МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего

образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Отчет по заданию №2

по дисциплине

«ГЛУБИННОЕ ОБУЧЕНИЕ»

**«Задание №2 – Рекуррентные нейронные сети для обработки последовательностей»**

Выполнил

Студент гр. 932226

\_\_\_\_\_\_\_\_\_\_\_Савенкова М. М. «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г

Проверил

\_\_\_\_\_\_\_\_ к.т.н. Аксёнов С.В.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Томск – 2023

Оглавление

[1 Индивидуальное задание 3](#_Toc151554117)

[2 Цель работы 4](#_Toc151554118)

[3 Листинг программ 5](#_Toc151554119)

[4 Входные данные 6](#_Toc151554120)

[5 Предобработка данных 7](#_Toc151554121)

[6 Построение моделей 9](#_Toc151554122)

[6.1 Регрессор с RNN в качестве скрытого слоя 9](#_Toc151554123)

[6.1.1 Построенные модели 9](#_Toc151554124)

[6.2 Регрессор с ячейкой LSTM в скрытом слое 17](#_Toc151554125)

[6.2.1 Построенные модели 17](#_Toc151554126)

[6.3 Регрессор – стек двух слоёв с LSTM 23](#_Toc151554127)

[6.3.1 Построенные модели 23](#_Toc151554128)

[7 Сравнение результатов с различным количеством предыдущих наблюдений 30](#_Toc151554129)

[8 Выводы 32](#_Toc151554130)

# 1 Индивидуальное задание

Используя язык программирования Python, построить три нейросетевые модели:

a) Регрессор, использующий слой RNN в качестве скрытого слоя

b) Регрессор, ячейку LSTM (или GRU) в скрытом слое

c) Регрессор – стек двух слоёв с LSTM (или GRU)

для прогноза средней температуры в городе Якутск на следующий день по набору параметров T, P0, P, U за предыдущие дни.

Выполнить загрузку и предварительную обработку данных из наборов. Разделить каждую выборку на обучающую, тестовую и валидационную. Произвести обучение набора нейросетевых архитектур, отличающихся разным набором параметров: количество нейронов в слоях (ячеек), процедур оптимизации:

* Подобрать архитектуры нейронных сетей, которые с одной стороны позволяют получить модели с лучшими метриками качества работы, с другой стороны не являются избыточными и не переобученными. Сделать подбор количества предыдущих значений, используемых для прогнозной модели, для более точной оценки целевой переменной. Сделать эксперименты с прореживанием (Dropout).
* Вычислить следующие метрики работы: R2, MSE, MAE для всех исследованных моделей.

Сделать выводы по результатам построения моделей.

# 2 Цель работы

Получить навыки создания нейронных сетей для анализа сигналов с помощью моделей рекуррентных сетей и ячеек LSTM (GRU).

# 3 Листинг программ

Листинги программ представлены в отдельном файле: RNNs\_(Savenkova\_932226).ipynb.

# 4 Входные данные

Входные данные представлены в виде таблицы, столбцам которой соответствуют следующие признаки:

* T – температура в градусах Цельсия на расстоянии 2-х метров от поверхности земли.
* P0 – атмосферное давление в мм. ртутного столба на уровне станции.
* P – атмосферное давление в мм. ртутного столба, приведенное к среднему на уровне моря.
* U – относительная влажность в процентах на уровне 2-х метров от поверхности земли.
* DD – направление ветра на высоте 10-12 метров над поверхностью.

Предсказываемым параметром является температура T.

# 5 Предобработка данных

В качестве предобработки данных были выполнены следующие шаги:

* проверка пропущенных значений;
* кодирование категориальных переменных;
* отбор признаков;
* сериализация данных.

С помощью библиотеки missingno убеждаемся, что в наборе данных есть пропущенные значения (рисунок 1).

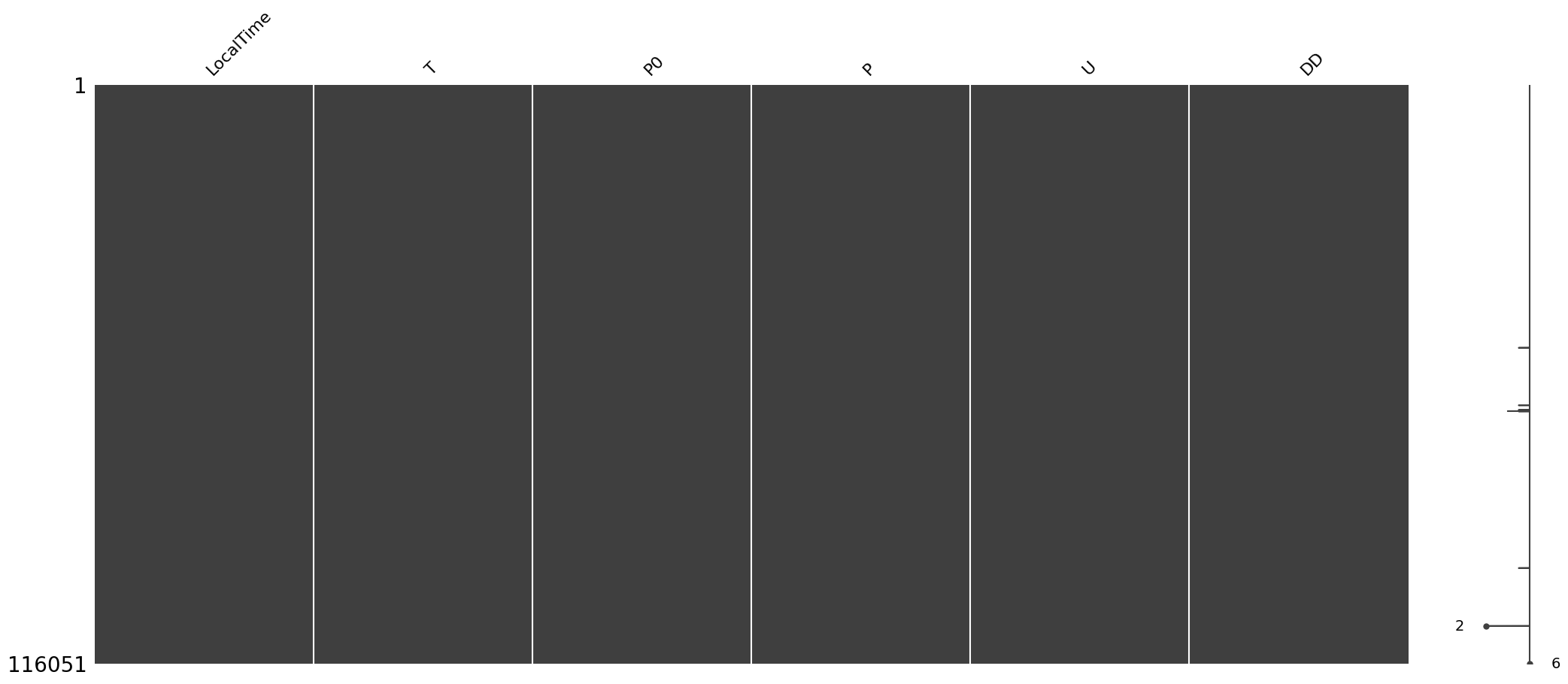


Рисунок 1 — Пропущенные значения

Так как пропусков немного, их можно просто удалить.

После удаления пропущенных значений, категориальные данные были закодированы с помощью LabelEncoder библиотеки scikit-learn, который кодирует переменные метками со значением от 0 до n\_classes-1, где n\_classes – это общее число различных значений признака.

