

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Южно-Уральский государственный университет (национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

**РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ АВТОМАТИЗИРОВАННОЙ
ОБРАБОТКИ РАДИОЛОКАЦИОННОЙ ИНФОРМАЦИИ ДЛЯ
ПРОГНОЗИРОВАНИЯ ТРАЕКТОРИИ ДВИЖЕНИЯ ВОЗДУШНОГО
СУДНА НА ОСНОВЕ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ**

Рецензент:

начальник подразделения

1107 НИО ПО АО

НИИИТ-РК им. А.М. Брейгина

Л.В. Сивоплясов

Научный руководитель:

профессор кафедры СП, д.ф.-

м.н., доцент

М.Л. Цымблер

Автор:

студент группы КЭ-229

Я.И. Сотин

Челябинск, 2024

АКТУАЛЬНОСТЬ

- Есть спрос на улучшение качества систем управления воздушных систем в части траекторной обработки
- Требуются новые подходы в разработках систем ПВО
- Традиционные алгоритмы имеют ограничения



ЦЕЛИ И ЗАДАЧИ ИССЛЕДОВАНИЯ

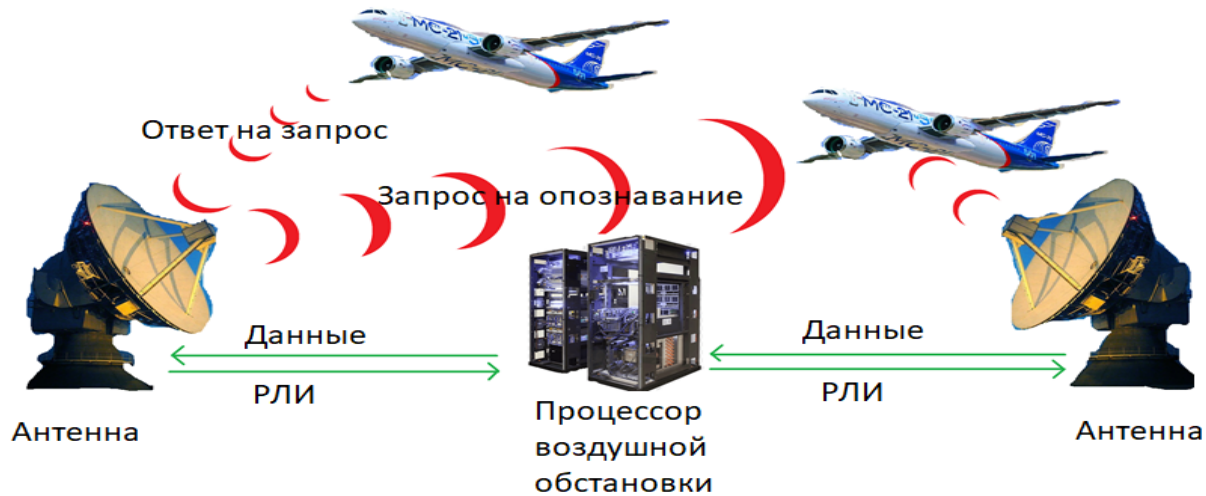
Цель:

Выяснить, возможно ли заменить существующие математические методы прогнозирования траектории движения нейросетевыми моделями. Повышение качества системы управления воздушным движением

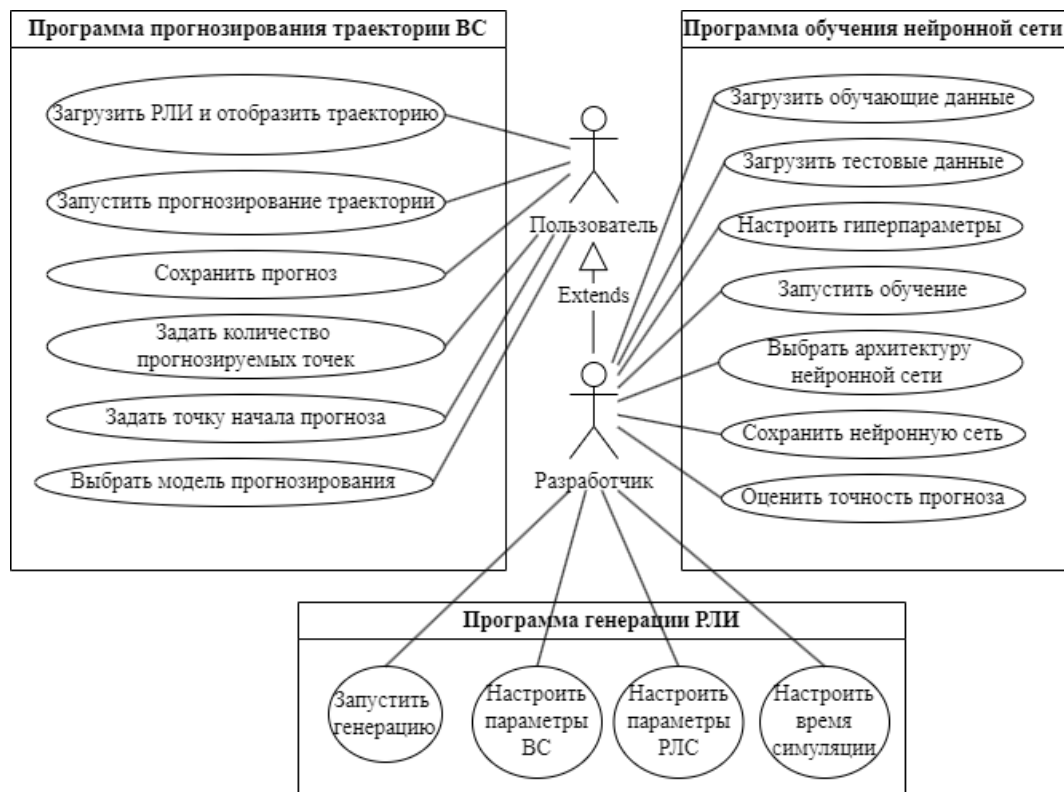
Задачи:

