



Прогнозирование динамики параметров распространения новых вирусов на основе методов машинного обучения

Выполнил: студент гр.5040103/10301 Курлевский А.А.

Научный руководитель: доцент ВШТМиМФ, к.ф.-м.н. Ле-Захаров А.А.

Консультант: ассистент ВШТМиМФ Перец Д.С.

Цель

- Спрогнозировать динамику параметров распространения COVID-19, используя методы машинного обучения

Задачи

- Провести анализ существующих методов прогнозирования временных рядов.
- Предложить алгоритм машинного обучения для нахождения зависимости между значениями параметров распространения и характеристики динамического баланса(ХДБ).
- На основе сделанного прогноза оценить количество болеющих людей, используя принцип динамического баланса.

Актуальность исследования

Статистика заболеваемости по России на момент 13.05.2023

• **605**

Человека госпитализировано
за сутки

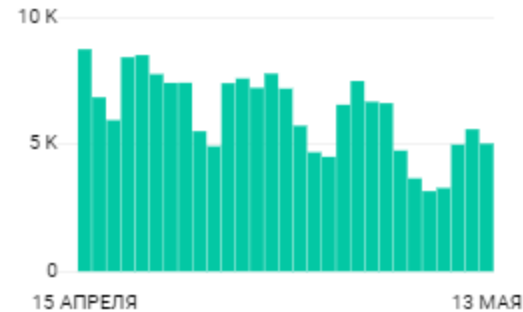


• **5 022**

Человека выздоровело
за сутки

22 316 817

Человек выздоровело



• **3 027**

Выявлено случаев
за сутки

22 895 380

Выявлено случаев

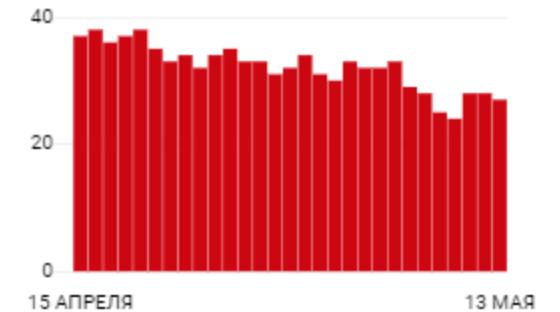


• **27**

Человек умерло
за сутки

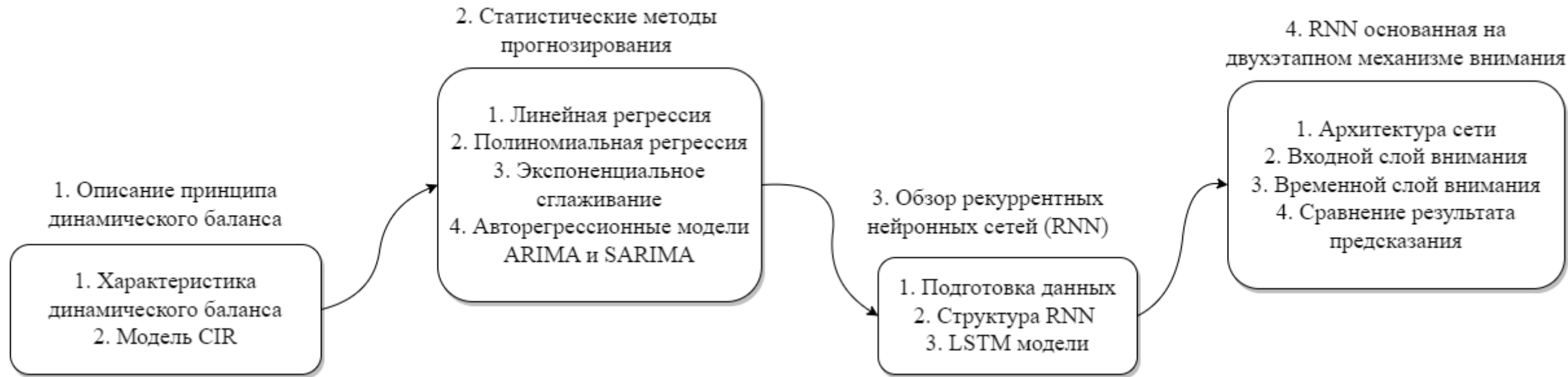
398 685

Человек умерло

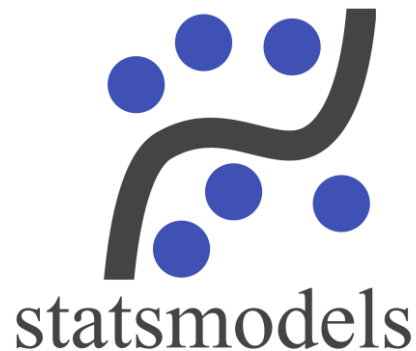


Методы прогнозирования позволяют оценить нагрузку на систему здравоохранения и помогают принять решение для профилактических мер.

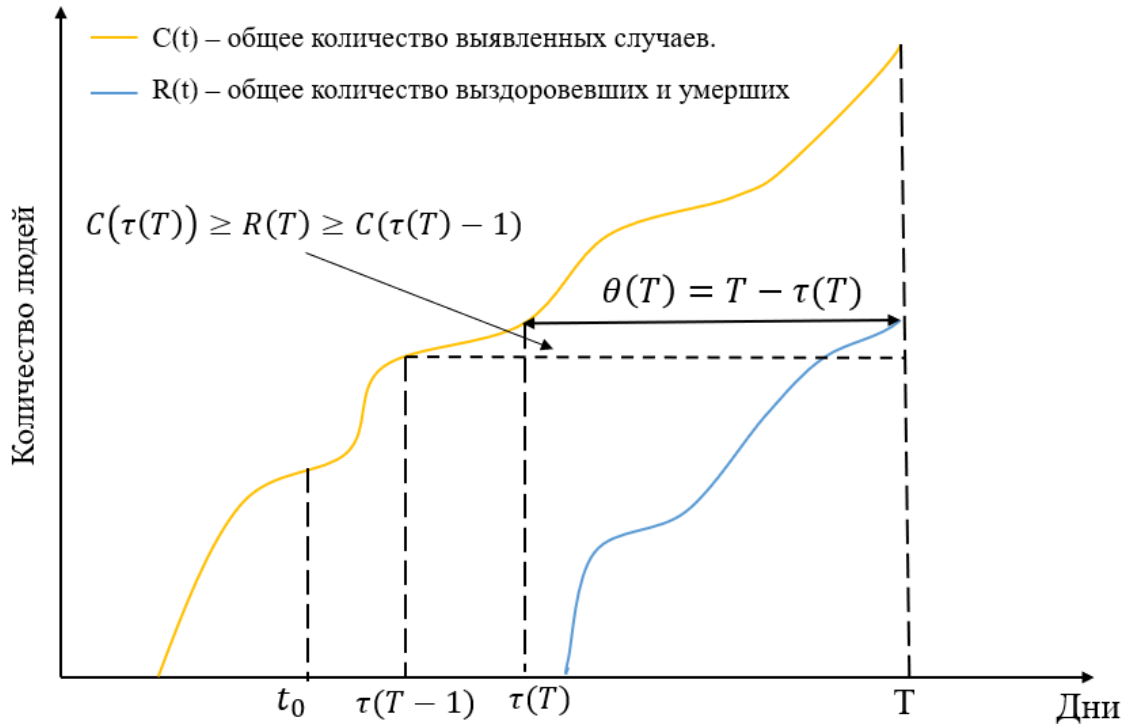
Структура исследования



Использованные технологии



Характеристика динамического баланса (ХДБ)



Задача:

$$\begin{cases} \min_{t_0 \leq t \leq T} t \\ C(t) \geq R(t) \end{cases}$$

$\tau(T)$ - решение задачи.

Теорема (принцип динамического баланса). Пусть заданы значения $t_0 \geq 0$ и $T > t_0$, такие, что $R(T) > C(t_0) > 1$. Тогда:

$$C(\tau(T)) \geq R(T) \geq C(\tau(T) - 1)$$

$\theta(T) = T - \tau(T)$ - характеристика динамического баланса

Следствие: В условиях теоремы функция $R(T)$ представима в виде

$$R(T) = \lambda_T C(\tau(T) - 1) + (1 - \lambda_T) C(\tau(T)), \lambda \in [0, 1]$$