Визуализируем значения каждого признака (рисунок 2):

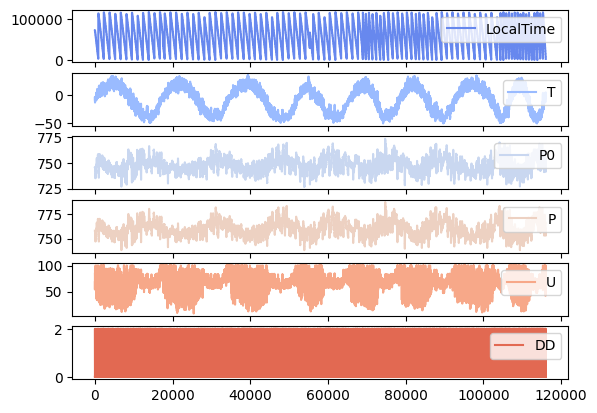


Рисунок 2 — Визуализация признаков

Как мы видим, признак DD (направление ветра) не обладает такой периодичностью, как остальные признаки, поэтому его можно исключить из рассмотрения. LocalTime (дата и время наблюдения) исключается тоже, так как не обладает нужной информативностью.

Таким образом, предсказания температуры будут осуществляться по признакам T, P0, P, U.

Далее, перед разделением данных на выборки для обучения, валидации и тестирования модели, данные были сериализованы, то есть представлены в формате последовательностей размером series\_size, который был выбран равными 20.

# 6 Построение моделей

6.1 Регрессор с RNN в качестве скрытого слоя

6.1.1 Построенные модели

RNN является одним из типов нейронных сетей, которые поддерживают обработку последовательных данных, таких как временные ряды или естественные языки.

Простая рекуррентная нейронная сеть реализована в Keras с помощью класса SimpleRNN.

SimpleRNN имеет один скрытый слой, который рекурсивно передает информацию из предыдущего временного шага в следующий. Это связано с возможностью моделирования долгосрочных зависимостей в данных. SimpleRNN может быть использован для задач классификации, регрессии или генерации последовательностей.

6.1.1.1 Модель №1

**Параметры архитектуры и обучения**

Создаем полносвязную модель рекуррентной нейронной сети с одним скрытым слоем. Выходной слой возвращает одно число – температуру.

Слой RNN содержит 32 нейрона, функция активации – relu. Параметр return\_sequences означает, следует ли возвращать последний вывод в выходной последовательности или полную последовательность, в данном случае он установлен в False, то есть слой RNN возвращает только последний вывод. Размеры входного вектора (input\_shape) – (количество наблюдений в серии, количество признаков).

Слой Dense содержит 1 нейрон, функция активации – линейная.

**Параметры обучения**

В качестве функции потерь используем MSE, используемые метрики – MAE, MSE, R2. Оптимизатор – adam.

Модель обучена на 20 эпохах с размером батча равным 32.

**Полученные метрики**

Метрики для модели показаны на рисунке 3.

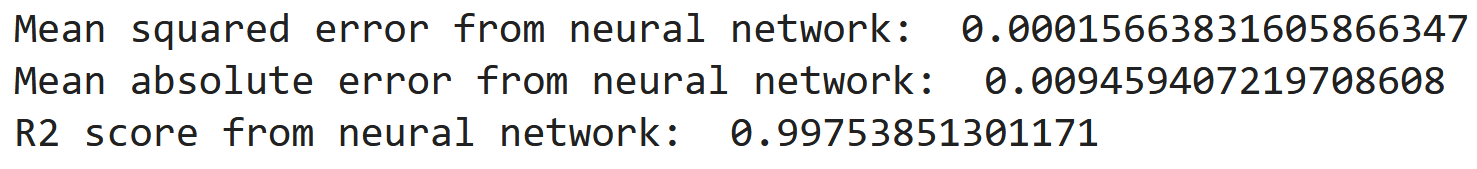


Рисунок 3 — Метрики

**Графики обучения**

График визуализации обучения (рисунок 4).

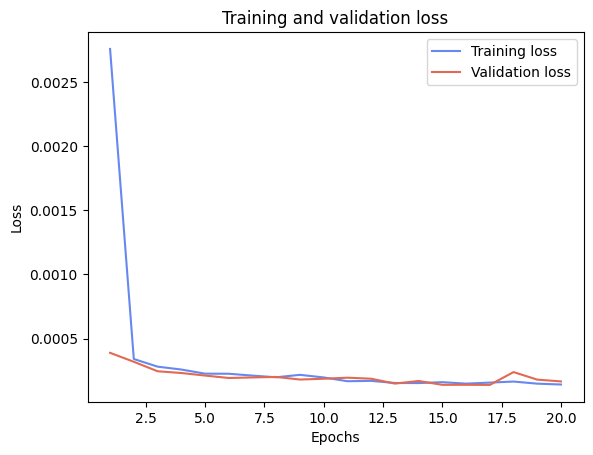


Рисунок 4 — График обучения

График оценки предсказаний (рисунок 5).

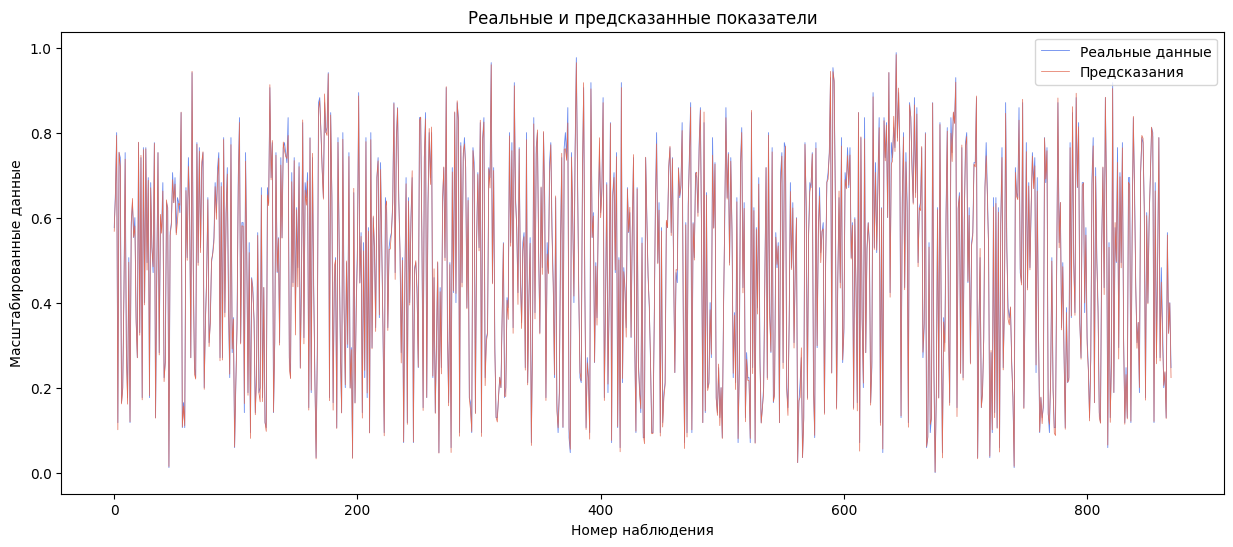


Рисунок 5 — Оценка предсказаний

6.1.1.2 Модель №2

**Параметры архитектуры и обучения**

Попробуем изменить оптимизатор на RMSprop и уменьшить размер батча до 8. Увеличим количество эпох до 30. Другие параметры остаются прежними.

**Полученные метрики**

Метрики для модели показаны на рисунке 6.