1. Провести анализ и обзор существующих решений и аналогов
2. Сгенерировать искусственные данные для обучения
3. Спроектировать и реализовать приложение
4. Провести тестирование разработанного приложения
5. Провести вычислительные эксперименты с обученными моделями и выяснить, какой метод эффективнее

ПРИНЦИП РАБОТЫ РЛС



МОДЕЛЬ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ



МАКЕТ ИНТЕРФЕЙСА ПРИЛОЖЕНИЯ



МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ТРАЕКТОРИИ САМОЛЁТА

▲	A	B	C	D	E	F
1		X	Y	Z	time	trace
2	0	0	0	0	19967	1
3	1	1179,712	-405,361	500	19977	1
4	2	2505,493	-887,469	1000	19987	1
5	3	3819,022	-1345,68	1500	19997	1
6	4	5144,622	-1816,13	2000	20007	1
7	5	6528,993	-1878,04	2500	20017	1

$$X = X_0 + V_0 \cdot \cos(\gamma) \cdot t + \frac{a \cdot \cos(\gamma) \cdot t^2}{2},$$

$$Y = Y_0 + V_0 \cdot \sin(\gamma) \cdot t + \frac{a \cdot \sin(\gamma) \cdot t^2}{2}$$

$$Z = Z_0 + V_0 \cdot \cos(\theta) \cdot t + \frac{a \cdot \cos(\theta) \cdot t^2}{2}$$

