



«Санкт-Петербургский политехнический университет  
Петра Великого»

Институт компьютерных наук и технологий  
**Высшая школа киберфизических систем и  
управления**

**Прогнозирование технологических параметров  
автоматизированного теплового пункта методами  
машинного обучения**

Направление 27.03.04 – Управление в технических системах  
Профиль 27.03.04 \_05 – Интеллектуальные системы  
обработки информации и управления

Выполнил

студент гр. 3532704/80501  
Федотовских Евгений Александрович

Научный руководитель

старший преподаватель  
Киселева Людмила Анатольевна



# Цель и задачи

## Цель работы:

создание программного модуля на основе моделей машинного обучения для возможности предиктивного управления интенсивностью отопления на автоматизированном тепловом пункте

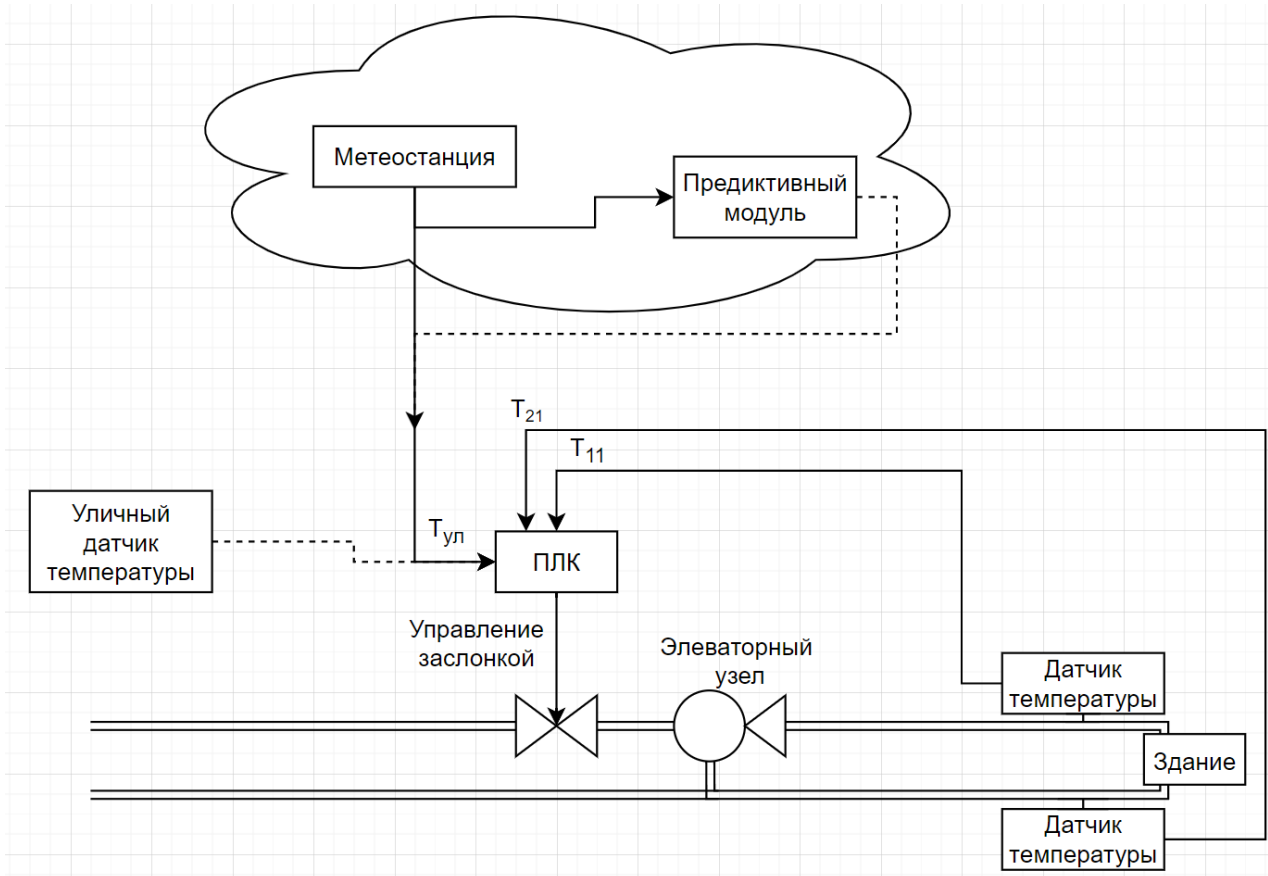
## Задачи:

- 1) Изучить работу автоматизированного теплового пункта;
- 2) Изучить методы машинного обучения для прогнозирования временных рядов;
- 3) Проанализировать предоставленные данные для обучения, предобработать их;
- 4) Реализовать программные модули для прогнозирования одномерных временных рядов, используя изученные методы машинного обучения;
- 5) Сравнить качество получившихся моделей машинного обучения для прогнозирования одномерных временных рядов, оптимизировать модели, выбрать оптимальную;
- 6) Реализовать программный модуль прогнозирования технологических параметров.

## Рассмотренные методы:

1. Случайный лес
2. Градиентный бустинг над решающими деревьями
3. Классические искусственные нейронные сети
4. Искусственные рекуррентные нейронные сети LSTM

# Функциональная схема автоматизированного теплового пункта



**Обозначения:**  $T_{у\text{л}}$  – температура наружного воздуха;  $T_{11}$  – температура носителя к потребителю;  $T_{21}$  – температура носителя от потребителя;

# Описание и предобработка измеряемых данных



## Параметры создания датасетов:

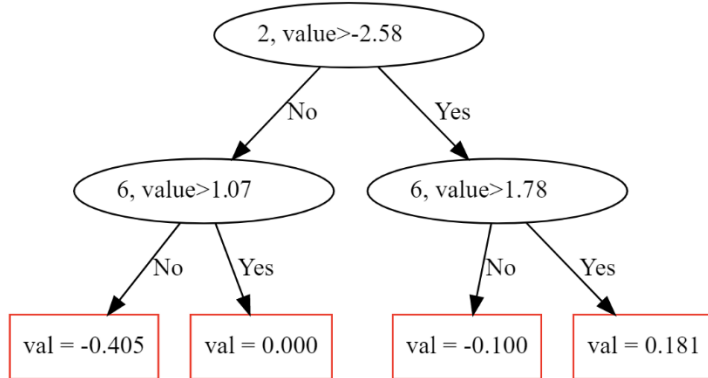
windows\_size – размер окна

forward\_number – кол-во элементов на которое предсказываем

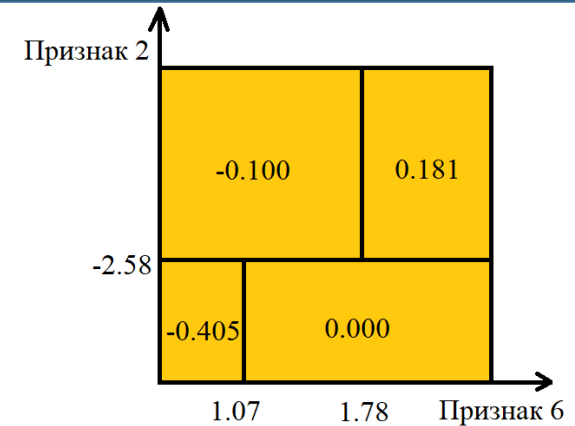


Разделение на датасеты для обучения, проверки качество модели, тестирования происходит в процентном отношении 70-20-10

# Случайный лес



Решающее дерево



Разделение пространства признаков предикатами решающего дерева

Случайный лес – ансамбль из решающих деревьев

$$a(X) = \frac{1}{k} (b_1(X_1) + b_2(X_2) + \dots + b_k(X_k))$$

Фрагмент листинга:

```

regr = RandomForestRegressor(n_estimators=1000, max_depth=10, min_samples_split=2, min_samples_leaf=1, verbose=1)

regr.fit(x_train, y_train)

prediction = regr.predict(x_test)
  
