

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**

РАБОТА ПРОВЕРЕНА

Рецензент
Начальник подразделения № 1107
НИО ПО АО НИИИТ-РК
им. А.М. Брейгина
_____ Л.В. Сивоплясов

« ___ » _____ 2024 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н.,
профессор
_____ Л.Б. Соколинский

« ___ » _____ 2024 г.

**Разработка приложения для автоматизированной обработки
радиолокационной информации для прогнозирования
траектории движения воздушного судна на основе
нейросетевых технологий**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ –09.04.04.2024.308-1500.ВКР

Научный руководитель,
профессор кафедры СП, д.ф.-м.н.,
доцент

_____ М.Л. Цымблер

Автор работы,
студент группы КЭ-229

_____ Я.И. Сотин

Ученый секретарь
(нормоконтролер)

_____ И.Д. Володченко

« ___ » _____ 2024 г.

Челябинск, 2024 г.

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**

УТВЕРЖДАЮ

Зав. кафедрой СП

_____ Л.Б. Соколинский

29.01.2024 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы магистранта

студенту группы КЭ-229

Сотину Якову Ивановичу,

обучающемуся по направлению

09.04.04«Программная инженерия»

(магистерская программа «Искусственный интеллект и инженерия данных»)

1. Тема работы (утверждена приказом ректора от 22.04.2024 г. № 764-13/12)

Разработка приложения для автоматизированной обработки

радиолокационной информации для прогнозирования траектории движения
воздушного судна на основе нейросетевых технологий.

2. Срок сдачи студентом законченной работы: 20.05.2024 г.

3. Исходные данные к работе

3.1. Hamilton J. D. A new approach to the economic analysis of nonstationary
time series and the business cycle. // *Econometrica: Journal of the econometric
society*, 1989. – С. 357–384.

3.2. Yang Z., Pang C., Zeng X. Trajectory Forecasting Using Graph
Convolutional Neural Networks Based on Prior Awareness and Information
Fusion. // *ISPRS International Journal of Geo-Information*, 2023. – Т. 12. –
№. 2. – 77 с.

3.3. Al-Chalabi H., Al-Douri Y. K., Lundberg J. Time Series Forecasting using
ARIMA Model: A Case Study of Mining Face Drilling Rig. // 12th International
Conference on Advanced Engineering Computing and Applications in Sciences
(ADVCOMP 2018), Athens, Greece, November 18–22, 2018. – International
Academy, Research and Industry Association (IARIA), 2018. – С. 1–3.

4. Перечень подлежащих разработке вопросов

- 4.1. Существующие аналоги в решении задач прогнозирования.
- 4.2. Разработать программу для генерации данных.
- 4.3. Разработать программу, способную решать задачу прогнозирования несколькими методами.
- 4.4. Сравнительный анализ результатов, полученных прогнозированием, через приложение, по нескольким методам.

5. Дата выдачи задания: 29.01.2024 г.

Научный руководитель,
профессор кафедры СП, д.ф.-м.н., доцент

М.Л. Цымблер

Задание принял к исполнению

Я.И. Сотин

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
1. СОВРЕМЕННЫЕ МОДЕЛИ И МЕТОДЫ РАДИОЛОКАЦИИ.....	9
1.1. Принцип работы РЛС и современные методы радиолокации	9
1.2. Методы прогнозирования траектории движения	12
1.2.1. Математические модели временных рядов.....	13
1.2.2. Нейронные сети.....	14
1.3. Обзор работ близких к тематике исследования.....	17
2. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	19
2.1. Модель вариантов использования.....	19
2.2. Сценарии вариантов использования	20
2.3. Интерфейс приложения.....	21
2.3.1. Требования к интерфейсу приложения и макет интерфейса ...	21
2.3.2. Описание интерфейса	22
3. ПРОЕКТИРОВАНИЕ	24
3.1. Диаграммы компонентов и требования к входным данным программы генерации синтетической РЛИ.....	24
3.2. Требования к входным и выходным данным прогнозирования .	28
4. РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ	29
4.1. Программа генерации синтетической РЛИ.....	29
4.2. Модели прогнозирования траектории	31
4.3. Тестирование	33
5. ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ.....	40
ЗАКЛЮЧЕНИЕ	40
ЛИТЕРАТУРА.....	42
ПРИЛОЖЕНИЕ. Алгоритм программы генерации РЛИ	45

ВВЕДЕНИЕ

Актуальность

В конце 30-х годов двадцатого века спустя 30 лет с момента создания первого радио, многие развитые страны начали разрабатывать и производить первые радиолокационные системы (РЛС) или радары (сокр. radio detection and ranging) для обнаружения воздушных, морских и наземных объектов.