$$\gamma_{max} = \frac{1}{\left(\frac{V_0^2}{25}\right)^2}$$

$$\theta_{max} = \frac{1}{\left(\frac{V_0^2}{15}\right)^2}$$

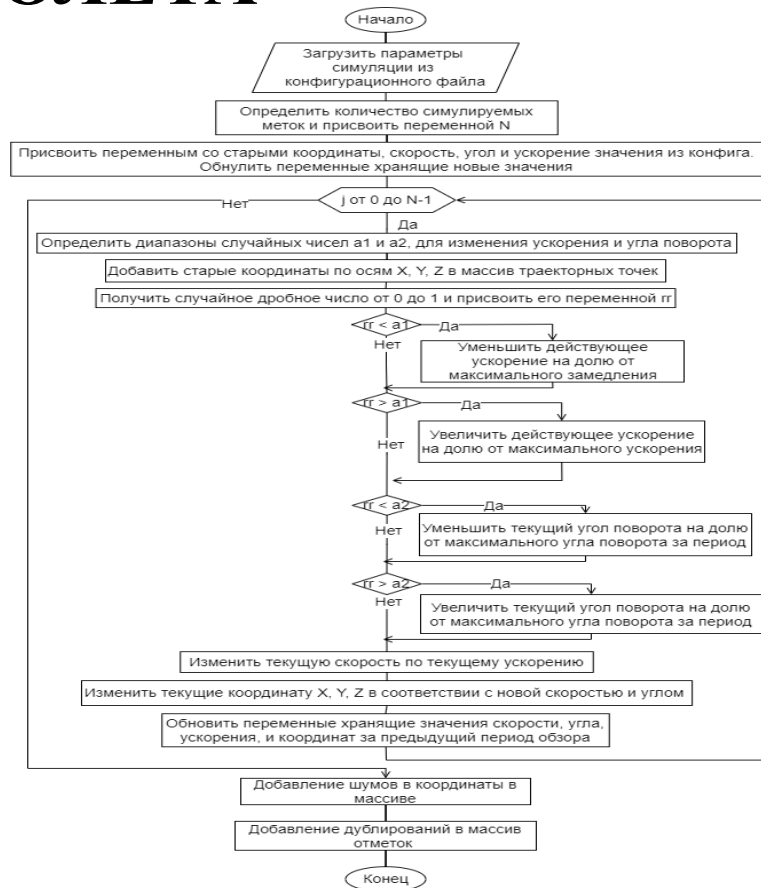
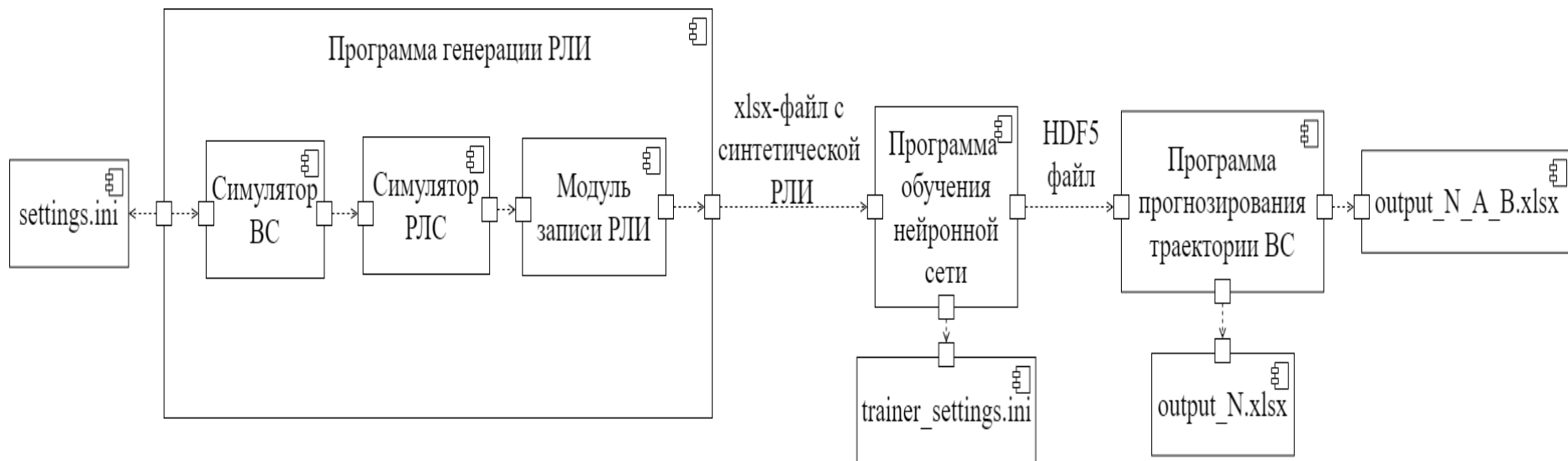
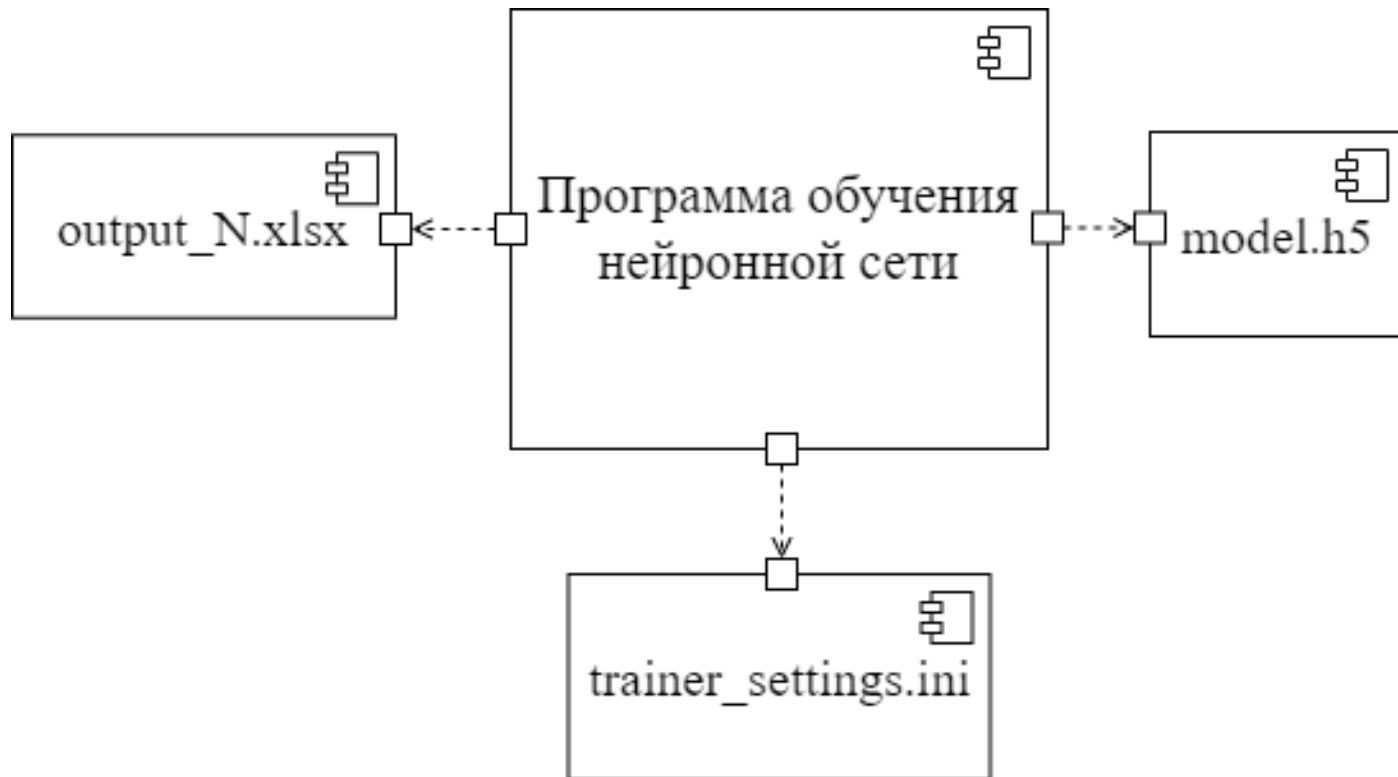