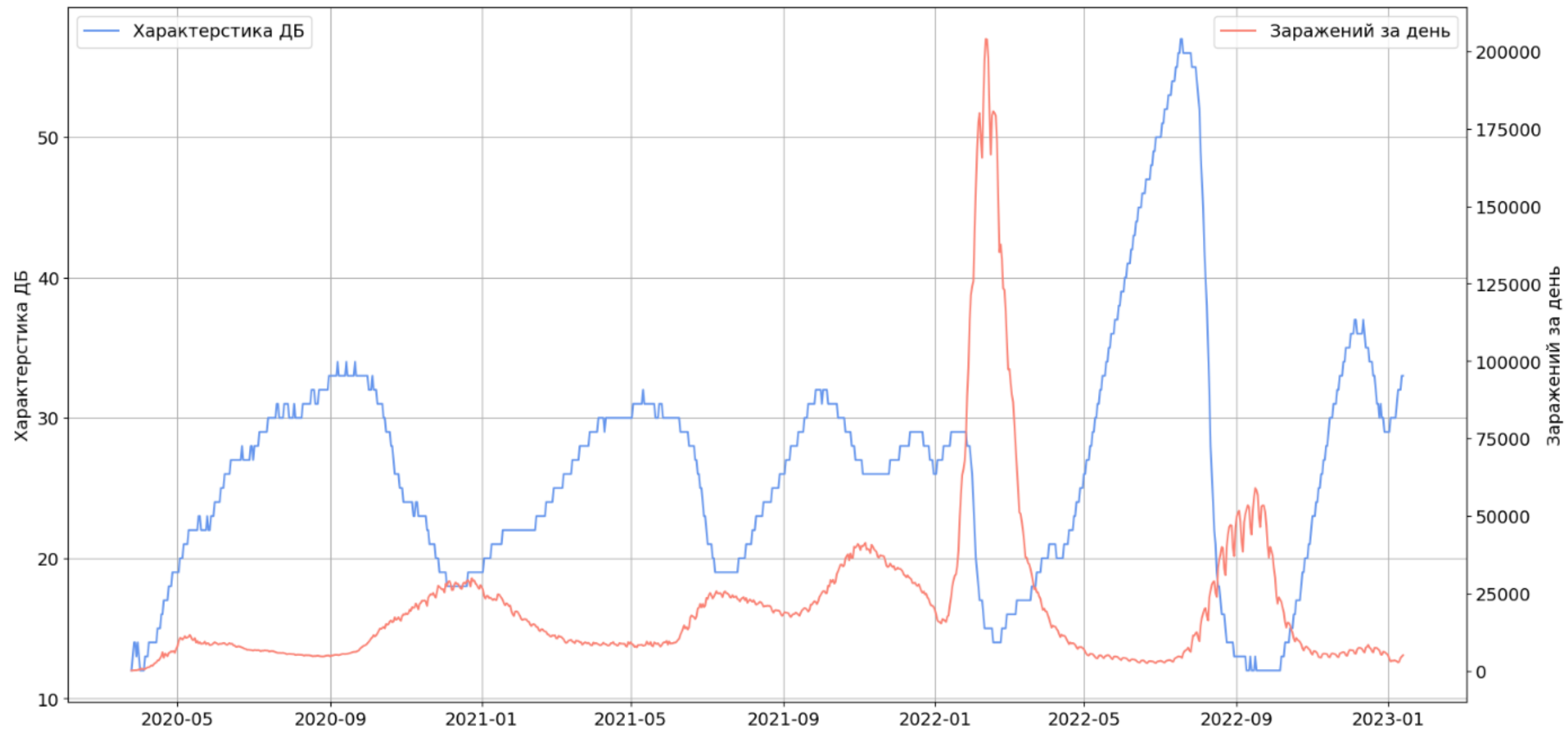
Захаров В. В., Балыкина Ю. Е. Балансовая модель эпидемии COVID-19 на основе процентного прироста
 //Информатика и автоматизация. – 2021. – Т. 20. – №. 5. – С. 1034-1064.

Балансовая модель эпидемии COVID-19


- $C(t)$ – общее количество подтвержденных случаев, $I(t)$ – количество инфицированных людей случаев, $R(t)$ – количество выздоровевших и умерших, $r(t)$ – процентный прирост общего количества выявленных случаев
- $C(t) = I(t) + R(t)$ для любого момента времени

- Балансовая модель CIR:
$$\begin{cases} C(t) = \left(1 + \frac{r(t)}{100}\right) C(t-1) \\ I(t) = C(t) - R(t) \\ R(t) = \lambda_t(C(\tau(t)) - 1) + (1 - \lambda_t)C(\tau(t)) \end{cases}$$

Захаров В. В., Балыкина Ю. Е. Балансовая модель эпидемии COVID-19 на основе процентного прироста
//Информатика и автоматизация. – 2021. – Т. 20. – №. 5. – С. 1034-1064.



Основные методы прогнозирования временных рядов



Статистические модели

- ARIMA, SARIMA
- Экспоненциальное сглаживание
- Регрессия
- И другие

Модели с использованием

методов машинного обучения

- Рекуррентные нейронные сети: LSTM, GRU
- Деревья решений
- Метод опорных векторов
- И другие

Статистические модели

1. Линейная регрессия с несколькими предикторами:

$$x_t = \alpha_1 x_t^1 + \alpha_2 x_t^2 + \alpha_3 x_t^3$$

2. Полиномиальная регрессия:

$$x_t = \alpha_1 t + \alpha_2 t^2 + \dots + \alpha_n t^n$$

3. Метод тройного экспоненциального сглаживания:

$$L_t = \alpha \frac{x_t}{S_{t-s}} + (1 - \alpha)(L_{t-1} + T_{t-1}),$$

$$T_t = \beta(L_t - L_{t-1}) + (1 - \beta)T_{t-1}$$

$$S_t = \gamma \frac{x_t}{L_t} + (1 - \gamma)S_{t-s}, \quad x_{t+h} = (L_t + hT_t)S_{t-s+h}$$

α, β, γ -параметры сглаживания; s -количество наблюдений, составляющих сезонную вариацию; x_{t+h} -предсказание для периода $t + h$

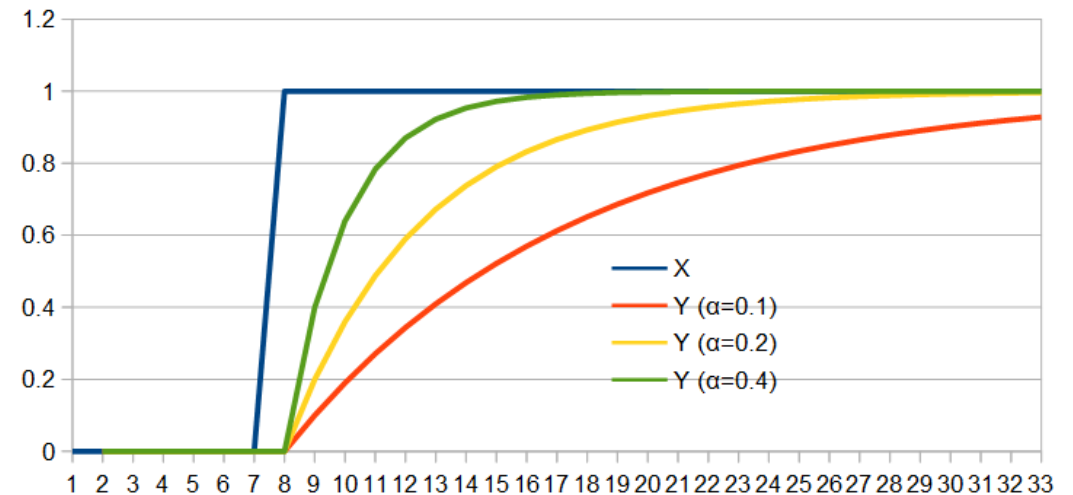


Иллюстрация работы простого сглаживания

4. Авторегрессионная модель $ARIMA(p,q,d)$:

$$\Delta^d x_t = c + \sum_{i=1}^p \alpha_i \Delta^d x_{t-i} + \sum_{j=1}^q b_j \varepsilon_{t-j} + \varepsilon_t$$

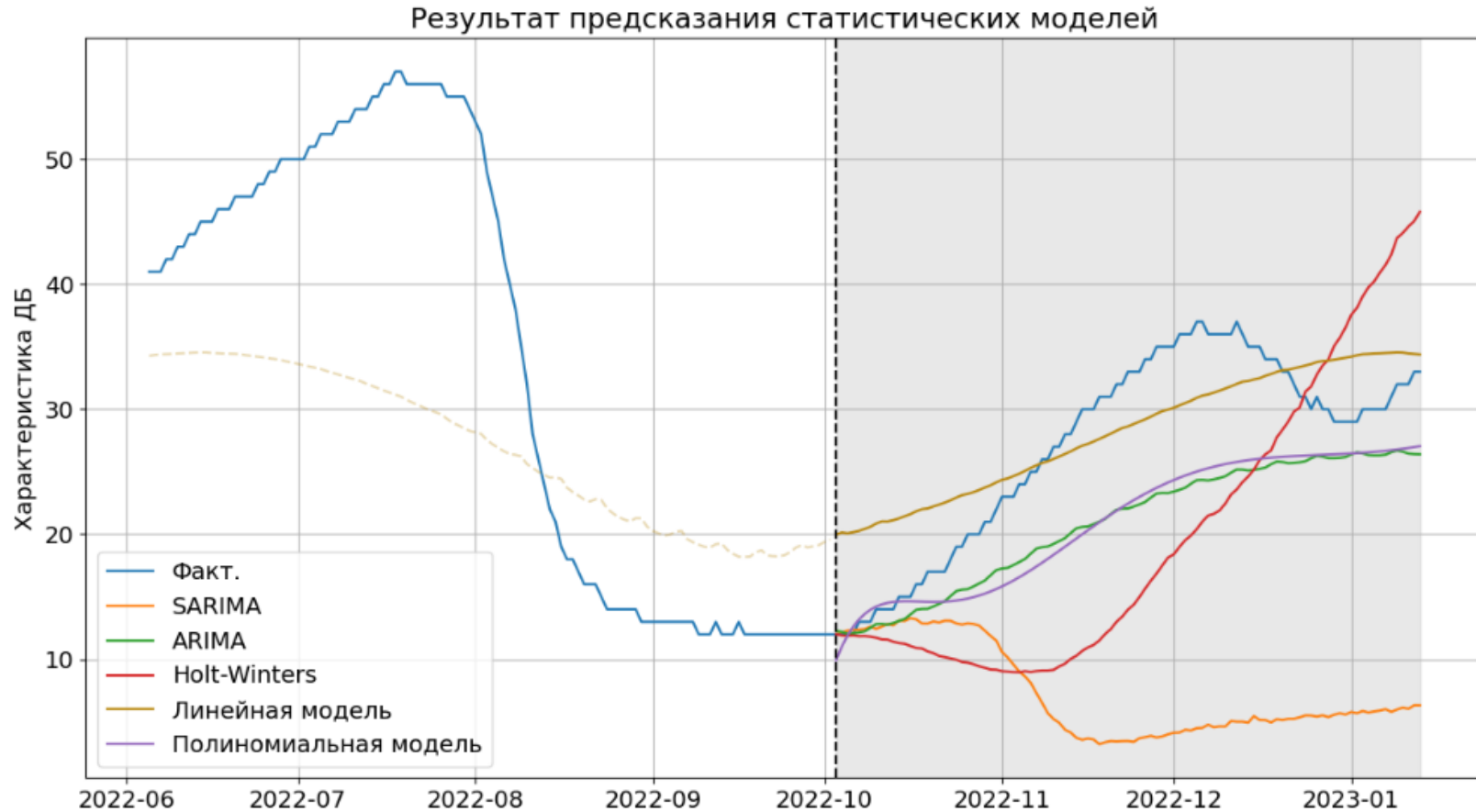
Где p -количество наблюдений в авторегрессионной части; d -порядок разности, которую нужно применить, чтобы привести ряд к стационарному виду; q -количество прошлых ошибок прогноза; ε_t -белый шум; Δ -оператор разности; α_i , b_j и c параметры модели.