С того момента было утверждено множество международных протоколов передачи радиосигналов для обеспечения охраны и безопасности воздушного пространства. Одной из задач радиолокации является определение направления движения воздушных судов в пространстве при помощи радиоволн.

В 1948 году Н. Винером была опубликована книга «Кибернетика: или Контроль и коммуникация у животных и машин», в которой была описана математическая модель биологического нейрона, а в 1958 году Ф. Розенблаттом была разработана и собрана первая однослойная нейронная сеть – перцептрон.

Эта нейронная сеть могла решать простые задачи классификации и распознавания. С того момента прошло более 60 лет, было разработано и реализовано множество концептов и моделей нейронных сетей, способных решать сложные нелинейные задачи.

Развитие новых архитектур стало возможным благодаря росту вычислительных мощностей видео ускорителей, которые и были выбраны для обработки данных в нейронных сетях.

На данный момент нейронные сети являются самым быстрорастущим и перспективным направлением развития технологий искусственного интеллекта. Так было выяснено, что нейронные сети способны решать задачи восстановления и прогнозирования временных рядов.

Такому подходу к анализу временных рядов можно найти широкое применение в финансовой деятельности [1, 2], медицине [3], системах связи [4] и языковом анализе [5].

К задачам прогнозирования временных рядов можно также отнести прогнозирование траектории движения объектов.

Были проведены исследования, которые доказали эффективность использование нейронных сетей в задачи прогнозирования траектории движения морских судов [6].

В ходе данной выпускной работы изучается возможность применение нейронных сетей для решения задачи прогнозирования траектории движения летательных аппаратов (ЛА).

Разрабатываемая система прогнозирования могла бы повысить эффективность контроля за воздушным пространством. Подобное ПО можно было бы применить как в гражданской, так и в военной сферах.

Постановка задачи

В рамках выпускной квалификационной работы необходимо разработать приложение для автоматизированной обработки радиолокационной информации и прогнозирования траектории движения воздушного судна при помощи нейронных сетей.

Сложность задачи прогнозирования движения воздушного судна заключается в том, что это движение включает маневры, выполняемые пилотом, из-за чего данная задача является нелинейной. Для обучения модели нейронной сети требуется данные – обучающая выборка.

В данном случае обучающей выборкой является данные о положении воздушного судна в ортогональной системе координат хуz [1]. Чем сложнее задача, тем больше требуется данных.

Данные (задокументированная радиолокационная информация) о траектории движения полета реальных воздушных судов, полученные при работе радиолокационных систем, имеются, но их недостаточно для обу-

чения нейронной сети такого качества, чтобы точность ее прогнозов была удовлетворительной.

Ввиду недостатка имеющихся данных решено использовать синтетические данные для расширения обучающей выборки.

Для генерации синтетических данных необходимо разработать программу, максимально правдоподобно имитирующую полет реального воздушного судна, а также максимально правдоподобно имитирующую процесс записи данных о положении детектируемого воздушного судна (процесс документирования радиолокационной информации) радиолокационной системой.

В рамках данной выпускной квалификационной работы будут выполнены следующие задачи:

- 1) разработка программы генерации синтетической радиолокационной информации по данным из симуляции;
- 2) проектирование дизайна пользовательского интерфейса приложения для прогнозирования траектории полета воздушного судна;
- 3) разработка приложения для прогнозирования траектории полета воздушного судна в части управления через пользовательский интерфейс;
- 4) разработка нейросетевой модели для прогнозирования траектории движения воздушного судна по данным об уже пройденной траектории;
- 5) обучение, тестирование и отладка нейросетевой модели;
- 6) встраивание нейросетевой модели в приложение для прогнозирования;
- 7) тестирование и отладка приложения для прогнозирования траектории полета воздушного судна в части точности выдаваемого прогноза;
- 8) проведение вычислительных экспериментов и сравнение с другими методами прогнозирования.