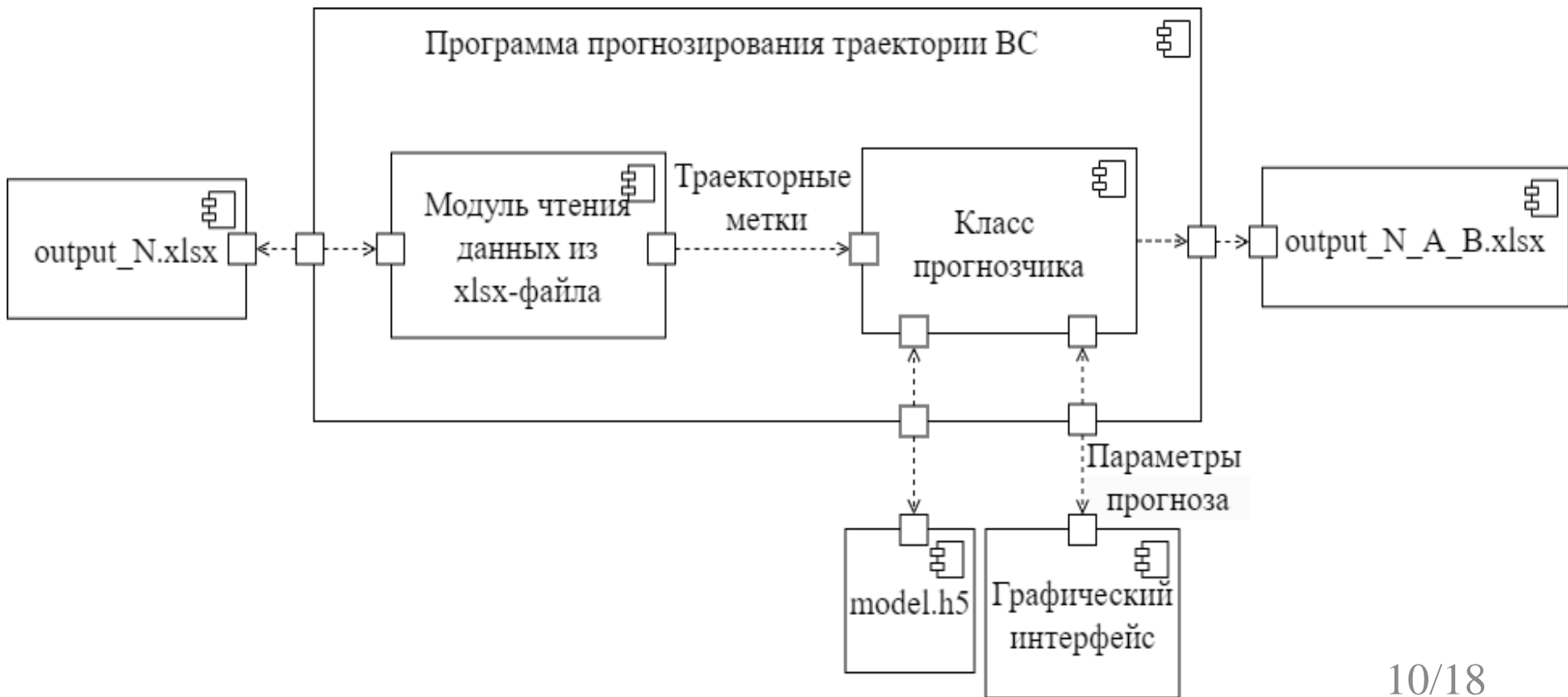
ДИАГРАММА КОМПОНЕНТОВ ПРИЛОЖЕНИЯ



КОМПОНЕНТ «ОБУЧЕНИЕ НЕЙРОСЕТИ»



КОМПОНЕНТ «ПРОГНОЗ ТРАЕКТОРИИ»



РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ

Инструменты реализации:

- Среда разработки Qt Creator для реализации работы интерфейса
- Облачная среда разработки Google Collab (для работы программы обучения)
- Библиотека глубокого машинного обучения keras версии 2.15
- Библиотека глубокого машинного обучения tensorflow версии 2.15
- Библиотека для анализа временных рядов statsmodels версии 0.14
- Язык программирования Python версии 3.1

РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ

Тестирование:

Была проверена работа интерфейса: функция загрузки файла с РЛИ, отображение траектории на экране, прогнозирование

Проверена программа генерации РЛИ, программа генерировала xlsh-файл с траекториями характерными для заданного самолёта

Проверена программа обучения НС, был сгенерирован файл HDF5

URL-ссылка на репозиторий с приложением:

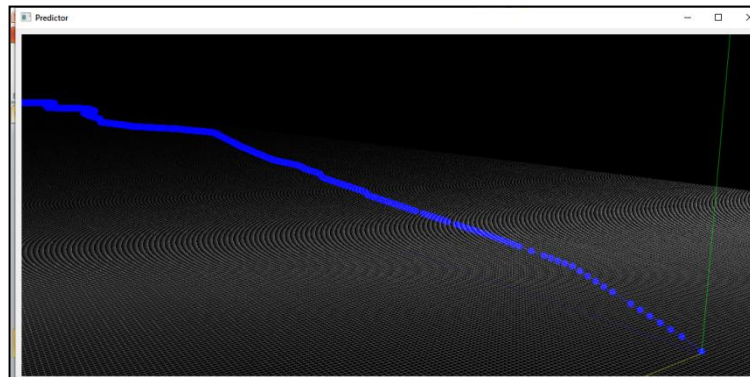
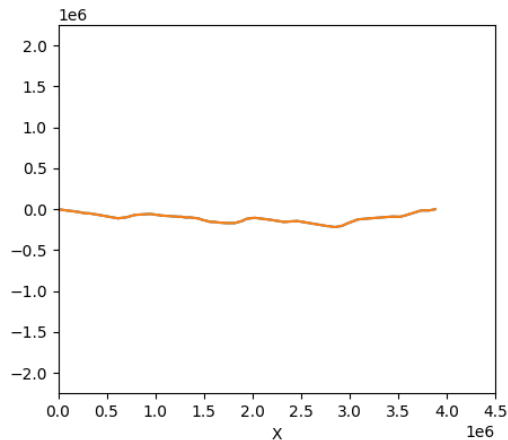
<https://github.com/Smusyc/Diploma-2024.git>