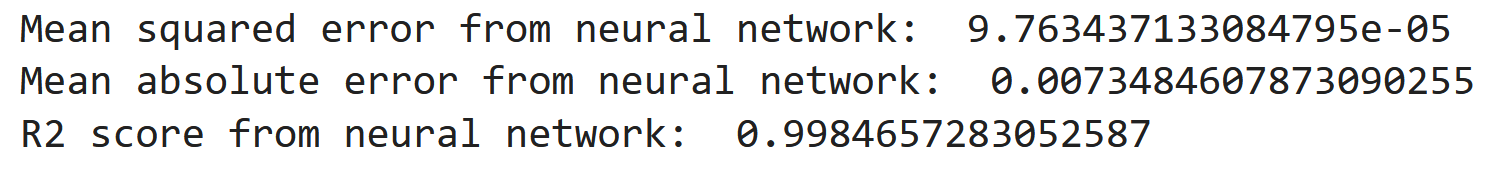


Рисунок 6 — Метрики

**Графики обучения**

График визуализации обучения (рисунок 7).

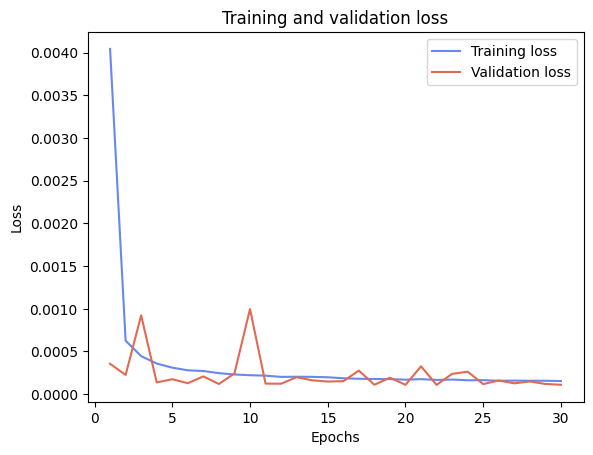


Рисунок 7 — График обучения

График оценки предсказаний (рисунок 8).

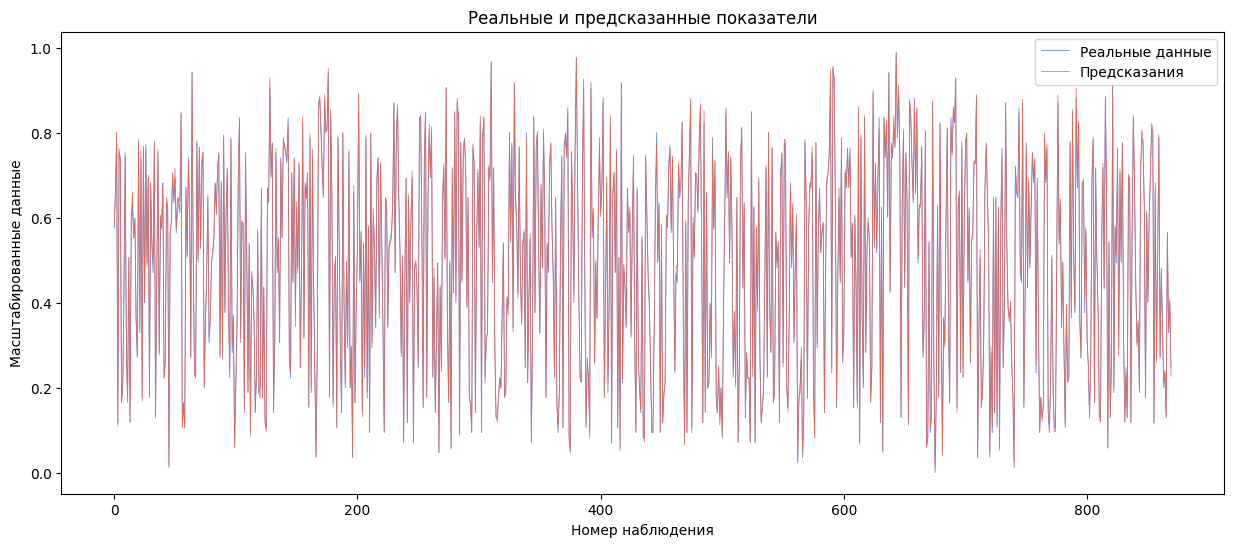


Рисунок 8 — Оценка предсказаний

6.1.1.3 Модель №3

**Параметры архитектуры и обучения**

Оставим в качестве оптимизатора adam, потому что с ним кривая потерь более гладкая. Увеличим количество нейронов в RNN до 64, вернем прежний размер батча – 32 и будем обучать модель в течение 30 эпох.

**Полученные метрики**

Метрики для модели показаны на рисунке

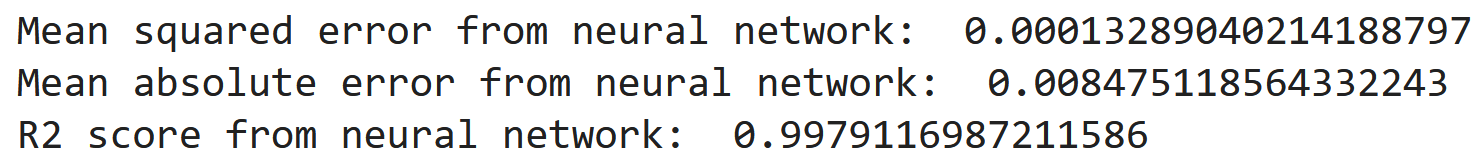


Рисунок 9 — Метрики

**Графики обучения**

График визуализации обучения (рисунок 10).

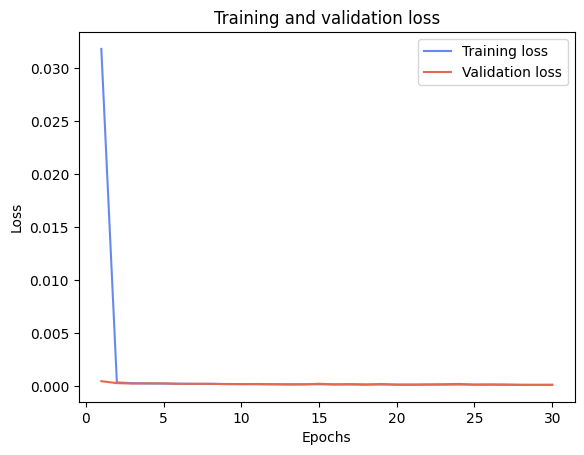


Рисунок 10 — График обучения

График оценки предсказаний (рисунок 11).

