dataloader и Dataset библиотеки PyTorch. Для этого был разработан собственный класс MyDataset, который возвращает данные со смещением на 1 шаг вправо. Листинг 3.3 класса представлен далее. Работа данного метода более подробно определена в теоретической части.

Листинг 3.3. Программная реализация класса MyDataset

```
class MyDataset(TensorDataset):
    def __init__(self, data, window, output):
        self.data = data
        self.window = window
        self.output = output

    def __getitem__(self, index):
        x = self.data[index:index+self.window]
        y = self.data[index+self.window:index+self.window+self.output]
        return x, y

    def __len__(self):
        return len(self.data) - self.window - self.output
```

Обучение проводилось на основе критерия, который основывается на метрике MAE.

4 Тестирование и отладка

4.1 Набор данных

В качестве входных данный было решено использовать датасет ETT-small [12]. Этот набор данных собран за два года из двух отдельных округов Китая. Чтобы изучить степень детализации проблемы прогнозирования временных рядов с длинной последовательностью, создаются различные подмножества: ETTh1 и ETTh2 для шага в один час и ETTm1 для шага в 15 минут. Каждая точка данных состоит из целевого значения «температуры масла» и шести характеристик силовой нагрузки.

4.2 Окно скользящего среднего

Перед начало обучения были проведены тесты (5, 6, 7) обучения с помощью скользящего окна для определения оптимального размера окна скользящего среднего. В качестве временного ряда был выбран столбец/признак HUFL и шаг в один день, начиная с 2016-10-09 00:00:00 и заканчивая 2018-06-26 00:00:00. В качестве прогноза используется вычисление среднего значения наблюдения.

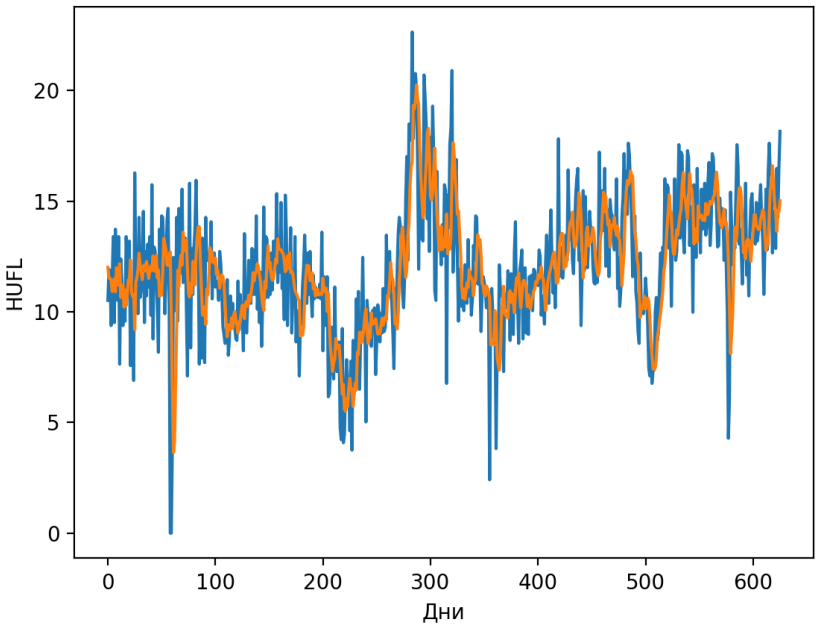

Рисунок 5. Размер окна скользящего среднего равен 4 (синий график - реальное значение временного ряда, оранжевый - прогноз).