ЦИКЛ ИСПОЛЬЗОВАНИЯ ИССЛЕДОВАТЕЛЕМ



ТЕСТИРОВАНИЕ ГЕНЕРАЦИИ

```
1 [DataGenerator]
2 starting_val = 5
3 T = 14400
4 delta_t = 10
5 start_hour = 5
6 start_minutes = 32
7 start_second = 47
8 X_start = 0
9 Y_start = 0
10 max_pos_boost = 6.5
11 max_neg_boost = 3.5
12 initial_boost = 0
13 min_speed = 140
14 max_speed = 250
15 initial_speed = 125
16 initial_angle = 1
17 max_angle_change = 20
18 max_angle = 20
19 prob_boost_change = 0.1
20 prob_angle_change = 0.05
21 prob_timeshift = 0.005
22 prob_missing = 0.10
23 prob_dubbling = 0.005
24 prob_inaccuracy = 1
25 inaccuracy_wide_x = 10
26 inaccuracy_wide_y = 10
27 max_height_speed=50
28 middle_height=7500
29 Z_start=0
30 count_of_traces = 1
```

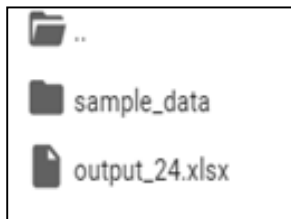


```
Администратор: C:\Windows\System32\cmd.exe
C:\Users\Yakov\PycharmProjects\PredictorProject>python data_generator.py
['output_1.csv', 'output_1.xlsx', 'output_10.xlsx', 'output_11.xlsx', 'output_12.xlsx',
'output_17.xlsx', 'output_18.xlsx', 'output_19.xlsx', 'output_2.csv', 'output_2.xlsx', 'output_2
'output_25.xlsx', 'output_26.xlsx', 'output_27.xlsx', 'output_28.xlsx', 'output_29.xlsx',
'output_4.xlsx', 'output_5.xlsx', 'output_6.xlsx', 'output_7.xlsx', 'output_8.xlsx', 'output_9.
C:\Users\Yakov\PycharmProjects\PredictorProject>
```

	A	B	C	D	E	F	G
1		X	Y	Z	time	trace	
2	0	0	0	0	19967	1	
3	1	1153,64	-498,482	500	19977	1	
4	2	2529,577	-660,995	1000	19987	1	
5	3	3927,567	-831,86	1500	19997	1	
6	4	5325,203	-890,415	2000	20007	1	
7	5	6720,982	-918,624	2500	20017	1	

ТЕСТИРОВАНИЕ ОБУЧЕНИЯ

```
settings_trainer.ini
1 [NNTrainer]
2   epoches = 5
3   metrics = accuracy
4   loss = mean_squared_error
5   optimizer = SGD
6   optimizer_settings = SGD_1
7   batch_size = 2
8   architecture = RNN_1
9
10 [RNN_1]
11   layer1 = SimpleRNN, 50, tanh
12   layer2 = Dropout, 0.2
13   layer3 = SimpleRNN, 50, tanh
14   layer4 = SimpleRNN, 50, tanh
15   layer5 = SimpleRNN, 50, tanh
16   layer6 = Dense, 1, sigmoid
17
18 [SGD_1]
19   momentum = 0.9
20   nesterov = True
21
```

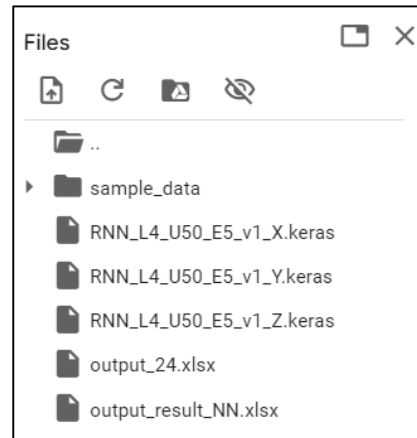


```
X_train : (850, 50, 1) y_train : (850, 1)
X_test : (357, 50, 1) y_test : (357, 1)
Epoch 1/5
425/425 [=====] - 61s 133ms/step - loss: 0.0074
Epoch 2/5
425/425 [=====] - 56s 132ms/step - loss: 0.0068
Epoch 3/5
425/425 [=====] - 55s 130ms/step - loss: 0.0067
Epoch 4/5
425/425 [=====] - 55s 130ms/step - loss: 0.0067
Epoch 5/5
425/425 [=====] - 55s 129ms/step - loss: 0.0067
Model: "sequential"
```

Layer (type)	Output Shape	Param #
simple_rnn (SimpleRNN)	(None, 50, 50)	2600
dropout (Dropout)	(None, 50, 50)	0
simple_rnn_1 (SimpleRNN)	(None, 50, 50)	5050
simple_rnn_2 (SimpleRNN)	(None, 50, 50)	5050
simple_rnn_3 (SimpleRNN)	(None, 50)	5050
dense (Dense)	(None, 1)	51

```
-----
Total params: 17801 (69.54 KB)
Trainable params: 17801 (69.54 KB)
Non-trainable params: 0 (0.00 Byte)
```

```
-----
12/12 [=====] - 1s 18ms/step
[[0.45025552]
 [0.37378752]
```