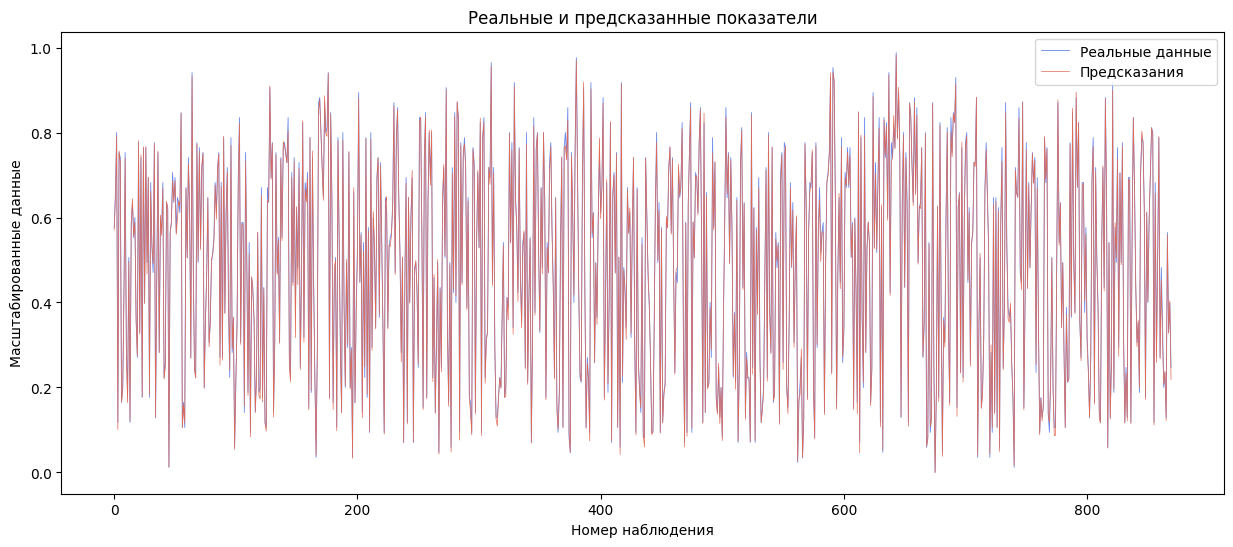


Рисунок 11 — Оценка предсказаний

6.1.1.4 Модель №4

**Параметры архитектуры и обучения**

Проведем эксперименты со слоем Dropout. Добавим в RNN-слой параметр dropout, равный 0.1. Количество эпох установив равным 30 и добавим callback-функцию для ранней остановки, которая остановит обучение, если в течение 5 эпох mae на валидационной выборке не уменьшится.

**Полученные метрики**

Метрики для модели показаны на рисунке 12.

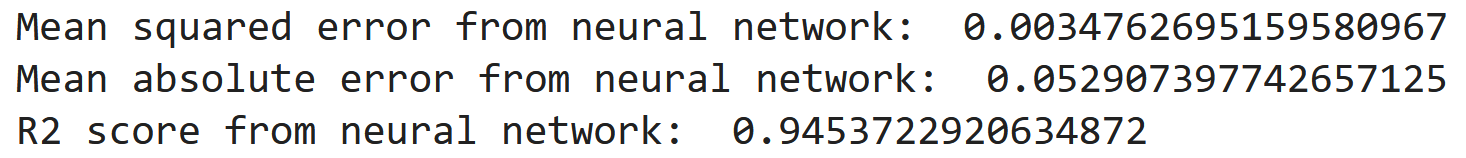


Рисунок 12 — Метрики

**Графики обучения**

График визуализации обучения (рисунок 13).

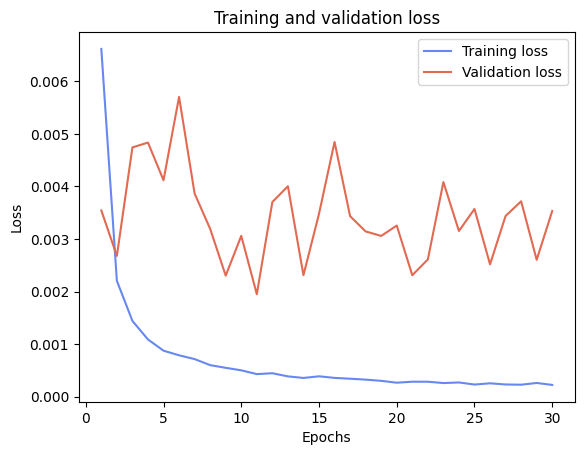


Рисунок 13 — График обучения

График оценки предсказаний (рисунок 14).

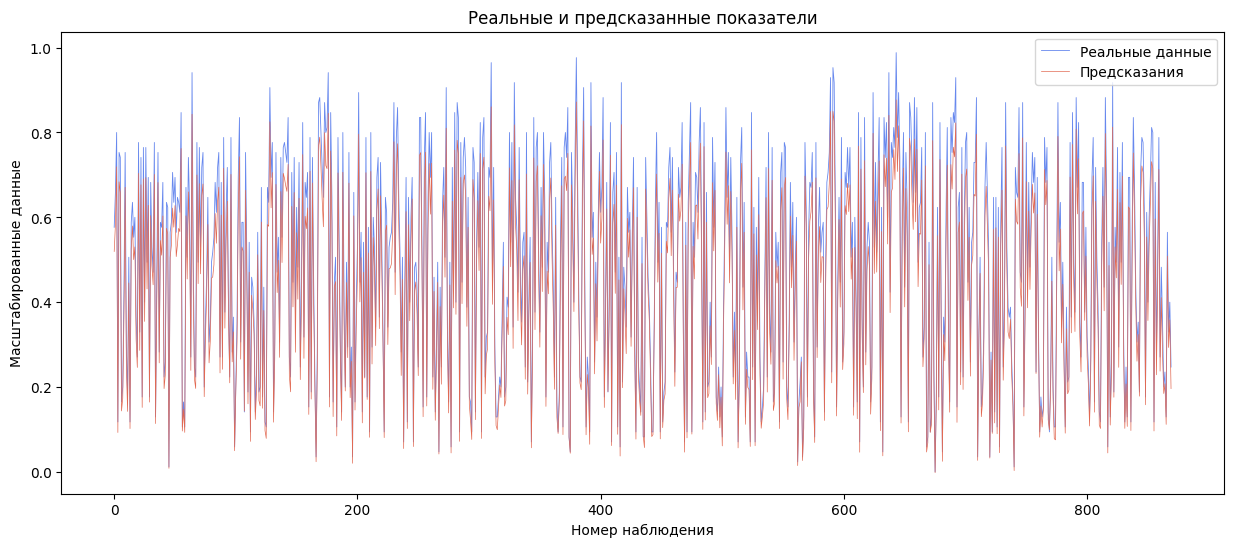


Рисунок 14 — Оценка предсказаний

6.2 Регрессор с ячейкой LSTM в скрытом слое

LSTM (Long Short-Term Memory) в Keras представляет собой тип рекуррентной нейронной сети, который используется для обработки последовательностей данных. LSTM способна запоминать и учитывать долгосрочные зависимости в последовательностях данных, что делает ее особенно полезной для работы с текстами, речью, временными рядами и другими типами последовательных данных.

LSTM является одним из типов рекуррентных нейронных сетей и представляет собой улучшенную версию стандартных простых рекуррентных нейронных сетей (RNN). LSTM включает в себя дополнительные блоки памяти, которые помогают сохранять и использовать информацию на протяжении более длительного времени.

6.2.1 Построенные модели

6.2.1.1 Модель №1

**Параметры архитектуры и обучения**

Слой LSTM содержит 32 нейрона, функция активации – relu. Параметр return\_sequences означает, следует ли возвращать последний вывод в выходной последовательности или полную последовательность, в данном случае он установлен в False, то есть слой LSTM возвращает только последний вывод. Размеры входного вектора (input\_shape) – (количество наблюдений в серии, количество признаков).

Слой Dense содержит 1 нейрон, функция активации – линейная.

В качестве функции потерь используем MSE, используемые метрики – MAE, MSE, R2. Оптимизатор – adam.

Модель будет обучена на 20 эпохах с размером батча равным 32.

**Полученные метрики**

Метрики для модели показаны на рисунке 15.

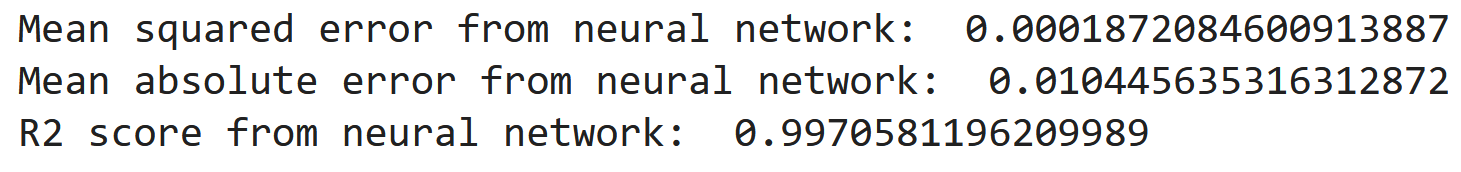


Рисунок 15 — Метрики

**Графики обучения**

График визуализации обучения (рисунок 16).