```

# Градиентный бустинг над решающими деревьями

$$L(y, x) = \frac{1}{2} \sum_{i=1}^N (y_i - a(x_i))^2 \rightarrow \min$$

$L(y, x)$  – значение функция потерь

$y_i$  – реальное значение  $i$ -го объекта

$a(x_i)$  – предсказание текущего ансамбля для  $i$ -го объекта

Формирование ансамбля:  $a_{K+1}(x) = b_1(x) + \eta b_2(x) + \dots + \eta b_{k+1}(x)$

$b_i(x)$  – предсказание  $i$ -го решающего дерева

$a_{K+1}(x)$  – предсказание ансамбля из  $K+1$  дерева

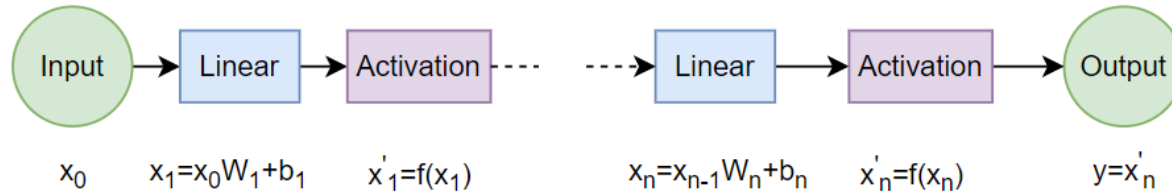
$\eta \in (0; 1]$  – темп обучения

Фрагмент листинга:

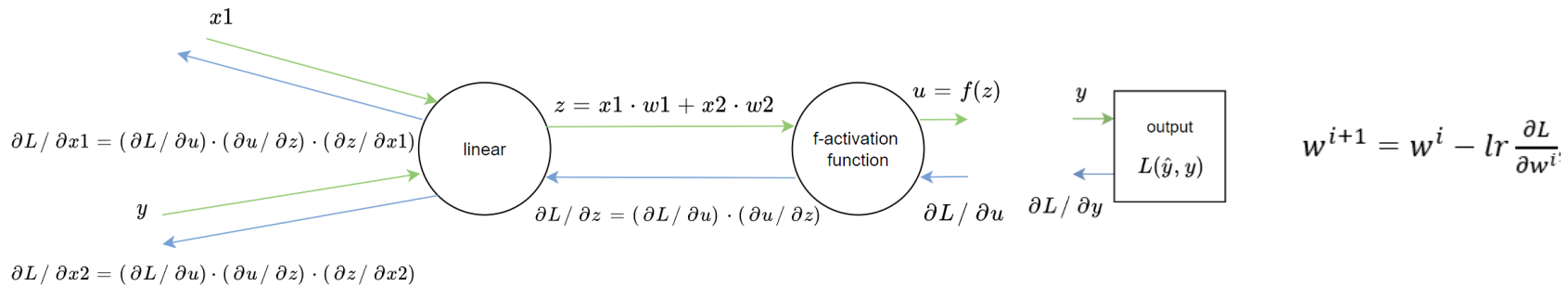
```
model = XGBRegressor(n_estimators=1000, max_depth=7, learning_rate=0.01)

model.fit(x_train, y_train,
          eval_set=[(x_train, y_train), (x_val, y_val)],
          eval_metric='mae',
          verbose=True, callbacks=[xgboost.callback.EarlyStopping(
              rounds=15, save_best=True)])
model.best_ntree_limit
results = model.evals_result()
```