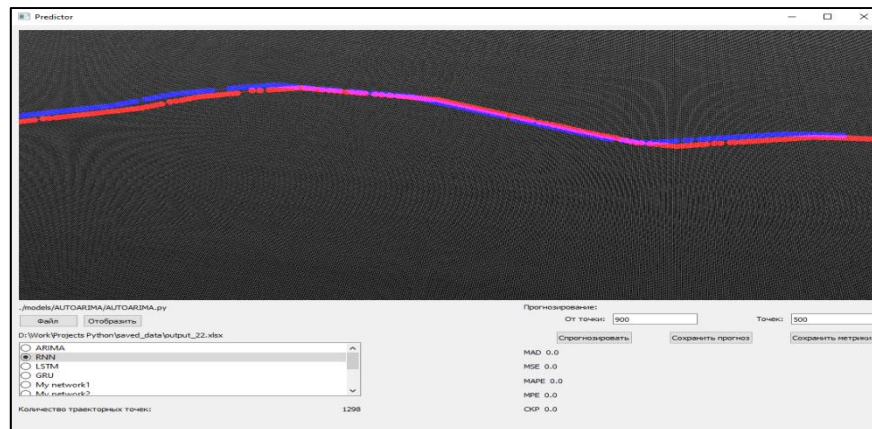
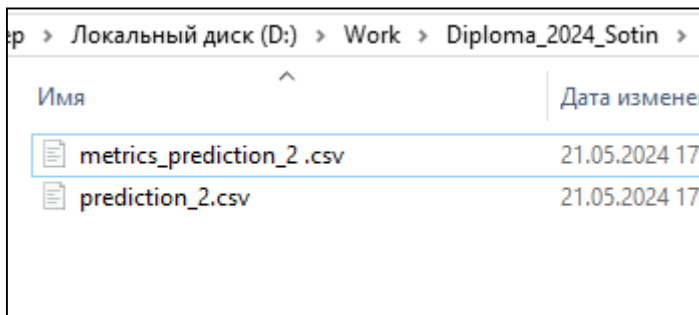
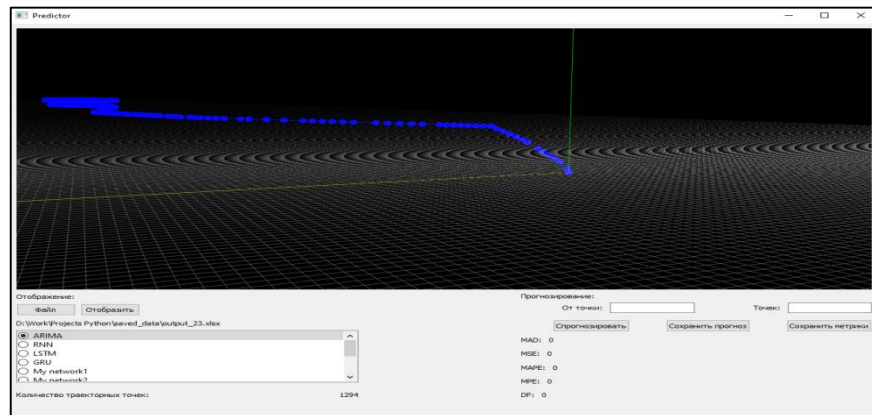
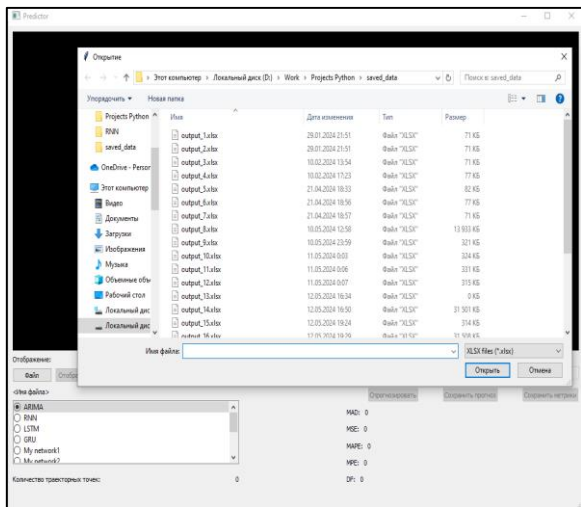


Упорядочить Создать

ер > Локальный диск (D:) > Work > Diploma_2024_Sotin > Diploma-2024 > Predictor > models > RNN

Имя	Дата изменения	Тип	Размер
RNN.py	21.05.2024 17:46	Python File	13 КБ
RNN_L4_U50_E5_v1_X.h5	21.05.2024 17:46	Файл "H5"	187 КБ
RNN_L4_U50_E5_v1_X.keras	21.05.2024 17:46	Файл "KERAS"	182 КБ
RNN_L4_U50_E5_v1_Y.h5	21.05.2024 17:46	Файл "H5"	187 КБ
RNN_L4_U50_E5_v1_Y.keras	21.05.2024 17:46	Файл "KERAS"	187 КБ
RNN_L4_U50_E5_v1_Z.h5	21.05.2024 17:46	Файл "H5"	187 КБ
RNN_L4_U50_E5_v1_Z.keras	21.05.2024 17:46	Файл "KERAS"	182 КБ
RNN_trainer.py	21.05.2024 17:46	Python File	6 КБ