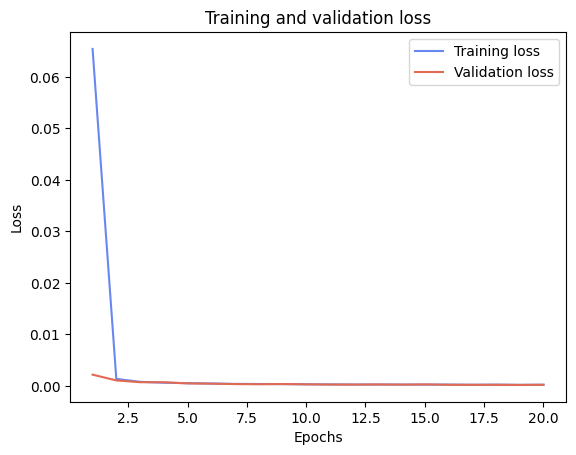


Рисунок 16 — График обучения

График оценки предсказаний (рисунок 17).

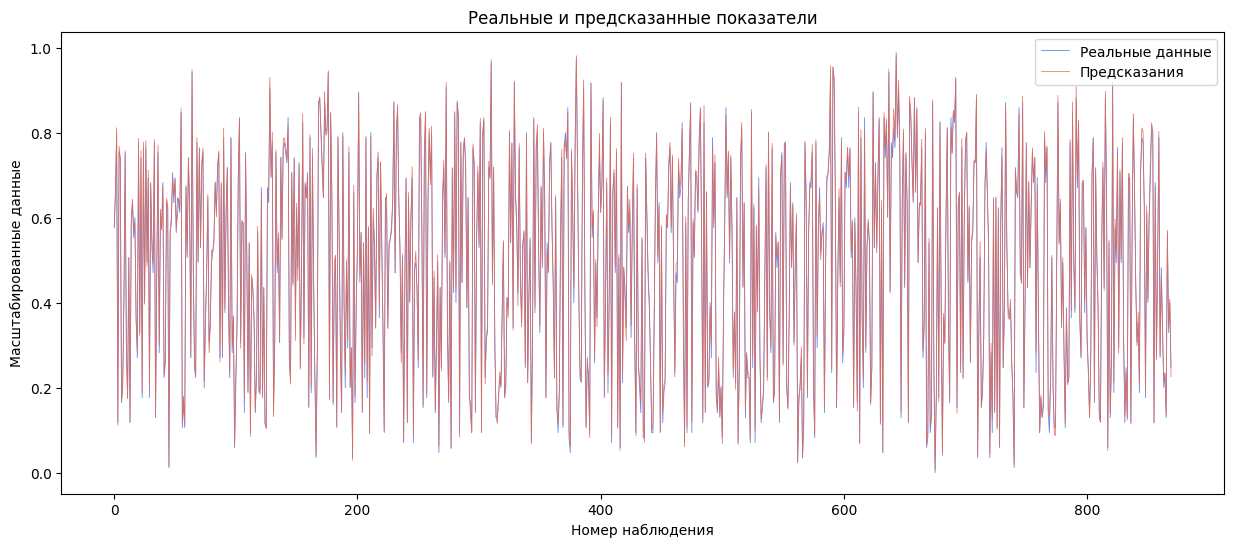


Рисунок 17 — Оценка предсказаний

6.2.1.2 Модель №2

**Параметры архитектуры и обучения**

Увеличим количество эпох до 30 и будем уменьшать коэффициент обучения каждую эпоху, начиная с пятой. Добавим также раннюю остановку.

**Полученные метрики**

Метрики для модели показаны на рисунке

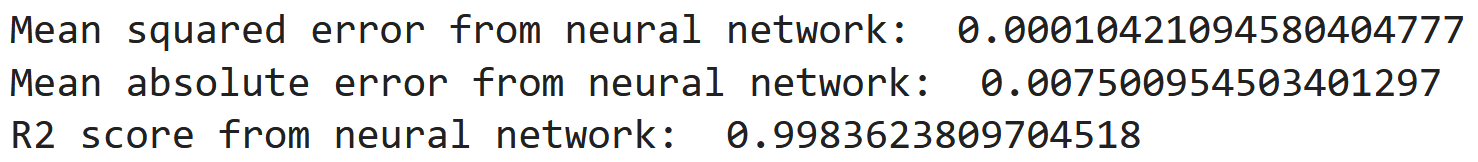


Рисунок 18 — Метрики

**Графики обучения**

График визуализации обучения (рисунок 19).

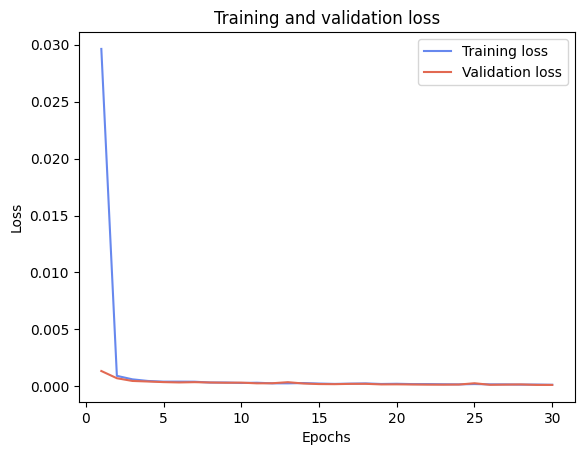


Рисунок 19 — График обучения

График оценки предсказаний (рисунок 20).

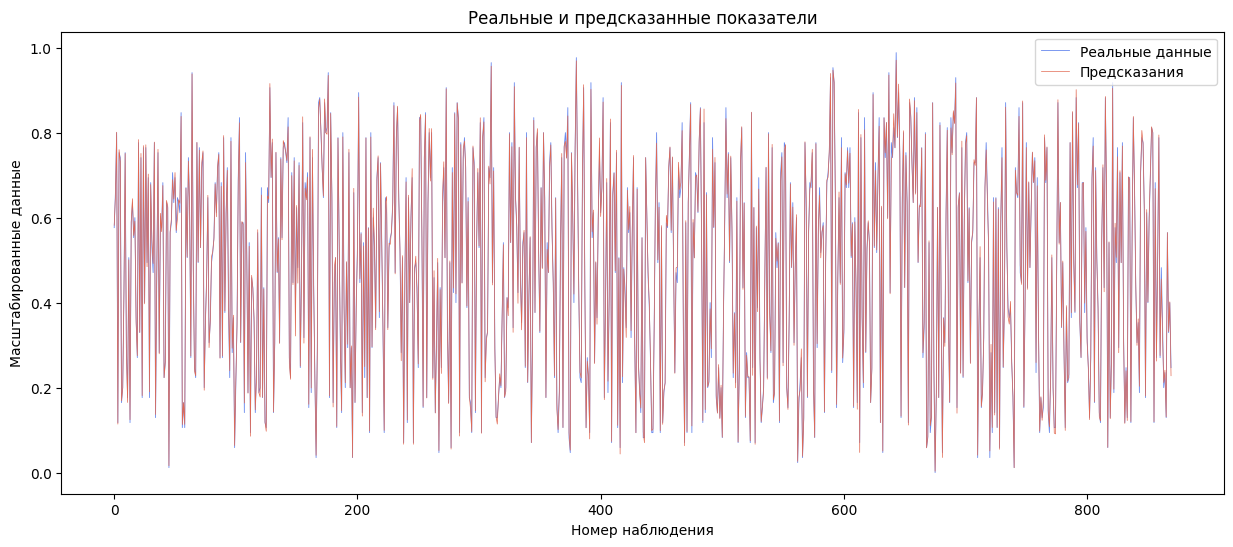


Рисунок 20 — Оценка предсказаний

6.2.1.3 Модель №3

**Параметры архитектуры и обучения**

Оставим callback, уменьшающий скорость обучения на каждой эпохе, но увеличим количество нейронов до 128 в LSTM-слое и добавим dropout с веростностью 0.1. Увеличим количество эпох обучения до 100.