5. Сезонная модификация авторегрессионной модели – $SARIMA(p, d, q)_{P,D,Q,S}$

$$\Delta^D X_t = c + \sum_{i=1}^P A_{is} \Delta^D X_{t-is} + \sum_{j=1}^Q B_{js} \varepsilon_{t-js} + \varepsilon_t$$

Параметры аналогичны, добавляется порядок сезонности s . Результат является сумма сезонной части и несезонной

Результат предсказания статистических моделей



Модель	MAPE, %	R ²
Линейная	18.03	0.67
Полиномиальная	21.15	0.13
ARIMA	21.27	0.11
Holter-Winters	38.35	-1.55
SARIMA	64.76	-7.64

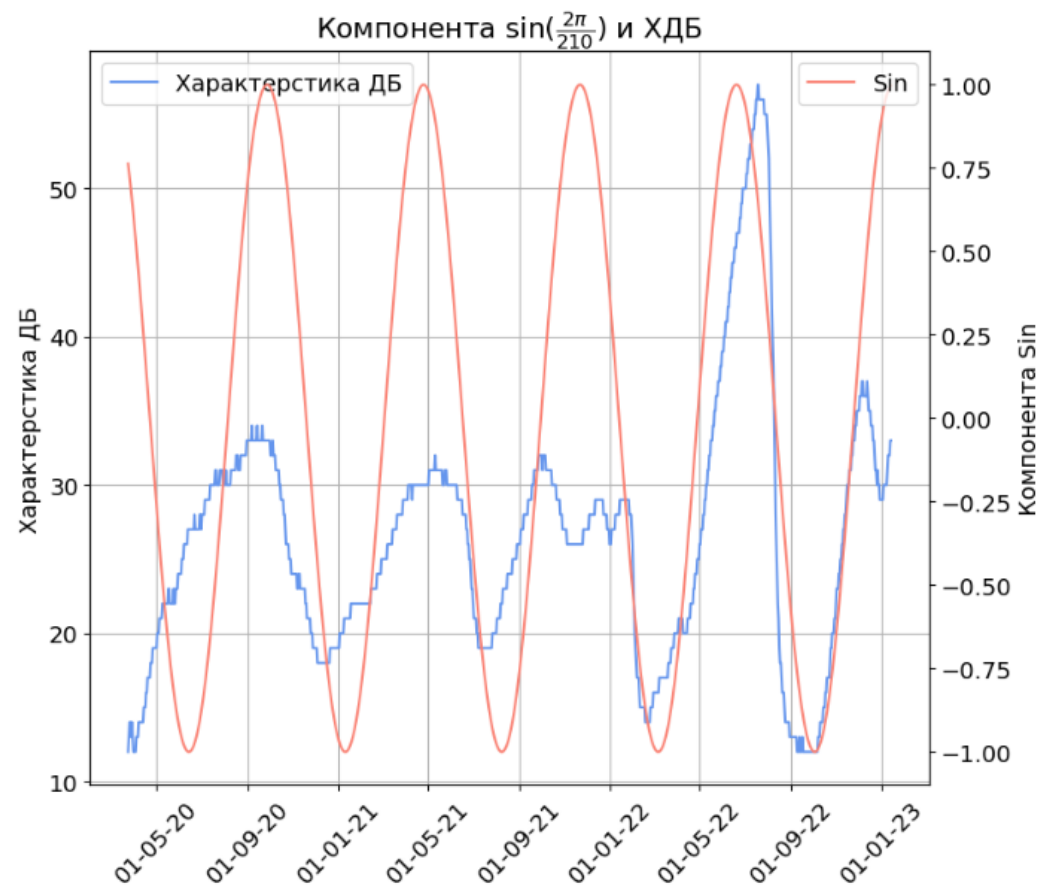
Подготовка данных для обучения нейронных сетей

Этапы подготовки данных:

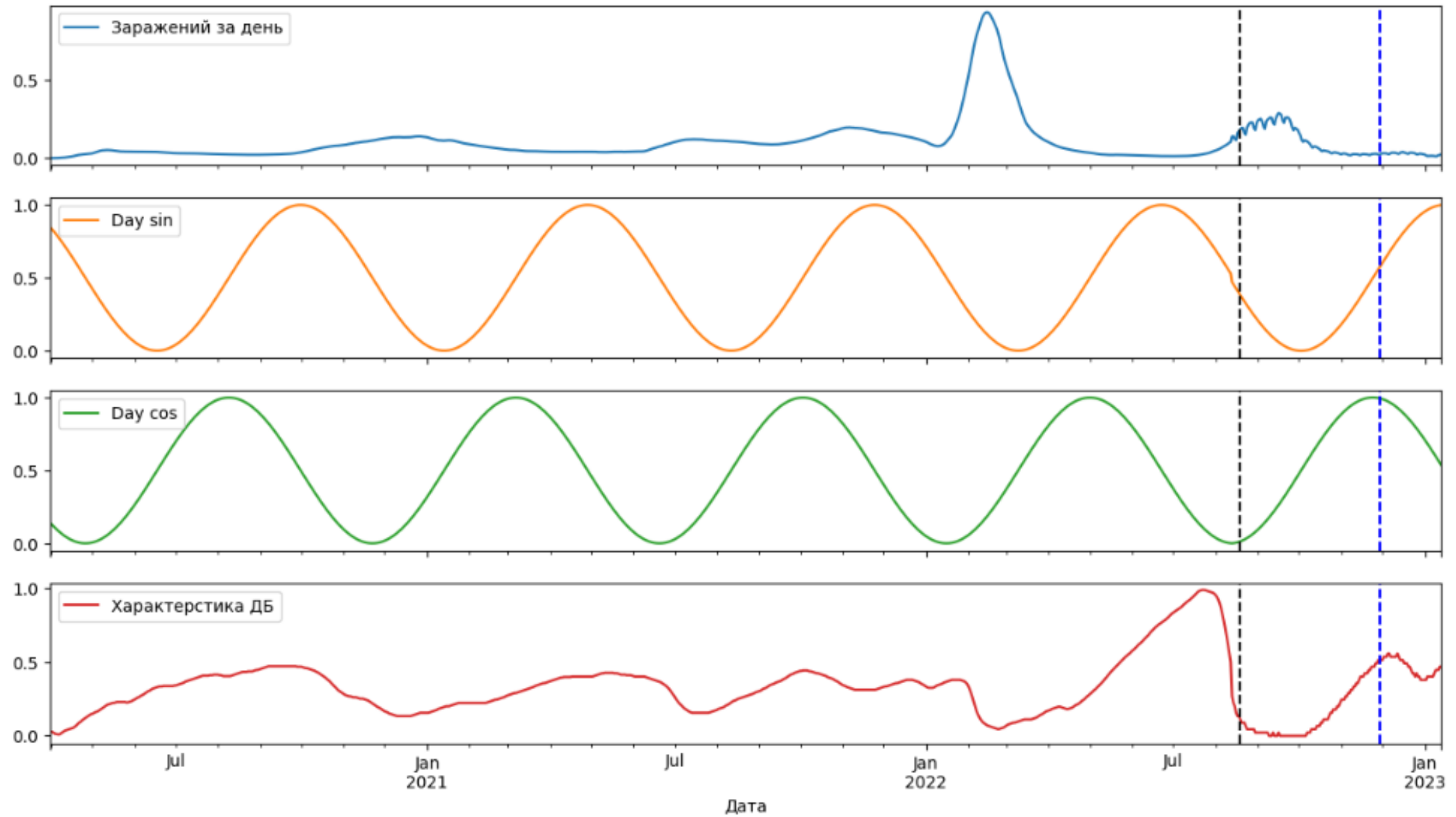
1. Разобьем ряд на тренировочную, валидационную и тестовую часть в пропорции 85%, 10%, 5% соответственно
2. Характер исследуемой величины имеет ступенчатый вид. Сгладим ряд, используя скользящее среднее с окном в 7 дней.
3. Необходимо, чтобы значения лежали в одном диапазоне. Для этого нормализуем их:

$$x_i^{scaled} = \frac{x_i - \min(X)}{\max(X) - \min(X)}$$

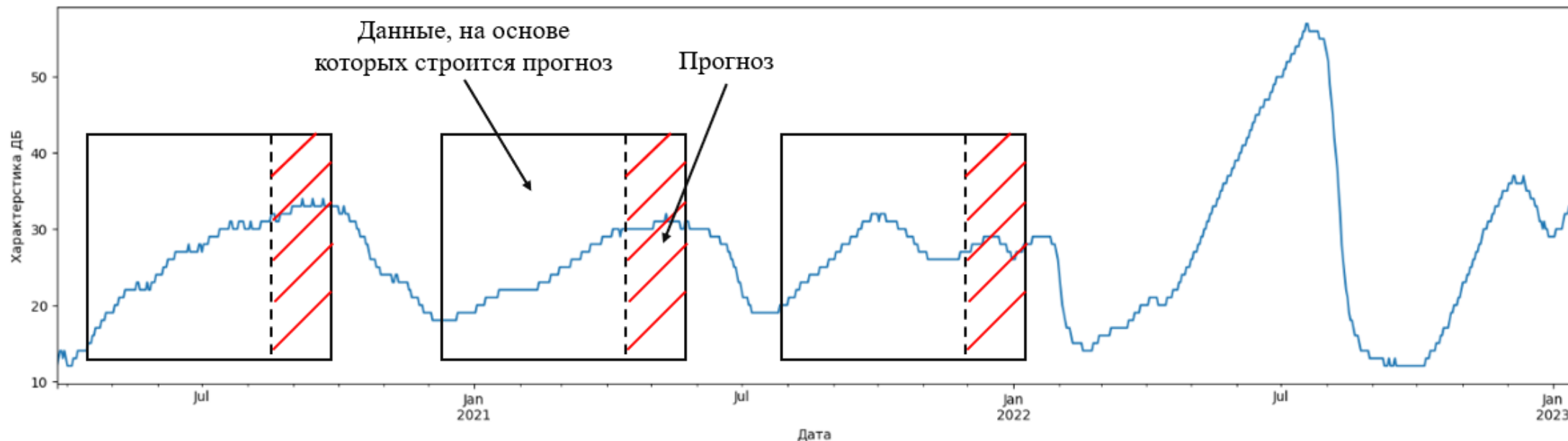
4. Полезно знать информацию о частоте исследуемой динамики. Предположим, что сезон равен 210 дней. Результат быстрого преобразования Фурье подтверждает сделанное предположение. Поэтому, кроме имеющейся информации, на вход будем подавать компоненты $\sin(\frac{2\pi}{210}t)$ и $\cos(\frac{2\pi}{210}t)$