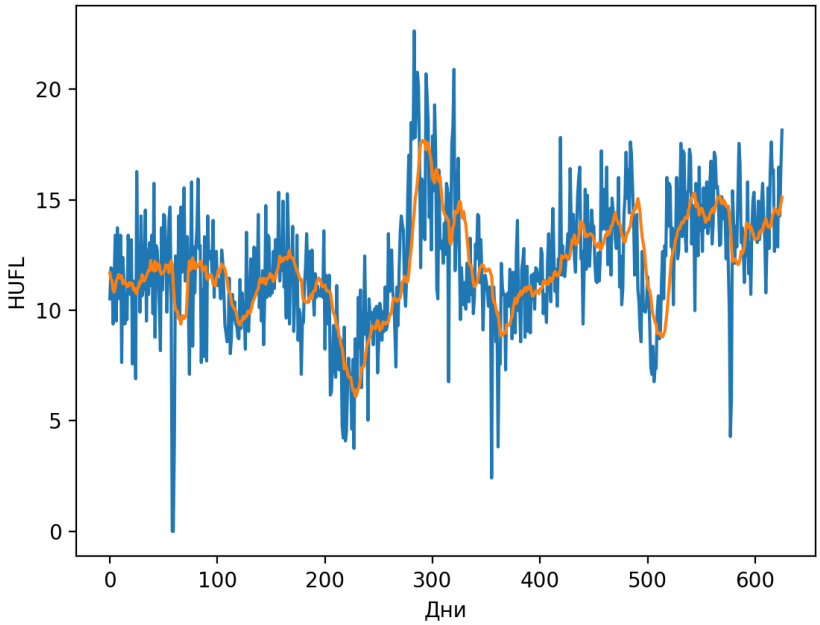

Рисунок 6. Размер окна скользящего среднего равен 13 (синий график - реальное значение временного ряда, оранжевый - прогноз).

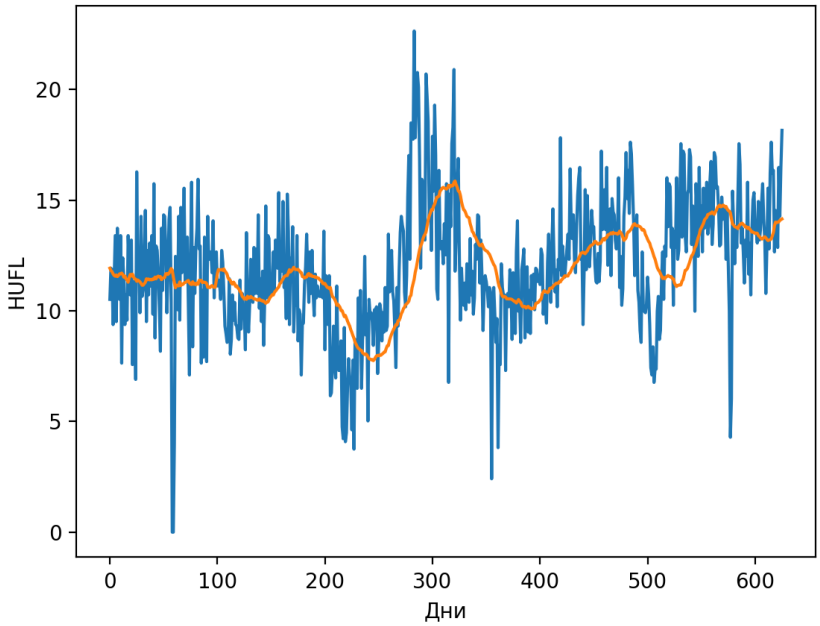

Рисунок 7. Размер окна скользящего среднего равен 40 (синий график - реальное значение временного ряда, оранжевый - прогноз).

На основе результатов, показанных на графиках 5, 6, 7, можно сделать вывод: чем больше окно просмотра тем, модель лучше повторяет уровень данных, но опять же не следует за фактическими движениями вверх и вниз.

5 Вычислительный эксперимент

Для анализа полученной модели и нахождения зависимостей ошибки от параметров прогнозирования были проведены тесты для шага 2, 12 и 23, результаты которых показаны в таблице 1 и рисунках 8 и 9. Тесты проводились на основе временного ряда, который был получен из столбца HUFL набора данных ETTh1.