**Полученные метрики**

Метрики для модели показаны на рисунке 21.

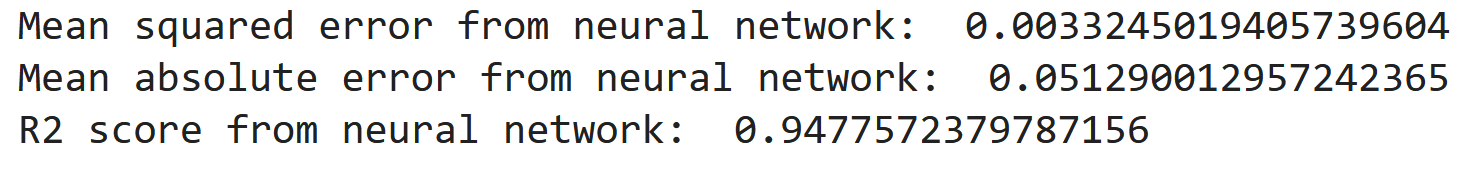


Рисунок 21 — Метрики

**Графики обучения**

График визуализации обучения (рисунок 22).

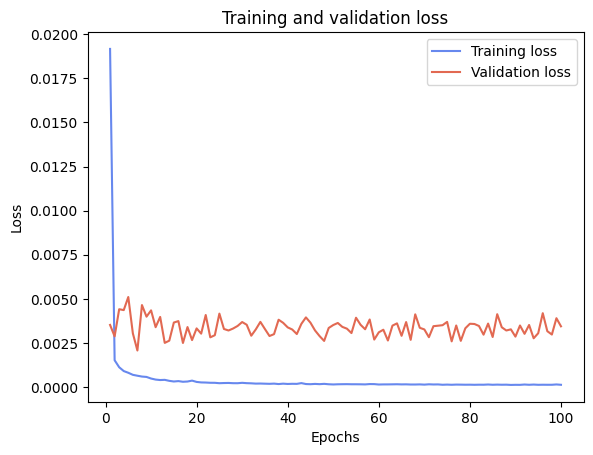


Рисунок 22 — График обучения

График оценки предсказаний (рисунок 23).

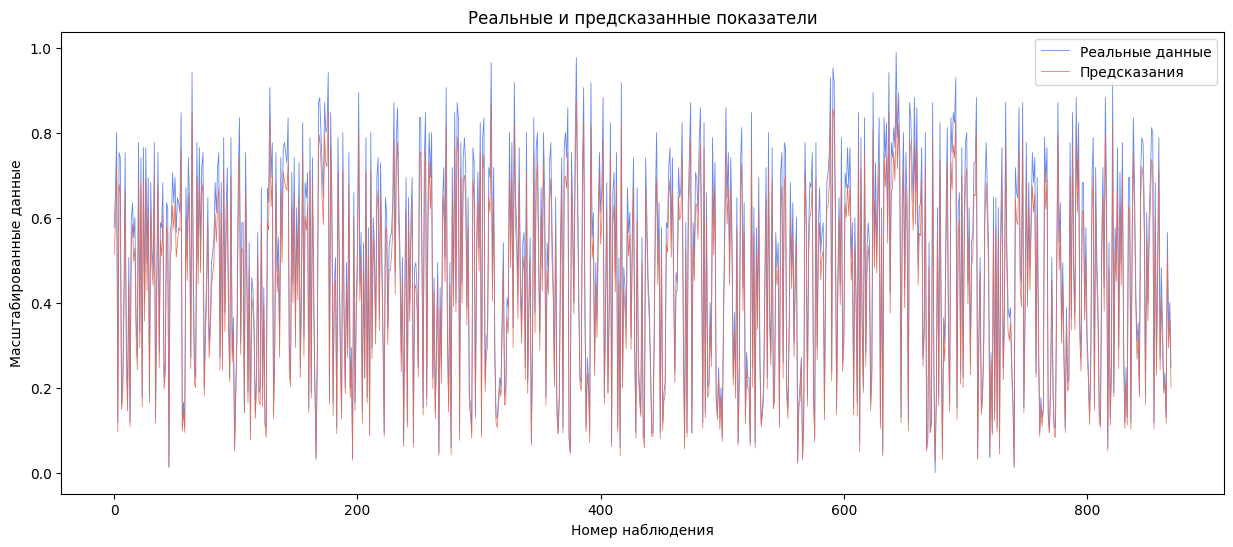


Рисунок 23 — Оценка предсказаний

6.3 Регрессор – стек двух слоёв с LSTM

Stacked LSTM (стековые LSTM) - это модель рекуррентной нейронной сети, которая состоит из нескольких LSTM-слоев, где каждый слой получает на вход выходной сигнал предыдущего слоя.

6.3.1 Построенные модели

6.3.1.1 Модель №1

**Параметры архитектуры и обучения**

Нейронная сеть содержит два LSTM-слоя. Первый слой содержит 64 нейрона, второй LSTM-слой содержит 32 нейрона, функции активации обоих слоев – relu. Размеры входного вектора (input\_shape) – (количество наблюдений в серии, количество признаков).

Параметр return\_sequences первого слоя установлен в True, то есть этот LSTM-слой будет получать на вход последовательности, и передавать последовательности на следующий слой. Во втором слое данный параметр установлен в False.

Слой Dense содержит 1 нейрон, функция активации – линейная.

В качестве функции потерь используем MSE, используемые метрики – MAE, MSE, R2. Оптимизатор – adam.

Модель будет обучена на 20 эпохах с размером батча равным 32.

**Полученные метрики**

Метрики для модели показаны на рисунке 24.

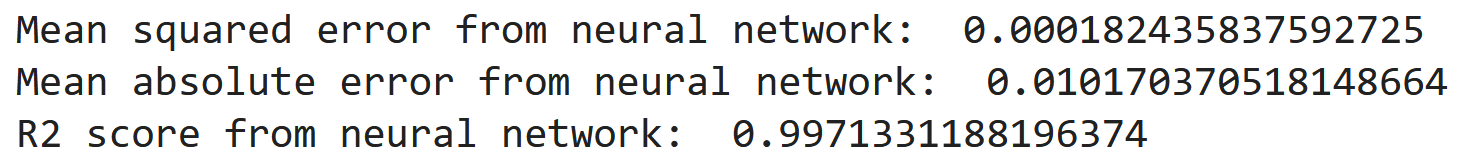


Рисунок 24 — Метрики

**Графики обучения**

График визуализации обучения (рисунок 25).