Данные для обучения



Подготовка данных для обучения нейронных сетей



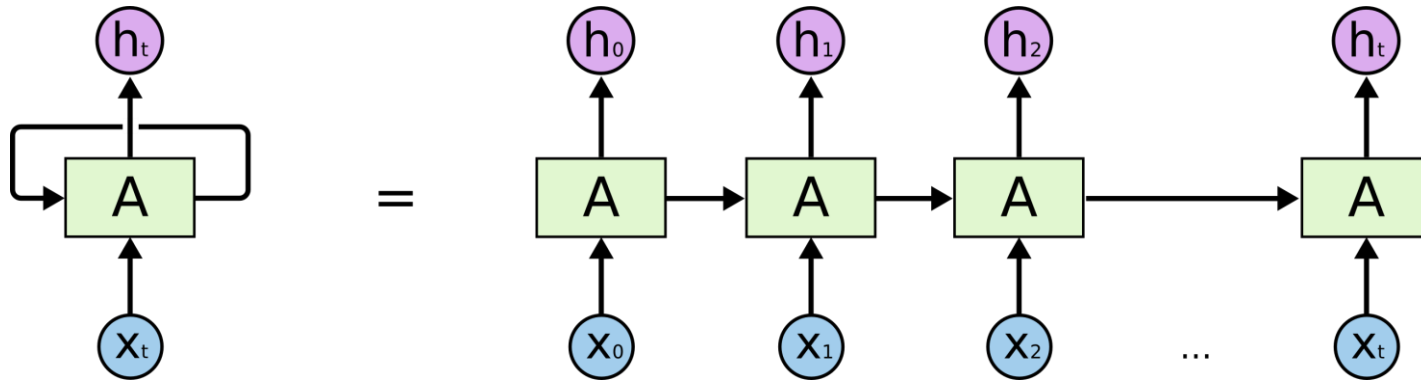
На вход модели подается набор окон. Окно характеризуется 2-мя параметрами: количество наблюдений, на основе которых строится прогноз (ширина окна), горизонт предсказания. Окна случайным образом перемешиваются (структура ряда внутри окна сохраняется) и разбиваются на группы. Итого модель на вход получает 3-х мерную структуру данных: (количество групп, ширина окна, количество характеристик)

Метрики качества предсказания

- $MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{y_i}$ - средняя абсолютная ошибка в процентах (количественная мера)
- $R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$ - коэффициент детерминации (показывает на сколько хорошо предсказан тренд)

y_i - фактическое значение, \hat{y}_i - предсказанное значение, \bar{y} - среднее значение

Рекуррентная нейронная сеть



Развернутая схема рекуррентной сети. A – блок нейронной сети, x_i – входные значения, h_i – выходные значения.

В простейшем случае выход нейронной сети следующим образом:

$$h_i = \tanh(W \cdot [h_{i-1}, x_i] + b)$$

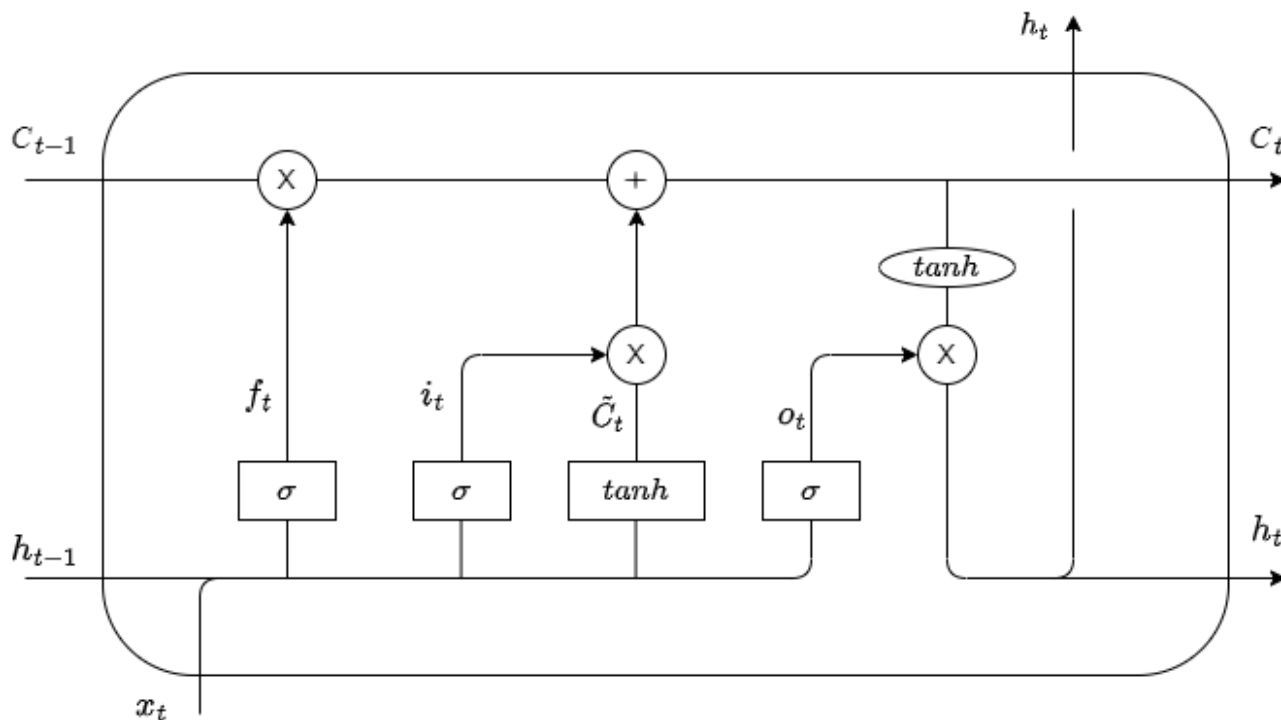
Достоинства:

- Способны запоминать историю данных.
- Адаптируются под множество задач

Недостатки:

- При обучении происходит затухание градиентов ошибок. Из-за этого параметры сети изменяются на слишком малые значения и модель плохо обучается
- Из-за большого количества параметров невозможно объяснить поведение модели

Сеть LSTM



Порядок выполнения вычислений:

1. Определяем какое количество информации можно убрать

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

2. Определяем какая часть новой информации будет содержаться в состоянии ячейки

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

3. Пересчитываем новое состояние ячейки:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

4. Вычисляем скрытое состояние ячейки, которое будем получать на выходе:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