ТЕСТИРОВАНИЕ ПРОГНОЗИРОВАНИЯ

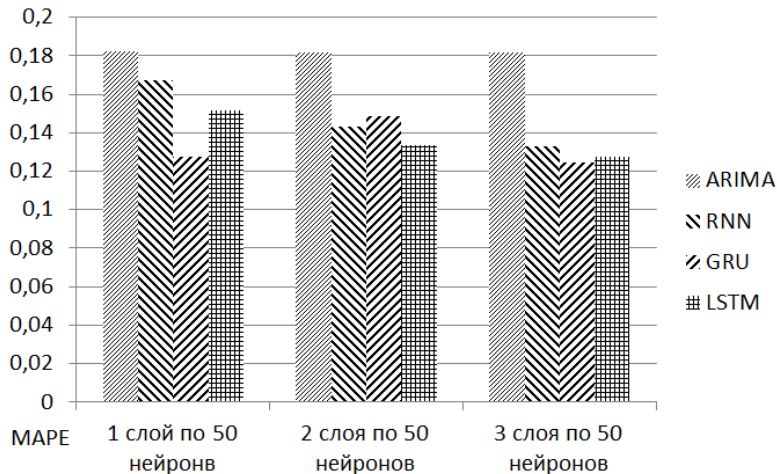


ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ

Для выявления самой эффективной модели, было проведено 9 испытаний всех трёх моделей, но в разных конфигурациях

Была использована РЛИ с 500 траекторий ВС, что соответствует среднему трафику пассажирского судна за полгода

В качестве меры точности сетей была использована регрессионная метрика MAPE (средняя абсолютная процентная ошибка)



$$MAPE = \frac{1}{n} \cdot \sum \frac{|Y_i - \hat{Y}_i|}{Y_i}$$

где Y_i – фактическое значение;

\hat{Y}_i – значение прогнозной модели за анализируемый период;

n – количество периодов.

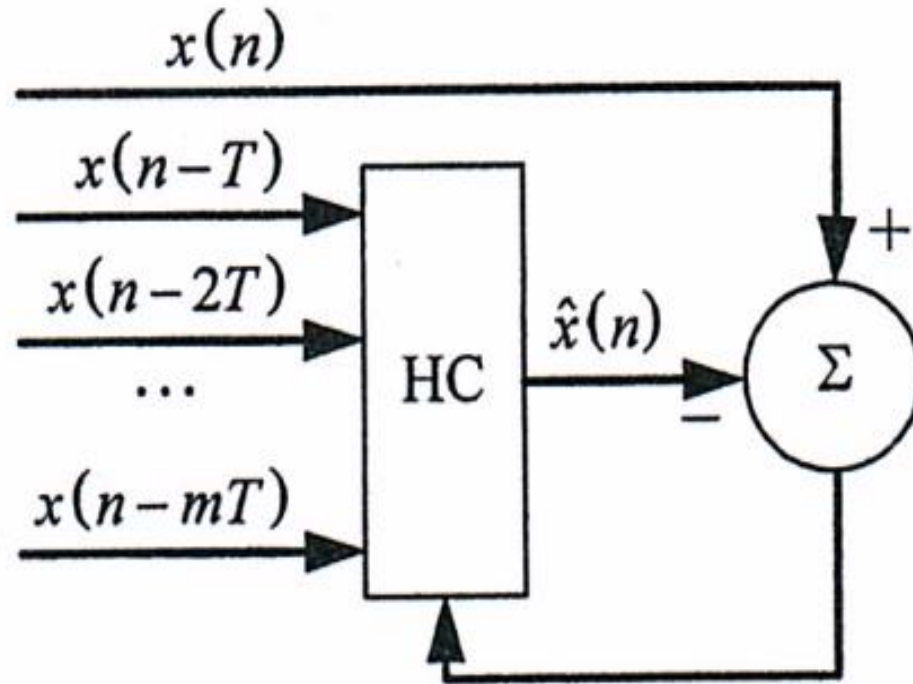
ОСНОВНЫЕ РЕЗУЛЬТАТЫ

В рамках данной выпускной работы были выполнены следующие задачи:

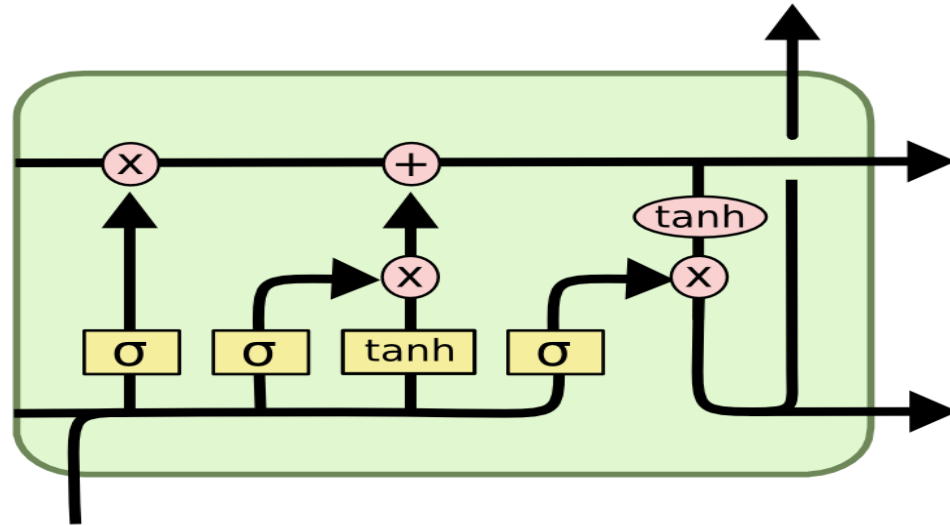
1. Проведён анализ и обзор существующих решений и аналогов
2. Сгенерированы искусственные данные для обучения
3. Спроектировано и разработано приложение
4. Проведено тестирование разработанного приложения
5. Проведены вычислительные эксперименты с обученными моделями и выявлен наиболее эффективный метод прогнозирования