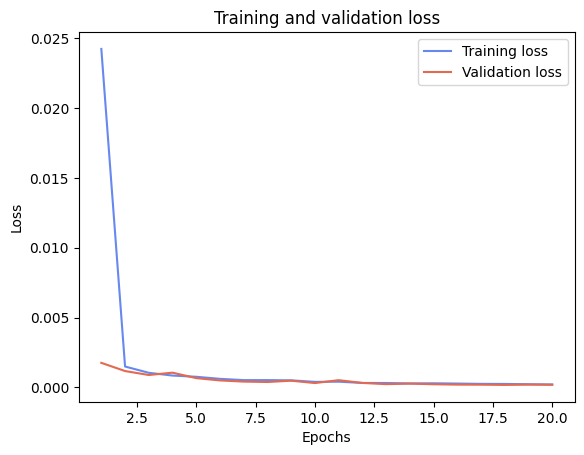


Рисунок 25 — График обучения

График оценки предсказаний (рисунок 26).

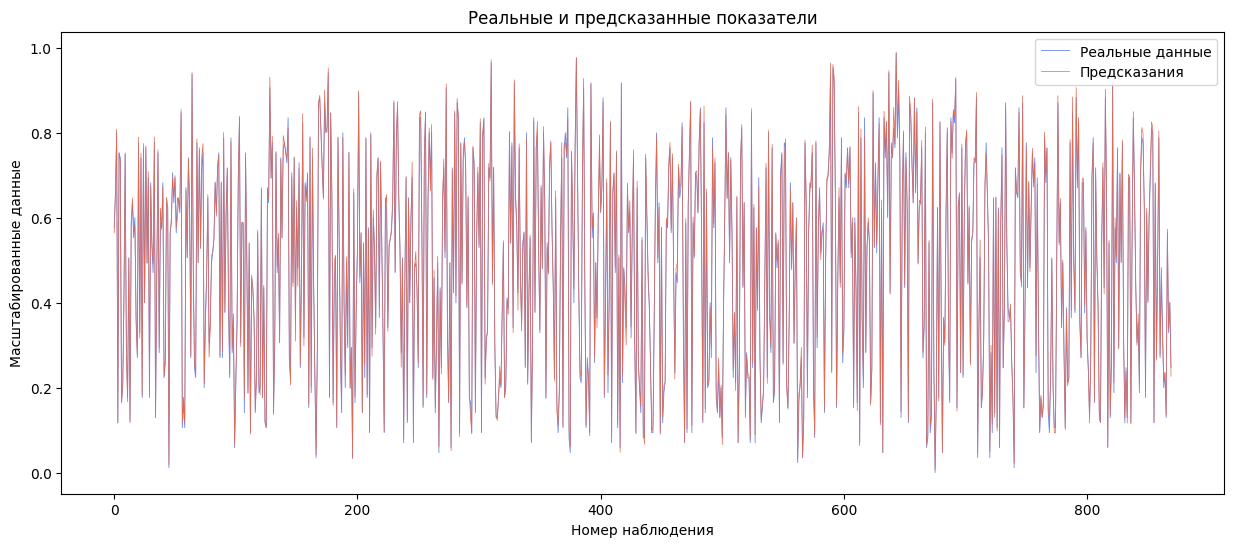


Рисунок 26 — Оценка предсказаний

6.3.1.2 Модель №2

**Параметры архитектуры и обучения**

Функцией активации в LSTM по умолчанию является гиперболический тангенс. Восстановим значения по умолчанию и оценим результат.

**Полученные метрики**

Метрики для модели показаны на рисунке 27.

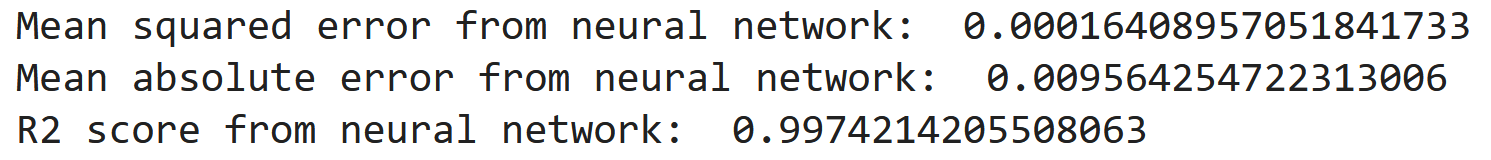


Рисунок 27 — Метрики

**Графики обучения**

График визуализации обучения (рисунок 28).

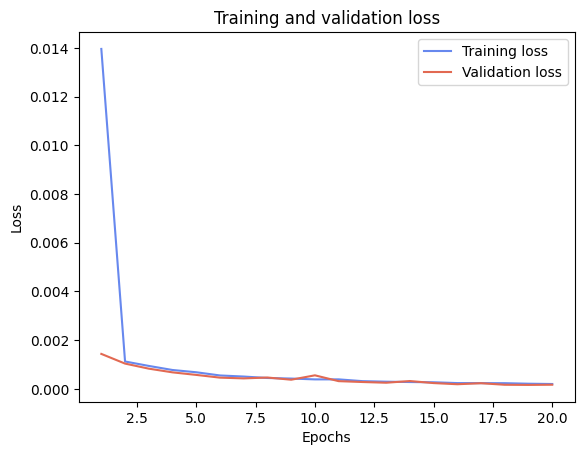


Рисунок 28 — График обучения

График оценки предсказаний (рисунок 29).

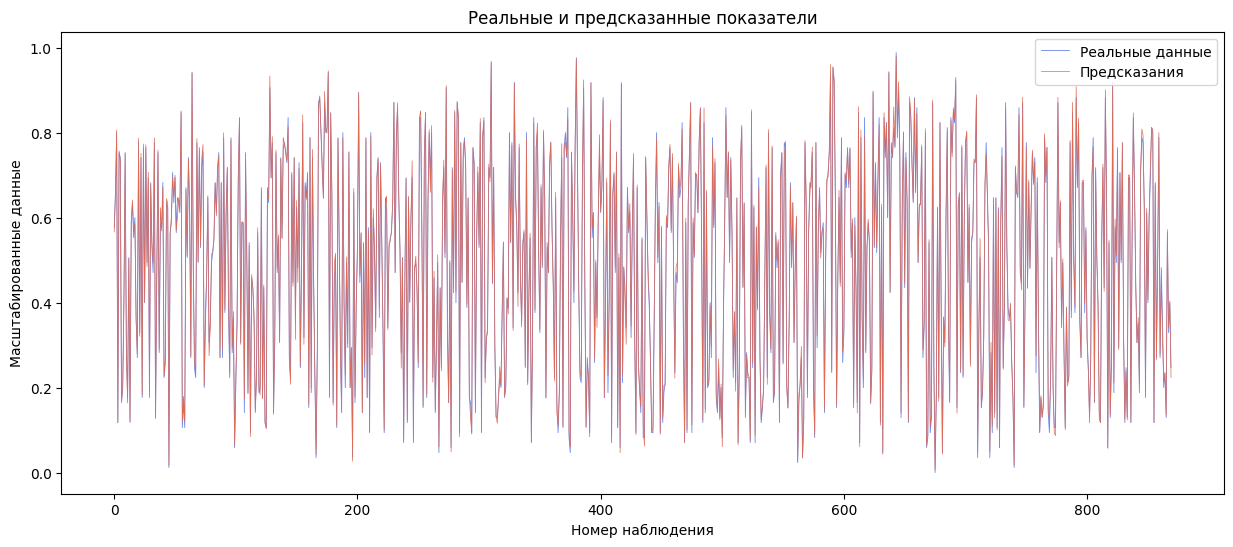


Рисунок 29 — Оценка предсказаний

6.3.1.3 Модель №3

**Параметры архитектуры и обучения**

Снова возьмем в качестве функции активации внутренних LSTM-слоёв relu. Добавим dropout. Установим соответствующий параметр в LSTM-слоях в 0.01 и 0.03.