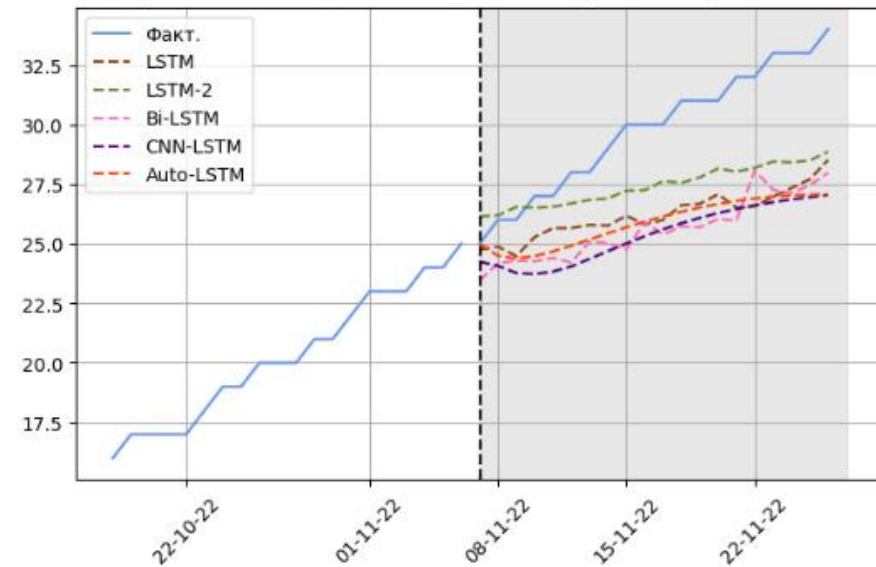
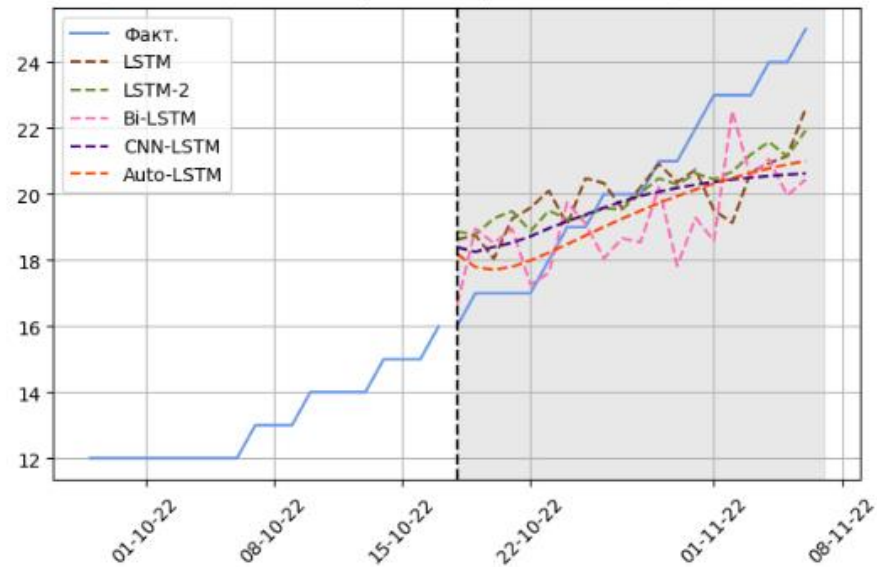
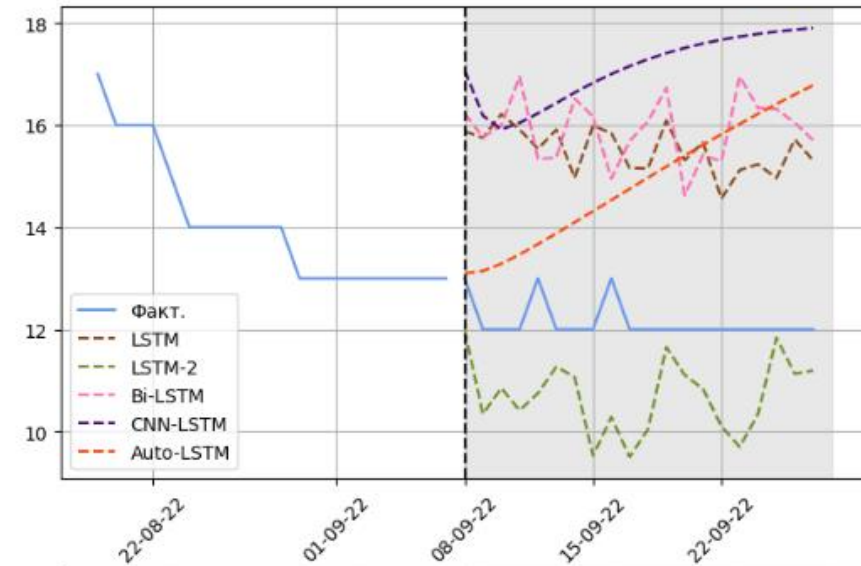
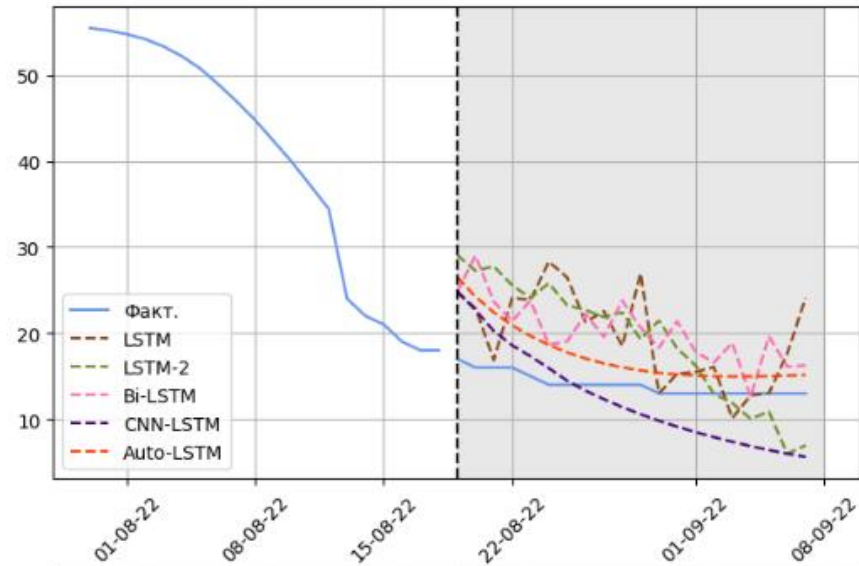
$$h_t = o_t * \tanh(C_t)$$

Виды LSTM сетей

1. Stacked. Это сети состоящие из нескольких LSTM слоев. В работе показан результат для однослойных и двуслойных моделей. Для большего количества слоев не удалось добиться качественного результата.
2. Bi-LSTM. Двунаправленные LSTM сети. Информация при обучении идет как слева направо, так и справа налево.
3. Encoder-Decoder. Модели состоят из двух блоков. В блоке Encoder информация кодируется, при этом извлекаются важные признаки. В блоке Decoder информация расшифровывается. В работе рассмотрена модель CNN-LSTM и DA RNN. Про последнюю модель более подробно дальше
4. Auto-LSTM. Авторегрессионная LSTM модель. Для предсказания горизонта прогнозирования модель использует уже предсказанные значения в качестве входных данных.

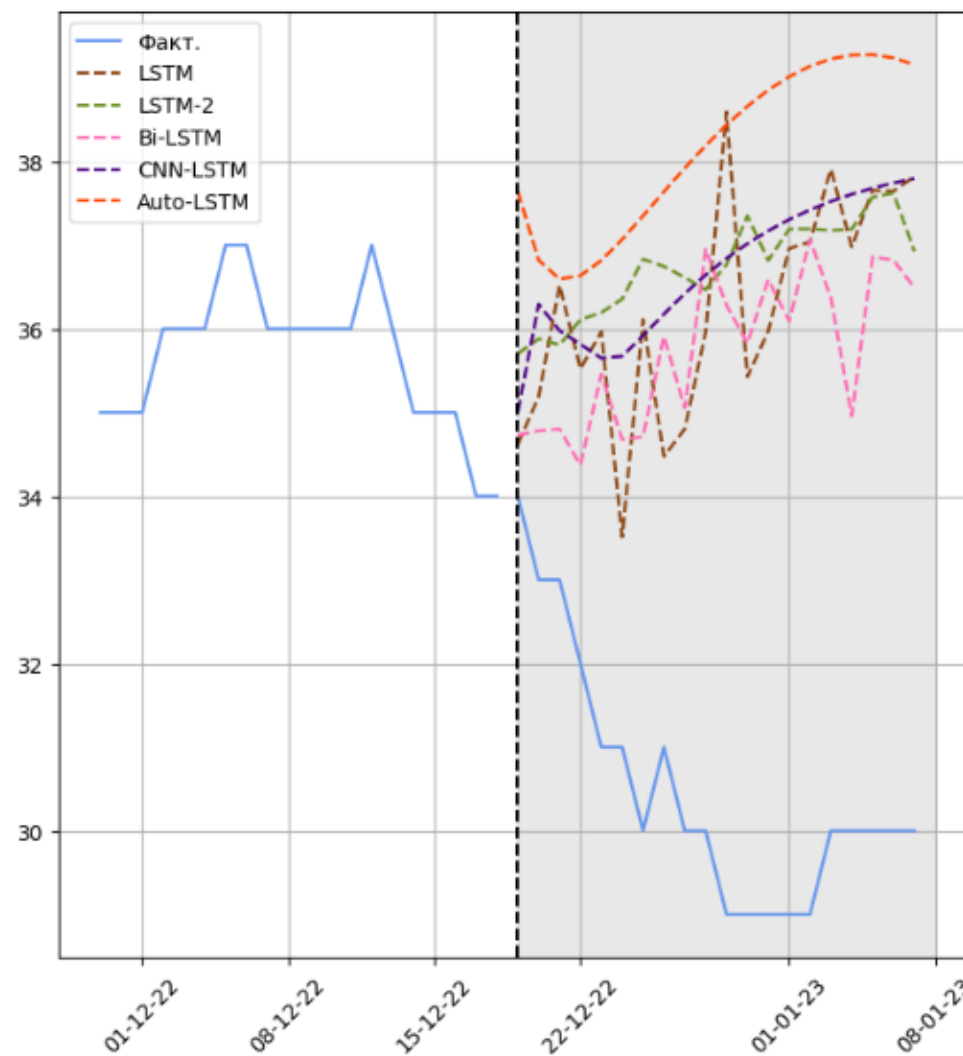
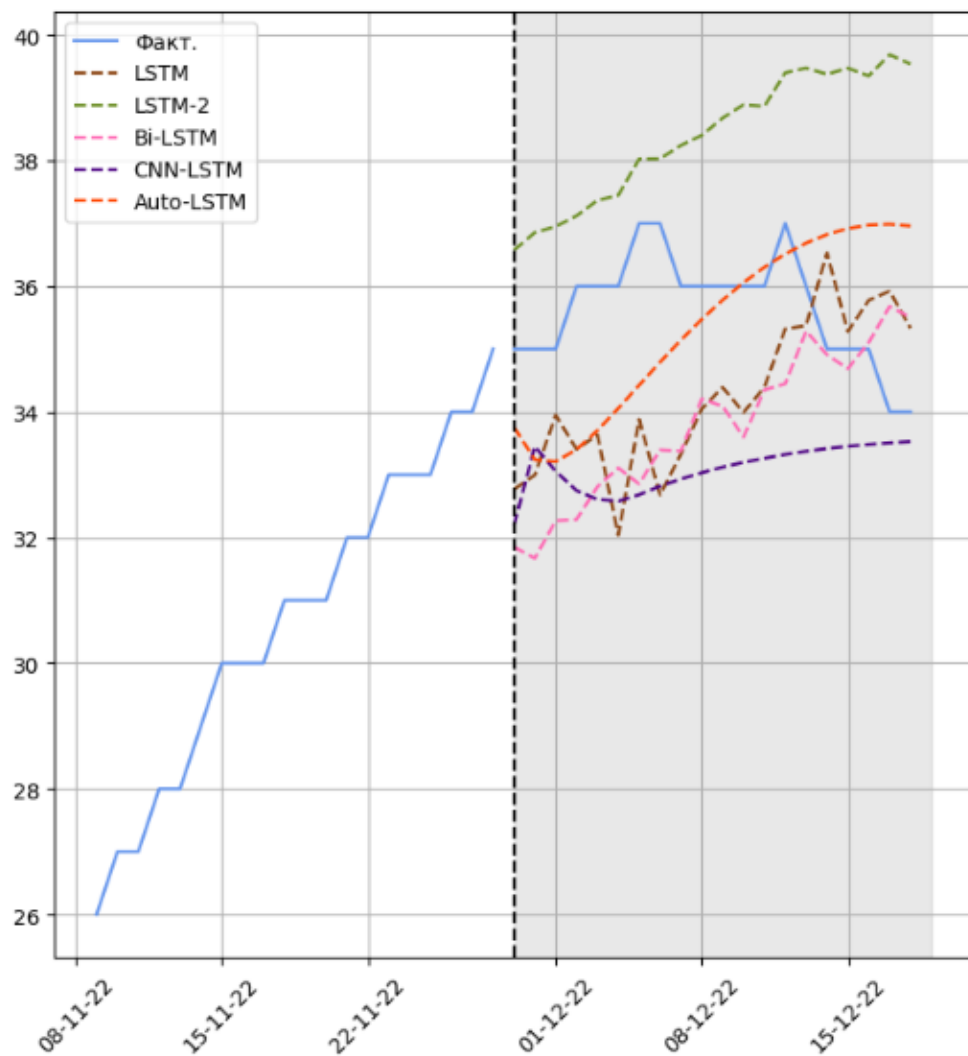
Результат предсказания на валидационном наборе