Окно	MAE(2)	MAE(12)	MAE(23)	MAPE(2)	MAPE(12)	MAPE(23)
5	9,75	9,42	8,06	1,00	1,53	0,92
15	7,37	6,20	5,99	2,04	1,01	1,21
30	6,99	4,28	5,87	1,83	0,98	1,17

Таблица 1. Результаты тестирования(в скобках указан шаг).

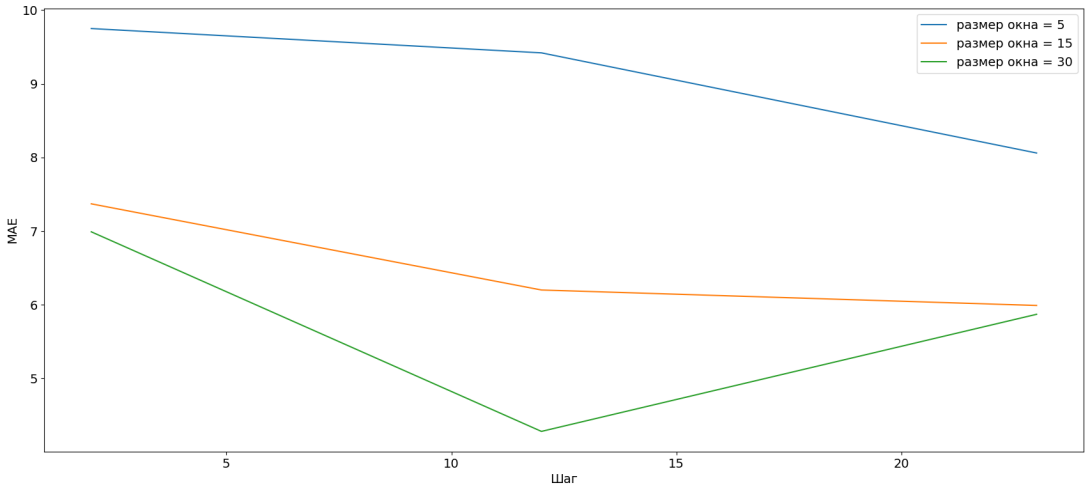

Рисунок 8. Зависимость MAE от шага для моделей обученных на разных размерах скользящего окна.

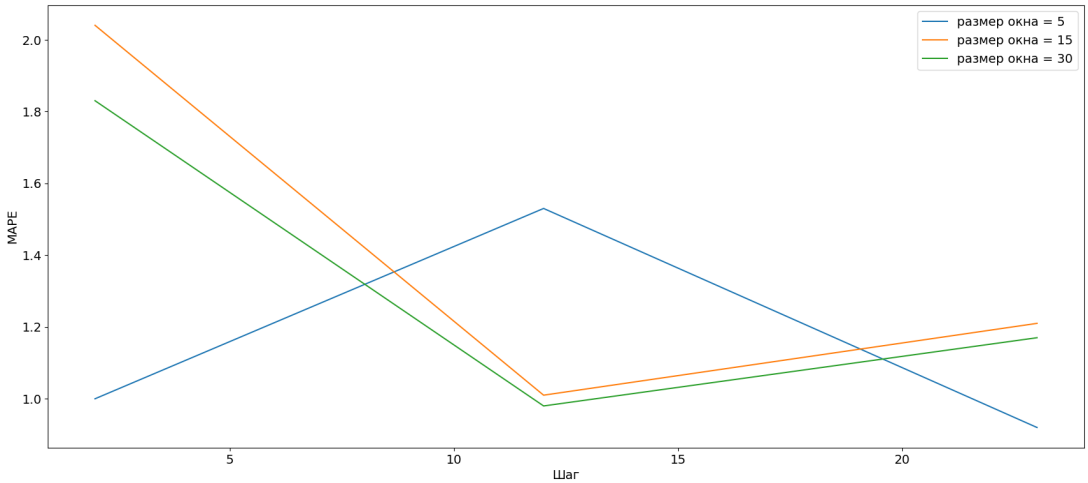

Рисунок 9. Зависимость MAE от шага для моделей обученных на разных размерах скользящего окна.