Структура и содержание работы

Работа состоит из введения, четырех глав, заключения и списка литературы. Объем работы составляет 45 страниц, объем списка литературы – 25 источников.

В первой главе проведен анализ предметной области. Рассмотрены принципы работы радиолокационных систем (РЛС) и формирования радиолокационной информации (РЛИ). Приведен обзор методов решения задачи прогнозирования траектории и обзор уже существующих программных решений, научных статей по тематике исследования и аналогов. Проведен анализ их достоинств и недостатков на основе имеющейся теоретической базы.

Во второй главе описан процесс проектирования общей архитектуры разрабатываемого приложения. В главе приведены диаграммы взаимодействия приложения, перечислены требования к входным и выходным данным для модулей, а также описан макет интерфейса разрабатываемого приложения и назначение его компонентов.

В третьей главе описывается внутренняя структура компонентов программы и приведены их диаграммы.

В четвертой главе описан процесс реализации разрабатываемого приложения и его компонентов, содержится описание тестовых испытаний для приложения и приведены их результаты в виде снимков экрана. Приведены результаты тестирования работоспособности разработанного приложения.

В пятой главе проведены вычислительные эксперименты с разными моделями прогнозирования и их результаты. В конце главы на основе полученных данных будет сделано заключение об уровне эффективности каждой модели и от чего эта эффективность зависит, а также будут упомянуты проведенные работы в ходе разработки и перспективы разработанного приложения.

1. СОВРЕМЕННЫЕ МОДЕЛИ И МЕТОДЫ РАДИОЛОКАЦИИ

1.1. Принцип работы РЛС и современные методы радиолокации

Явление отражения радиоволн от объектов было обнаружено еще в 1897 году создателем радио А.С. Поповым, когда во время опытов по радиосвязи между кораблями он выяснил, что их корпуса создают радиопомехи [7].

Первая радиолокационная система (РЛС) была изобретена в 1935 году шотландским физиком Робертом Уотсон-Уоттом, который примерно за год до этого получил первый патент на изобретение подобной системы.

Впервые о радиолокации как об области науки начали говорить в конце 30-х годов двадцатого века, когда в Великобритании было запущено производство промышленных образцов РЛС. Они были громоздки и их можно было располагать только на суше или на крупных кораблях. В 1937 году был испытан опытный образец компактной РЛС, пригодной для размещения на самолете [8].

Обнаружение и распознавание объектов, определение их местоположения в пространстве с помощью отраженных радиоволн называется радиолокацией. Радиолокационные системы также называются радиолокаторами или радарами [9]. Объекты радиолокации называются радиолокационными целями.

Зная скорость распространения радиоволны в пространстве, можно определить расстояние от радиопередатчика до цели по временному промежутку, за который отраженная от цели радиоволна вернулась обратно.

Современные РЛС состоят из процессора воздушной обстановки и антенны. Антенна излучает радиоволны в воздушное пространство вокруг себя и также ловит отраженный сигнал. По характеристикам полученного сигнала, а также по времени пути и углу поворота антенны в момент отправки сигнала, процессор определяет положение цели относительно передатчика. На рисунке 1 изображена схема взаимодействия ВС и РЛС.



Рисунок 1 – Принцип работы РЛС

Также, если на борту имеется устройство приема и ответа на радио-запросы (ответчик), то РЛИ может быть дополнена данными с датчиков самого воздушного судна (ВС), а также данными о цели. Такие данные передаются в зашифрованном виде для защиты от помех и искажений.

Данные, не очищенные от радиопомех, передаются процессору воздушной обстановки для обработки и отображения диспетчеру. С целью поддержки связи с воздушной целью, алгоритм отправки радиосигналов корректируется для сопровождения воздушной цели.

Данные в современных РЛС проходят три стадии обработки:

- 1) определение координат объекта по радиосигналам;
- 2) устранение ошибок и погрешностей;
- 3) сопряжение данных с нескольких РЛС.

Результатом первичной обработки является набор структур данных, описывающих радиолокационные метки (траекторная точка движения цели) [10].