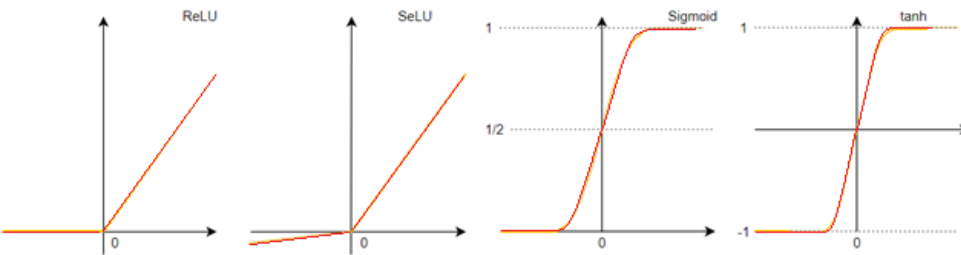
# Искусственные глубокие нейронные сети



Структура простейшей искусственной нейронной сети



Обратное распространение ошибки и обновление весов



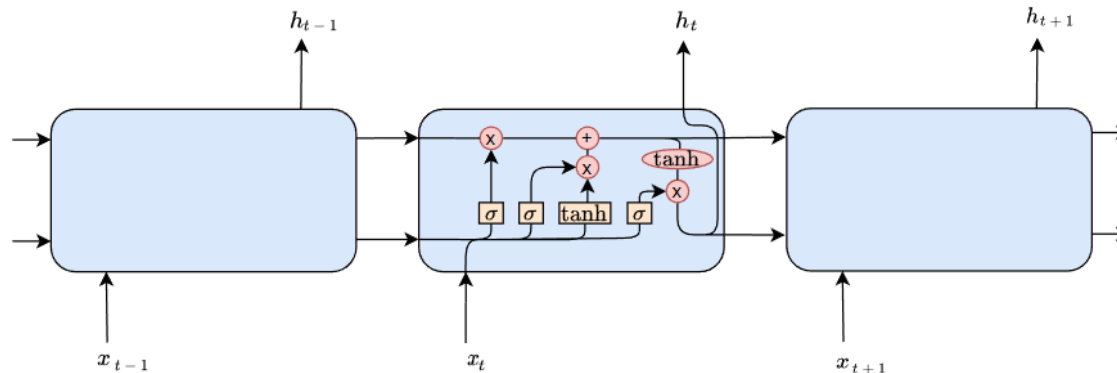
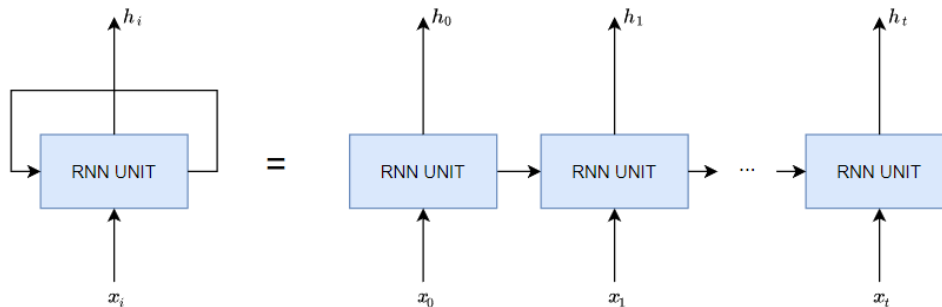
Функции активации

```
model = Sequential([
    Dense(128, activation='selu', input_shape=(window_size,)),
    Dropout(0.5),
    Dense(256, activation='selu'),
    Dropout(0.5),
    Dense(256, activation='selu'),
    Dropout(0.5),
    Dense(256, activation='selu'),
    Dropout(0.5),
    Dense(1)
])
```

Фрагмент листинга

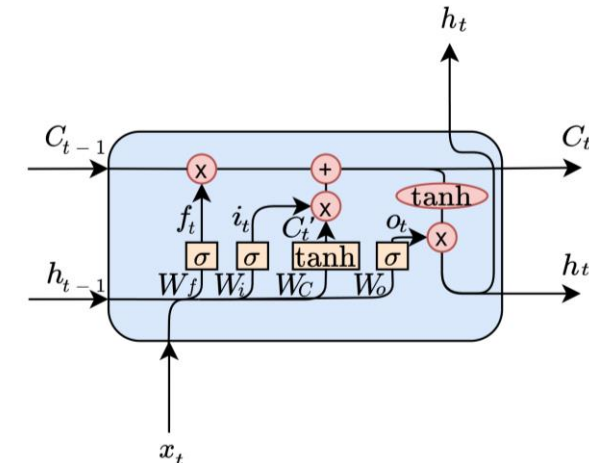
# Искусственные рекуррентные нейронные сети LSTM

Принцип работы рекуррентных нейронных сетей:



Формулы, описывающие работу блока LSTM:

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t]), \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t]), \\ C'_t &= \tanh(W_c \cdot [h_{t-1}, x_t]), \\ C_t &= f_t * C_{t-1} + i_t * C'_t, \\ O_t &= \sigma(W_o \cdot [h_{t-1}, x_t]), \\ h_t &= O_t * \tanh(C_t). \end{aligned}$$



Фрагмент листинга:

```
lstm_model = tf.keras.models.Sequential([
    tf.keras.Input((window_size, 1)),
    tf.keras.layers.LSTM(32, return_sequences=True),
    tf.keras.layers.LSTM(32, return_sequences=False),
    tf.keras.layers.Dense(1)
])
```



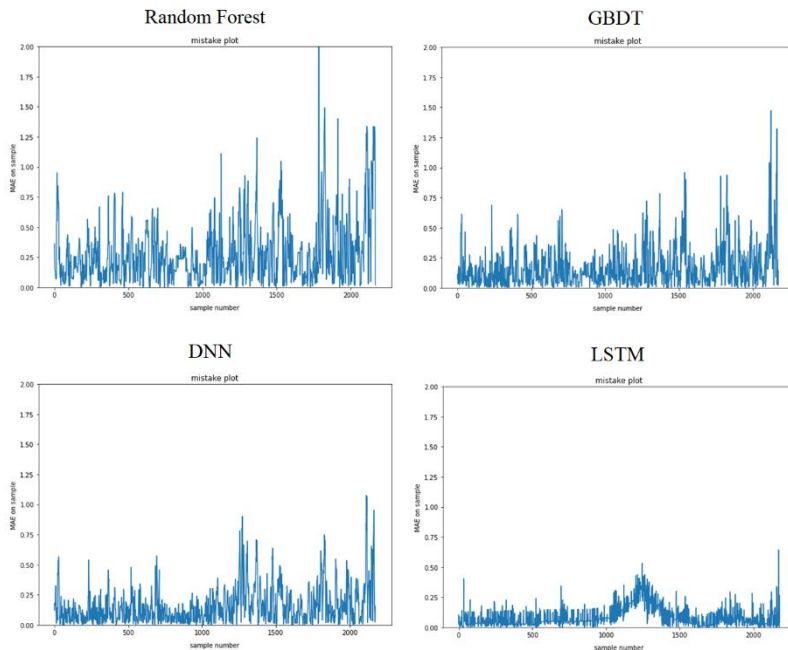
В качестве метрики была взята абсолютная ошибка:

$$MAE(y^{true}, y^{pred}) = \frac{1}{N} \sum_{i=1}^N |y_i - f(x_i)|$$

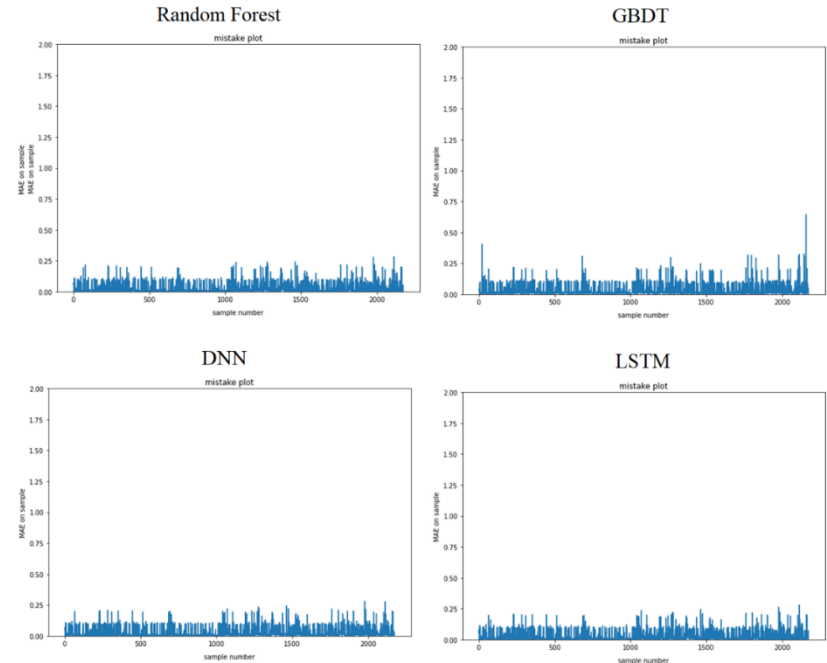
$y^{pred}$  – предсказание модели

$y^{true}$  – реальное значение

Графики MAE для разных моделей с использованием данных за 1 час и предсказанием на 28 минут вперед:

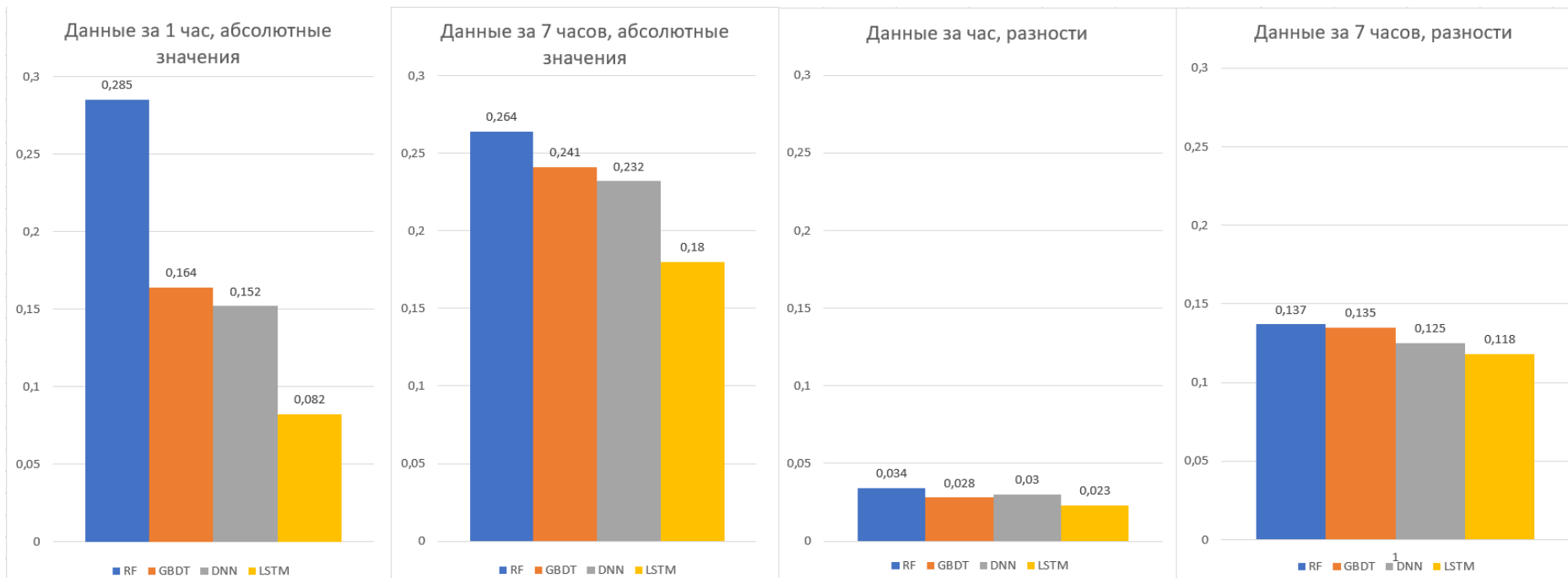


Абсолютные значения



Разности

Средняя абсолютная ошибка для разных методов формирования датасета и разных моделей:



В рамках ВКР были рассмотрены четыре метода прогнозирования временных рядов:

1. случайный лес
2. градиентный бустинг над решающими деревьями
3. искусственные глубокие нейронные сети
4. искусственные рекуррентные нейронные сети LSTM

Было проведено несколько экспериментов, модели разных методов обучались на:

- а) абсолютных значениях температуры
- б) разнице между текущим и прошлым значением;

Датасеты для обучения составлялись с использованием разного количества предыдущих измерений.

В качестве лучшей модели выбрана искусственная нейронная сеть LSTM.

Разработка всех моделей производилась на языке Python 3 в среде Jupiter Notebook.



Спасибо за внимание.