Проанализировав полученные результаты можно сделать вывод, что нет четкой зависимости точности предсказания от шага и размера скользящего окна. Пример прогнозирования с окном просмотра равным 30 и шагом 12 представлен на рисунке 10.

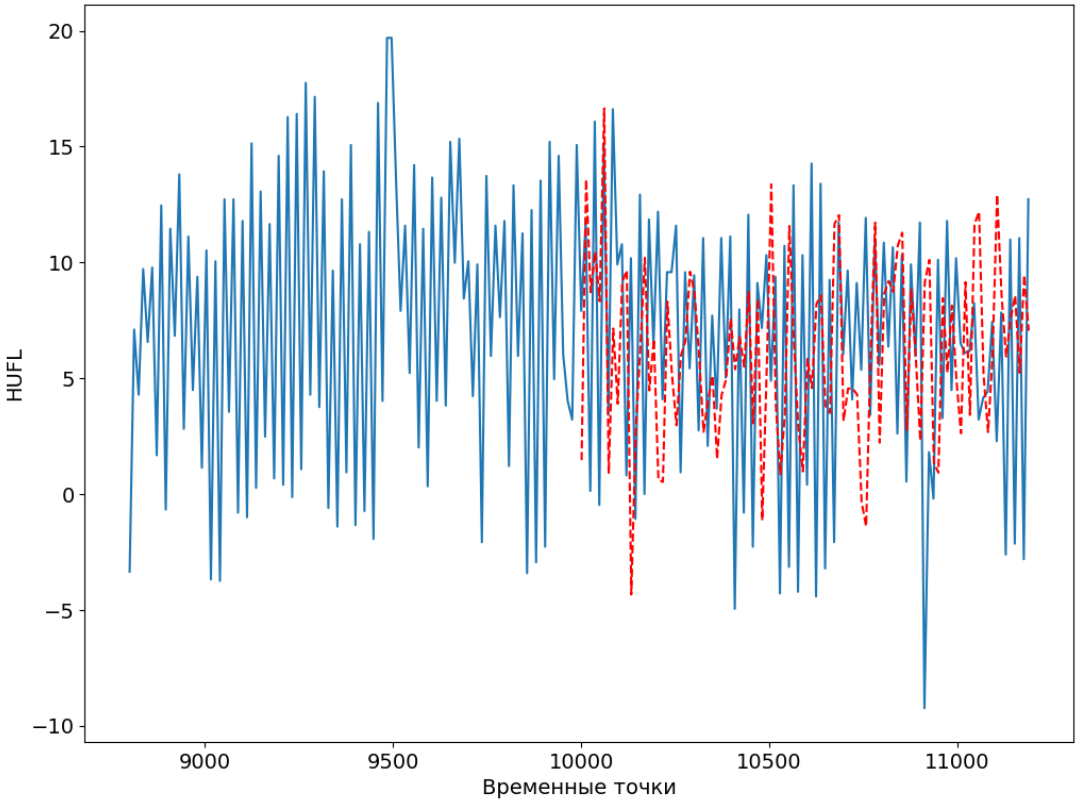

Рисунок 10. Результат прогнозирования временного ряда на основе модели обученной на размере окна равным 30 и шагом 12.

Видно, что ошибка предсказания велика и модель нуждается в усовершенствовании, стоит рассмотреть разные методы оптимизации, метрики и механизмы обучения. Также следует учитывать сразу несколько признаков для предсказания рассматриваемого параметра.

Заключение

В ходе выполнения работы были рассмотрены методы машинного обучения на основе трансформеров. Была реализована модель Dlinear, которая имеет перспективы развития и нуждается в последующей доработке. Также был изучен и реализован метод обучения модели машинного обучения с помощью механизма rolling window.

На примере выбранного датасета были проведены тесты DLinear с различными параметрами, такими как шаг, окно просмотра. Основываясь на полученных результатах можно сделать вывод о том, что применение модели transformer в задачах прогнозирования имеет перспективы. Однако текущая реализация нуждается в существенной доработке и детальном сравнении с более простыми статистическими методами.