Предсказание ХДБ с помощью RNN моделей на валидационном наборе



Результат предсказания на тестовом наборе

Предсказание ХДБ с помощью RNN моделей на тестовом наборе



Метрики качества прогноза

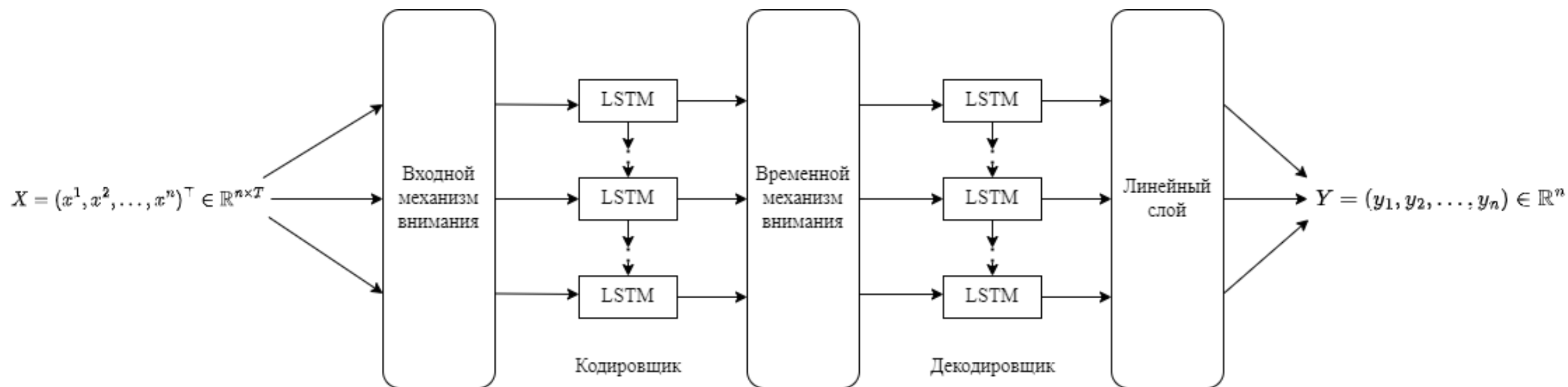
На валидационном наборе

Модель	MAPE, %	R^2
LSTM-2	13.89	-2.07
Auto-LSTM	15.79	-2.39
Bi-LSTM	17.46	-1.87
LSTM	18.25	-2.93
CNN-LSTM	23.04	-2.58

На тестовом наборе

Модель	MAPE, %	R^2
Bi-LSTM	12.03	-5.20
CNN-LSTM	12.63	-5.39
LSTM	12.91	-4.92
LSTM-2	17.36	-12.08
Auto-LSTM	20.55	-12.53

Dual-Stage Attention-Based RNN модель

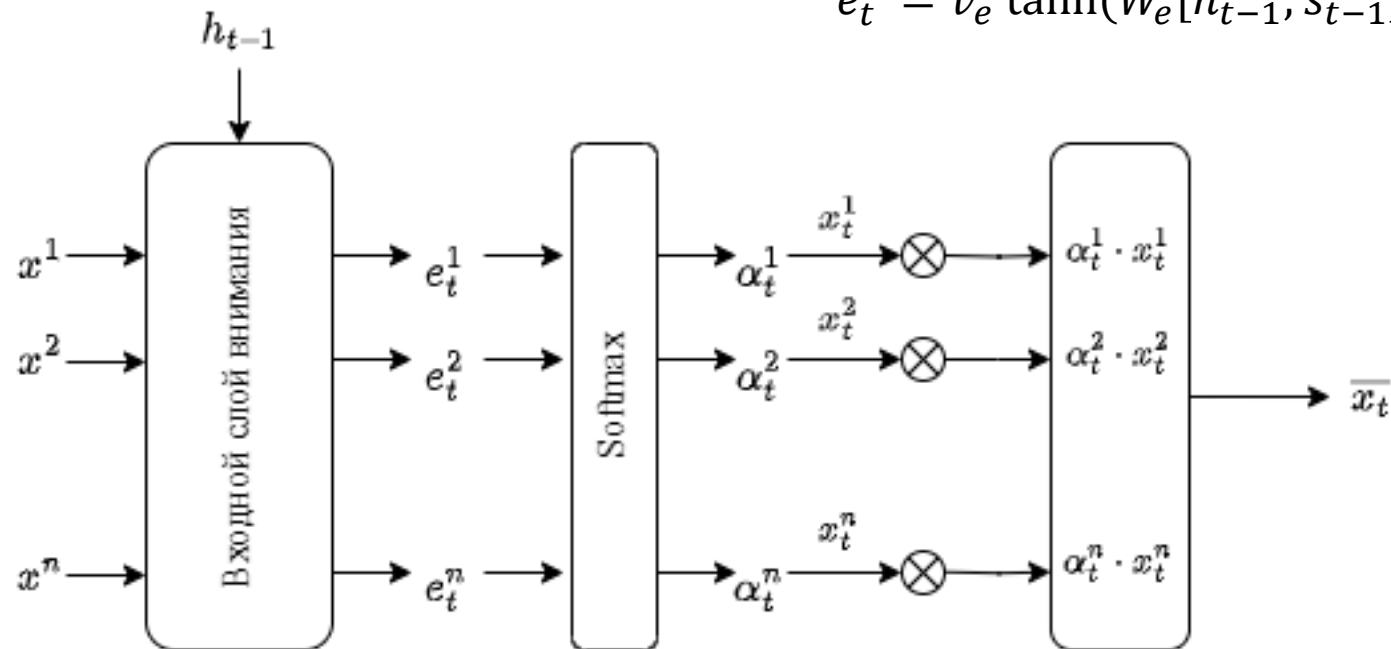


Qin Y. et al. A dual-stage attention-based recurrent neural network for time series prediction //arXiv preprint arXiv:1704.02971. – 2017.

Входной механизм внимания

Рассмотрим работу механизма на примере x^k окна. На вход механизм получает скрытое состояние ячейки $h_{t-1} \in \mathbb{R}^m$ и состояние ячейки $s_{t-1} \in \mathbb{R}^m$ с предыдущего шага:

$$e_t^k = v_e^\top \tanh(W_e[h_{t-1}; s_{t-1}] + U_e x^k)$$



Где $v_e \in \mathbb{R}^T$, $W_e \in \mathbb{R}^{T \times 2m}$, $U_e \in \mathbb{R}^{T \times T}$ – параметры для обучения, m – количество ячеек в кодировщике

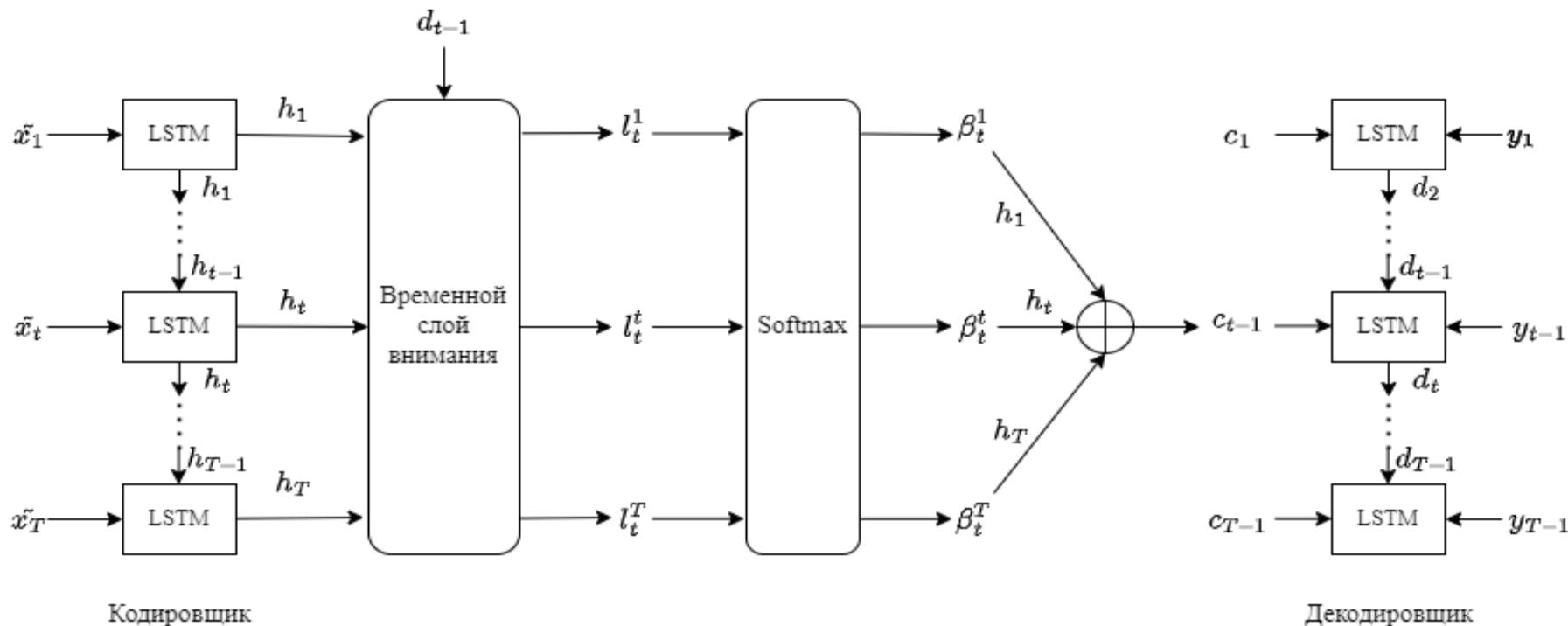
Затем вычисляются весовые коэффициенты:

$$\alpha_t^k = \frac{\exp(e_t^k)}{\sum_{i=1}^n \exp(e_t^i)}$$

Получаем вектор взвешенных наблюдений:

$$\tilde{x}_t = (\alpha_t^1 x_t^1, \alpha_t^2 x_t^2, \dots, \alpha_t^n x_t^n)^\top$$

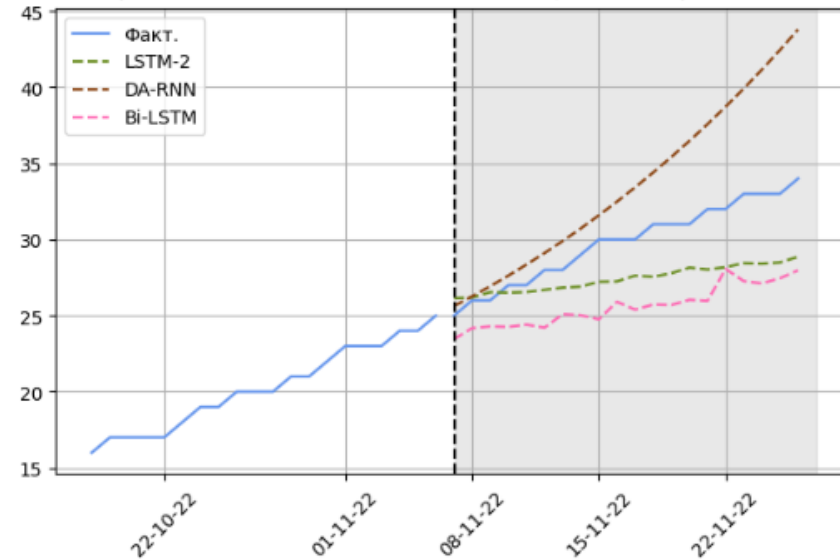
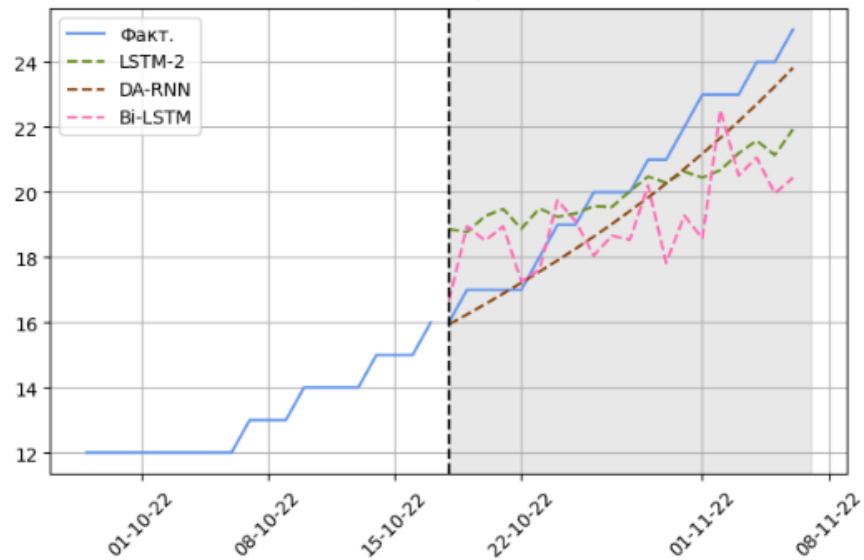
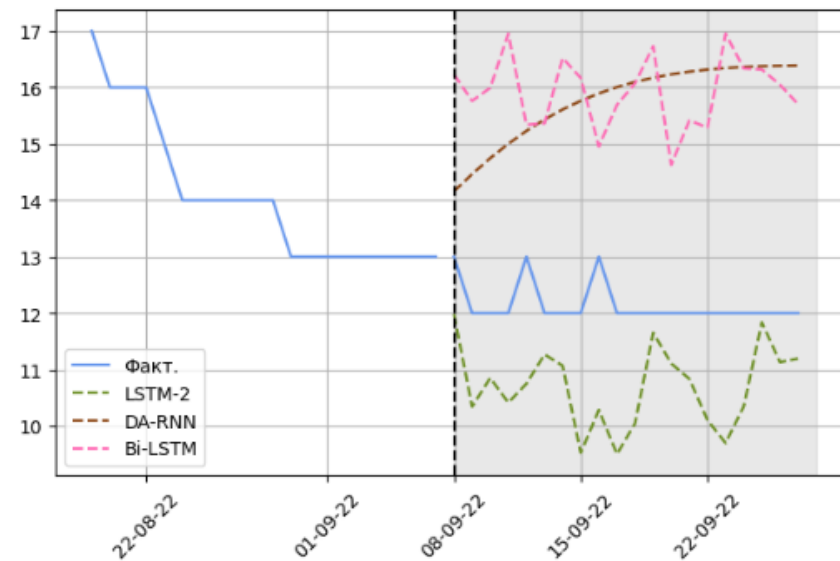
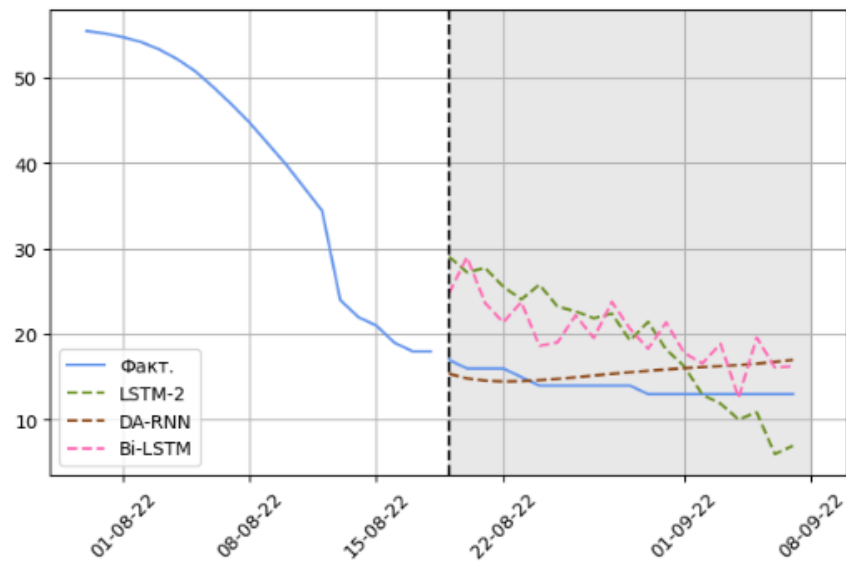
Временной механизм внимания



Логика такая же как и во входном механизме. Разница заключается в следующем: после того, как получены весовые коэффициенты β_t^i , вычисляется взвешенная сумма скрытых состояний кодировщика $c_t = \sum_{i=1}^T \beta_t^i h_i$ и вместе с целевым значением отправляется в полносвязный слой: $\tilde{y}_{t-1} = \tilde{w}^\top [y_{t-1}; c_{t-1}] + \tilde{b}$ \tilde{y}_{t-1} используется для обновления скрытого состояния декодировщика.