ДОПОЛНИТЕЛЬНЫЕ СЛАЙДЫ

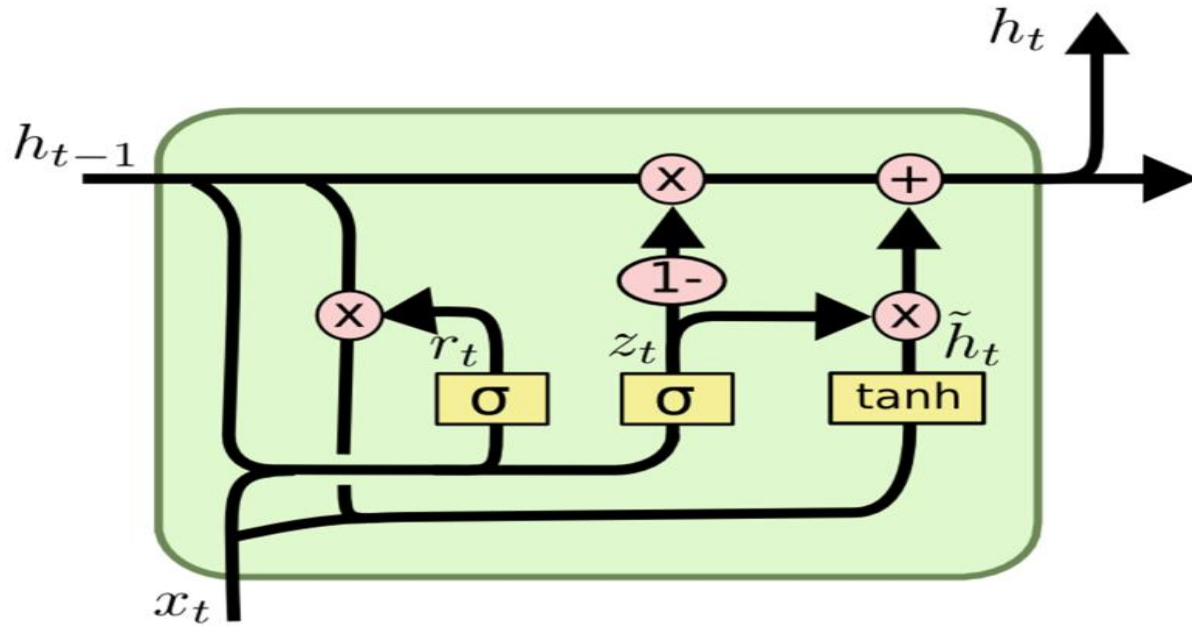
RNN



LSTM



GRU



СРАВНЕНИЕ МОДЕЛЕЙ

Свойства	RNN	LSTM	GRU
Скорость	1	2	3
Простота	3	1	2
Способность запоминать весь контекст	3	2	1
Способность запоминать важную информацию	1	3	2

ОБЗОР РАБОТ БЛИЗКИХ К ТЕМАТИКЕ ИССЛЕДОВАНИЯ

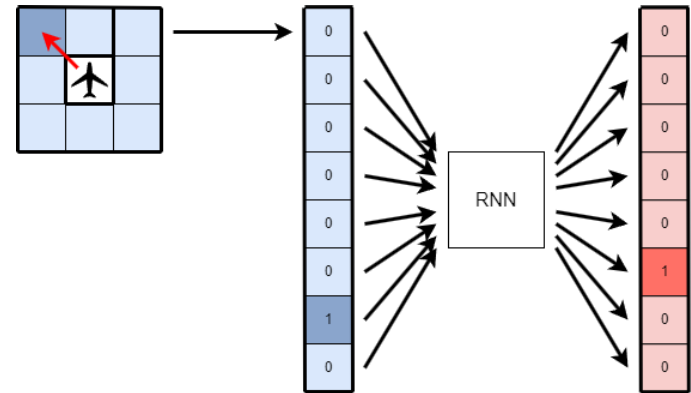
В статье «Прогнозирование траектории движения судна при помощи нейронной сети» авторы использовали нейронную сеть как регулятор управления движением корабля по заданной траектории.

Использовали фокусированную сеть прямого распространения с задержкой по времени, где на вход подавались координаты с предыдущих моментов времени, а на выходе прогноз следующей координаты. В качестве алгоритма обучения использовался метод Левенберга Маквардта, реализующий идею градиентного спуска.

Обученная сеть смогла превзойти по точности традиционно используемый для этих целей раннее фильтр Калмана.

ОБЗОР РАБОТ БЛИЗКИХ К ТЕМАТИКЕ ИССЛЕДОВАНИЯ

В статье «Применение искусственной нейронной сети для решения задач прогнозирования движения наземных объектов» использовали «пространственно закрепленную нейронную сеть» на основе RNN, но каждый отдельный нейрон соответствовал определенной точке на плоскости или области пространства, в котором осуществляется движение объекта. Можно сказать, что этот подход является идейным симбиозом рекуррентной и сверточной нейронных сетей.



ОБЗОР РАБОТ БЛИЗКИХ К ТЕМАТИКЕ ИССЛЕДОВАНИЯ

В статье «Trajectory prediction for moving objects using artificial neural networks» использовали НС для траектории движения робота пылесоса в помещении. Авторы статьи также обучали свою сеть на искусственных данных сгенерированные по кубическому полиному:

$$\chi(t) = \frac{1}{6}\alpha_0 t^3 + \frac{1}{2}\beta_0 t^2 + \gamma_0 t + \chi_0$$

где χ_0 – начальное положение;

$\alpha_0, \beta_0, \gamma_0$ – обновляющиеся коэффициенты для образования кривой.

В качестве входа использовались три параметра: дельты позиции, вектора скорости и вектора ускорения за временные периоды.

Данные были предварительно нормализованы.