Следует отметить, что применение глобальных нейросетевых моделей в случае нестационарных временных рядов остается открытой проблемой. Дальнейшие исследования должны быть направлены на разработку более эффективных методов анализа и прогнозирования нестационарных данных.

В заключение, можно сказать, что глобальные нейросетевые модели обладают значительным потенциалом для анализа и прогнозирования временных рядов, включая нестационарные временных ряды. Однако, для достижения наилучших результатов, необходимо учитывать особенности каждой конкретной задачи и применять соответствующие методы обработки данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Spyros Makridakis Evangelos Spiliotis Vassilios Assimakopoulos Artemios-Anargyros Semenoglou Gary Mulder, Nikolopoulos Konstantinos. Statistical, machine learning and deep learning forecasting methods: Comparisons and ways forward // Journal of the Operational Research Society. 2023. Т. 74, № 3. С. 840–859. URL: https://doi.org/10.1080/01605682.2022.2118629.

2 Ailing Zeng Muxi Chen Lei Zhang Qiang Xu. Are Transformers Effective for Time Series Forecasting? // Cornell University. 2022. URL: https://doi.org/10.48550/arXiv.2205.13504.

3 Haixu Wu Jiehui Xu Jianmin Wang Mingsheng Long. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting // Cornell University. 2021. URL: https://doi.org/10.48550/arXiv.2106.13008.

4 Bandara Kasun, Bergmeir Christoph, Smyl Slawek. Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach // Expert Systems with Applications. 2020. Т. 140. С. 112896. URL: https://www.sciencedirect.com/science/article/pii/S0957417419306128.

5 Hewamalage H. Ackermann K. Bergmeir C. Forecast evaluation for data scientists: common pitfalls and best practices. // Data Mining and Knowledge Discovery. 2023. Т. 37. С. 788–832. URL: https://doi.org/10.1007/s10618-022-00894-5.

6 Tian Zhou Ziqing Ma Qingsong Wen Xue Wang Liang Sun Rong Jin. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting // Cornell University. 2022. URL: https://doi.org/10.48550/arXiv.2201.12740.

7 Zhe Li Shiyi Qi Yiduo Li Zenglin Xu. Revisiting Long-term Time Series Forecasting: An Investigation on Linear Mapping // Cornell University. 2023. URL: https://doi.org/10.48550/arXiv.2305.10721.

8 Wei Li Xiangxu Meng Chuhao Chen Jianing Chen. Mlinear: Rethink the Linear Model for Time-series Forecasting // Cornell University. 2023. URL:

https://doi.org/10.48550/arXiv.2305.04800.

9 Ronghao Ni Zinan Lin Shuaiqi Wang Giulia Fanti. Mixture-of-Linear-Experts for Long-term Time Series Forecasting // Cornell University. 2023. URL: https://doi.org/10.48550/arXiv.2312.06786.

10 Zhang G.Peter, Qi Min. Neural network forecasting for seasonal and trend time series // European Journal of Operational Research. 2005. T. 160, № 2. C. 501–514. Decision Support Systems in the Internet Age. URL: https://www.sciencedirect.com/science/article/pii/S0377221703005484.

11 Rolling/Time series forecasting // tsfresh. URL: https://tsfresh.readthedocs.io/en/latest/text/forecasting.html.

12 Electricity Transformer Dataset (ETDataset) // GitHub. URL: https://github.com/zhouhaoyi/ETDataset/tree/main.

Выходные данные

Новокшанов Е.А.. Прогностическая способность глобальных нейросетевых моделей в случае нестационарных временных рядов по дисциплине «Модели и методы анализа проектных решений». [Электронный ресурс] — Москва: 2023. — 23 с. URL: `https://sa2systems.ru:88` *(система контроля версий кафедры РК6)*

Постановка: *Доцент, к.ф.-м.н., Соколов А.П.*

Решение и вёрстка: *студент группы РК6-76Б, Новокшанов Е.А.*

2023, осенний семестр