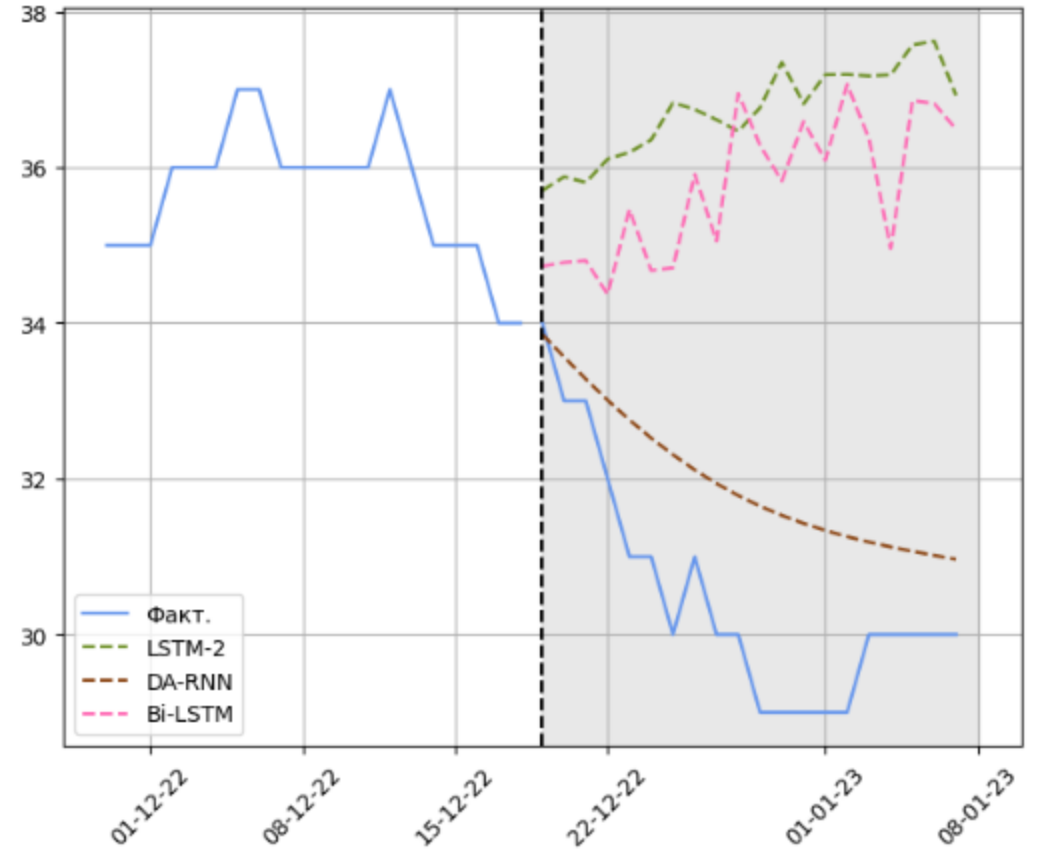
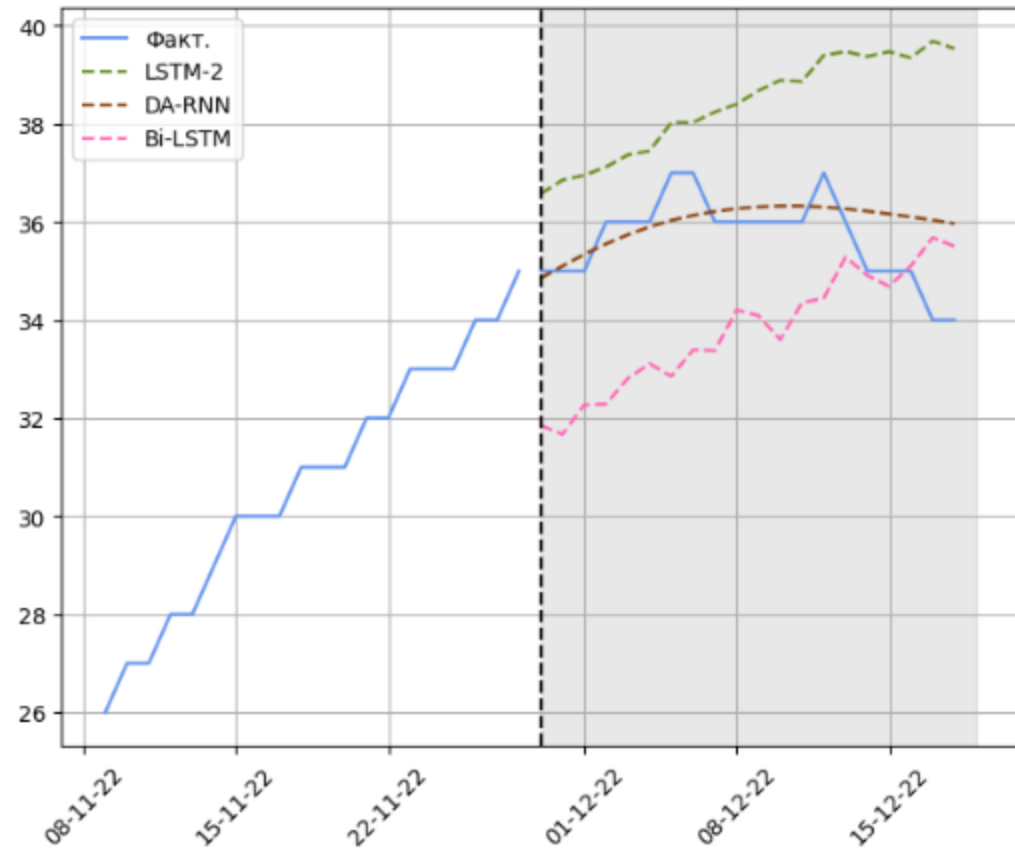
Предсказание DA RNN на валидационном наборе

Предсказание ХДБ с помощью RNN моделей на валидационном наборе



Предсказание DA RNN на тестовом наборе

Предсказание ХДБ с помощью RNN моделей на тестовом наборе



Метрики качества прогноза

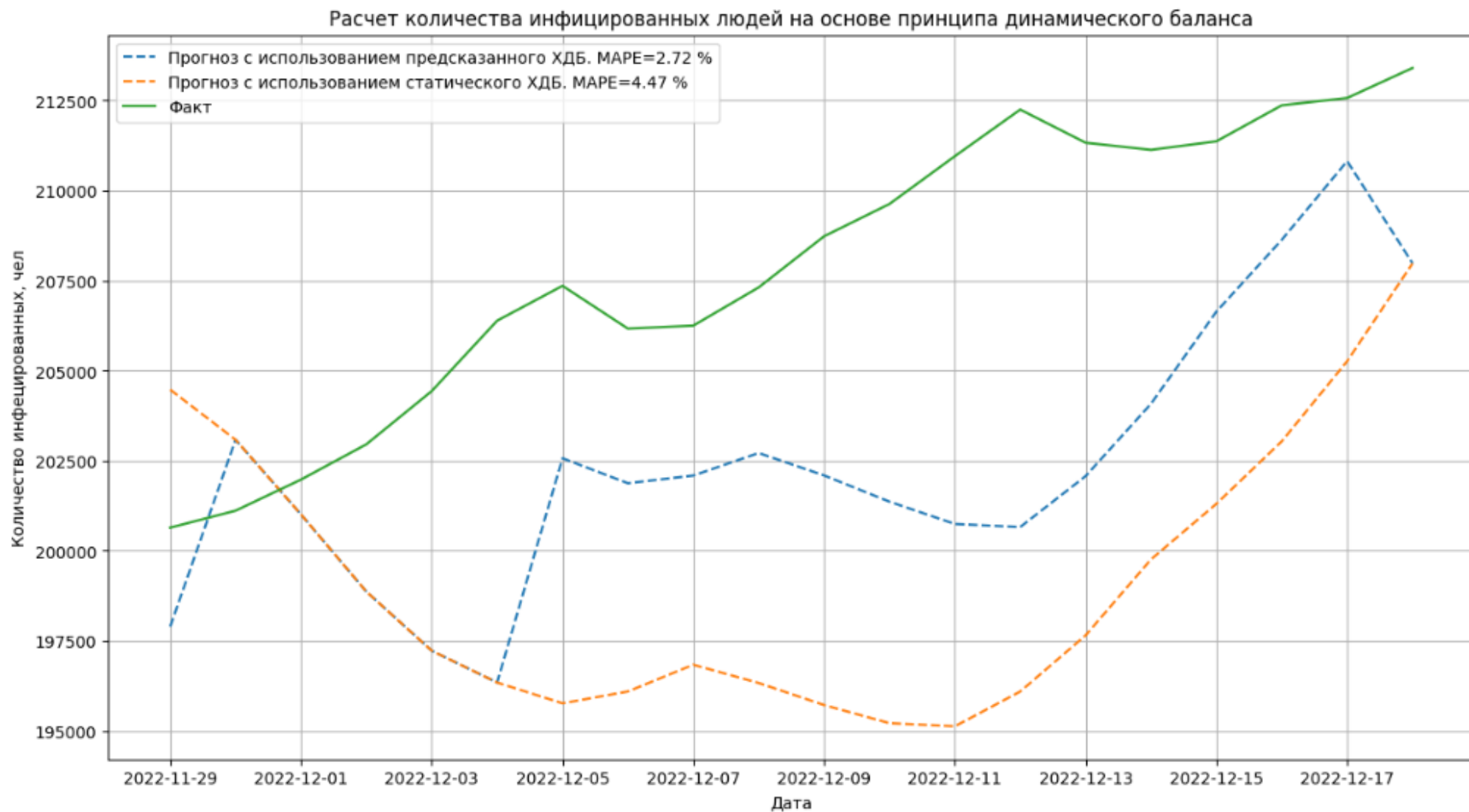
На валидационном наборе

Модель	MAPE, %	R^2
LSTM-2	13.89	-2.07
DA-RNN	14.66	-1.95
Auto-LSTM	15.79	-2.39
Bi-LSTM	17.46	-1.87
LSTM	18.25	-2.93
CNN-LSTM	23.04	-2.58

На тестовом наборе

Модель	MAPE, %	R^2
DA-RNN	6.23	-0.77
Bi-LSTM	12.03	-5.20
CNN-LSTM	12.63	-5.39
LSTM	12.91	-4.92
LSTM-2	17.36	-12.08
Auto-LSTM	20.55	-12.53

Оценка количества инфицированных людей, основанная на модели CIR



Заключение

Цели	Результат
Провести анализ существующих методов прогнозирования временных рядов.	Из классических моделей лучшей оказалась линейная модель (MAPE=18.03%). Используя рекуррентную нейронную сеть Bi-LSTM, удалось уменьшить ошибку (MAPE=12.03%)
Предложить алгоритм машинного обучения для нахождения зависимости между значениями параметров распространения и характеристики динамического баланса(ХДБ).	Реализованная модель DA-RNN снизила ошибку предсказания в 2 раза (MAPE=6.23%)
На основе сделанного прогноза оценить количество болеющих людей, используя принцип динамического баланса	Построенный прогноз ХДБ позволил уменьшить ошибку на 2% при оценке количества инфицированных людей в модели CIR (MAPE=2.72% с использованием предсказанного значения, MAPE=4.47% при использовании константного значения)