**Полученные метрики**

Метрики для модели показаны на рисунке 30.

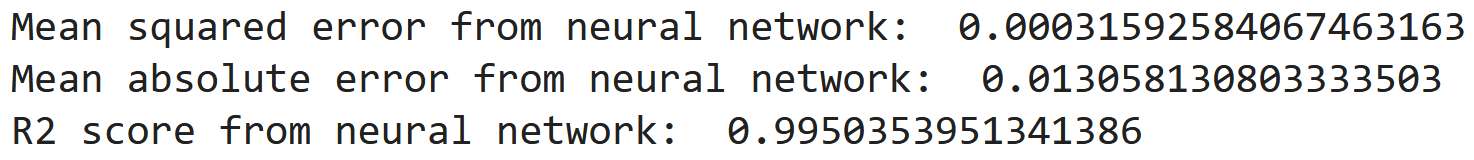


Рисунок 30 — Метрики

**Графики обучения**

График визуализации обучения (рисунок 31).

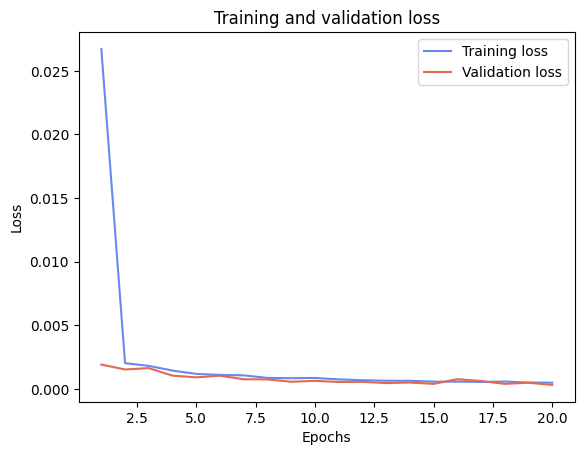


Рисунок 31 — График обучения

График оценки предсказаний (рисунок 32).

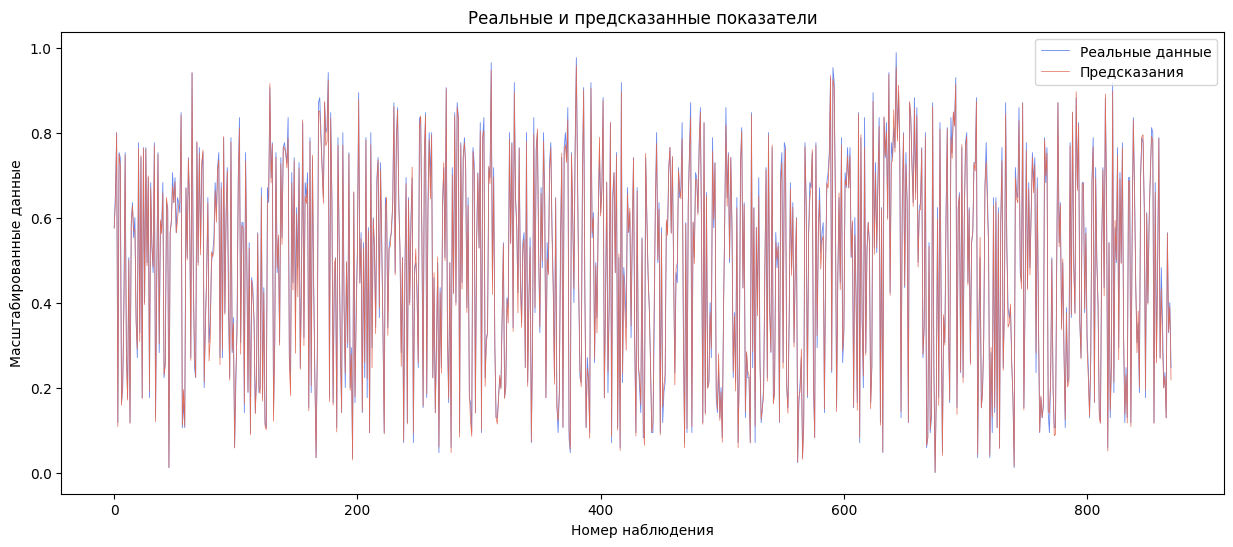


Рисунок 32 — Оценка предсказаний

# 7 Сравнение результатов с различным количеством предыдущих наблюдений

Построим график, отображающий значение коэффициента детерминации для каждой модели, и сравним его вид для различного количества предыдущих значений (параметр series\_size), равного 20, 8 и 72 (рисунки 33, 34 и 35 соответственно).

Изменение значения параметра повлекло изменение значений коэффициента детерминации в тысячных и десятитысячных долях. Так как наблюдения осуществлялись каждый час, это значит, что с большой долей достоверности можно предсказать погоду, обладая данными за трое суток. Значительное увеличение окна наблюдения может сказаться негативно на точности прогноза.

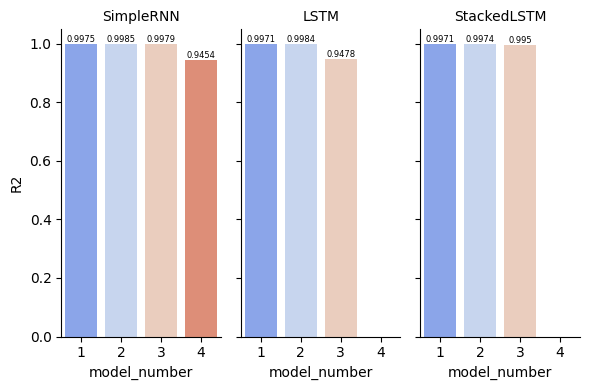


Рисунок 33 — series\_size равен 20

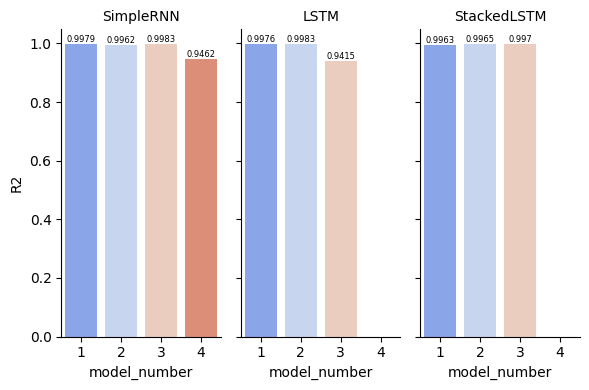


Рисунок 34 — series\_size равен 8

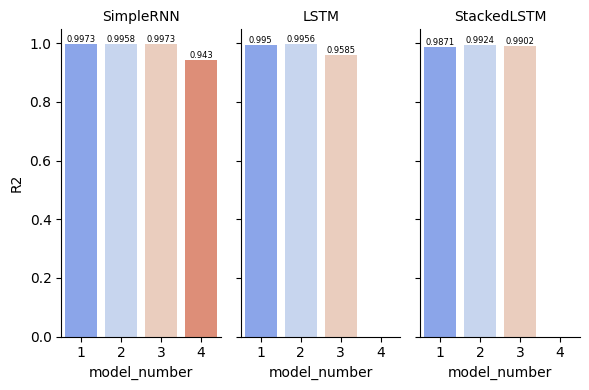


Рисунок 35 — series\_size равен 72

# 8 Выводы

В результате выполнения задания был построен ряд нейросетевых архитектур, решающих задачу регрессии с использованием рекуррентных нейронных сетей, и проведен их сравнительный анализ на основе заданных метрик. Большинство моделей продемонстрировали высокие результаты обучения.