Данные, прошедшие первичную обработку, могут содержать такие ошибки, как пропуск сигнала из-за шумов, а также задержки и ложное обнаружение, когда шумы были приняты за сигнал объекта и ложно иденти-

фицированы. Вероятность этих ошибок и интервал погрешности измерения описаны в технических характеристиках РЛС.

Во время вторичной обработки, система восстанавливает пропущенные метки объектов, уменьшает погрешности при измерении координат и отфильтровывает ложно идентифицированные цели.

Во время третьего этапа обработки, система занимается сбором данных из нескольких РЛС [11, 12, 13], чтобы сформировать общую картину.

В рамках данной работы наша РЛИ будет содержать только данные о координатах цели по осям Ox , Oy , Oz , времени обнаружения и идентификатор (номер) ее траектории.

Математическая модель радиолокации

Обычно в современных РЛС, из-за особенности конструкции антенно-излучателей, положение воздушной цели в пространстве однозначно определяется в полярной системе координат и имеет три параметра: R , ε , β (рисунок 2).

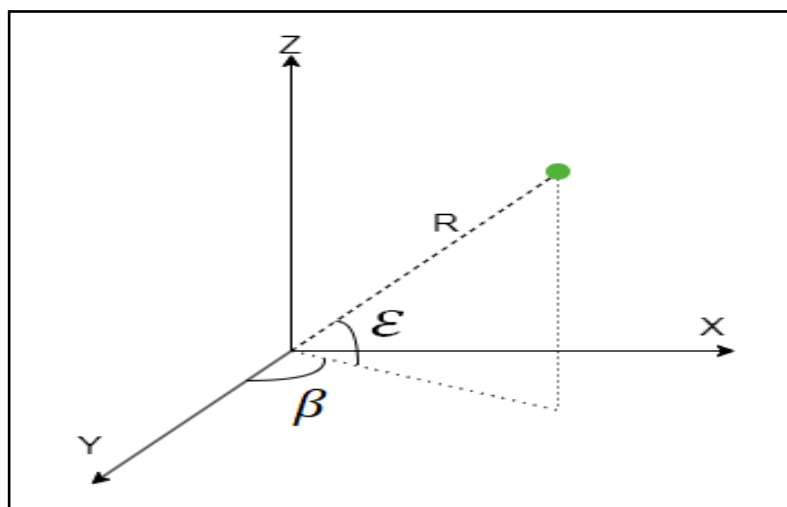


Рисунок 2 – Определение положения объекта в пространстве

Положение воздушного объекта (зеленая точка) в трехмерном пространстве определяется по трем параметра: R – радиус расстояния от антенны до цели, ε – угол высоты и β – угол поворота. Угол высоты и угол поворота определяются механическим положением антенны-излучателя, а расстояние определяется по времени отражения радиосигнала цели.

Однако для упрощения вычислений и более легкой интерпретации данных, координаты переводятся в декартову систему с осями X , Y и Z , где координата X соответствует широте, Y – соответствует долготе, а Z – высоте в метрах. Для определения X , Y , Z координат используются формулы (1)–(3):

$$X = R \cdot \cos(\varepsilon) \cdot \cos\left(\frac{\pi}{2} - \beta\right), \quad (1)$$

$$Y = R \cdot \cos(\varepsilon) \cdot \cos(\beta), \quad (2)$$

$$Z = R \cdot \sin(\varepsilon). \quad (3)$$

1.2. Методы прогнозирования траектории движения

Для прогнозирования движения объекта в пространстве необходимо вычислить закономерности его движения. Объекты по типу движения, обычно, разделяются на маневрирующие и не маневрирующие [14]. Если объект во время своего движения отклоняется от текущей траектории на определенный угол, тогда считается, что он маневрирующий. Любое движение, в общем случае, можно представить, как сложную функцию и потому во многих случаях при решении подобных задач используются такие модели, как: равномерное и прямолинейное движение, равноускоренное и прямолинейное, криволинейное и равноускоренное и т.д. В таком случае, прогнозируемую траекторную точку получают, либо методом линейной экстраполяции, либо с помощью метода наименьших квадратов, либо при помощи квадратичной экстраполяции [15].

Однако положение цели в следующий момент времени зависит от его положения в предыдущий (свойство автокорреляции), поэтому экстраполяция параметров движения объекта во временном периоде является задачей прогнозирования временных рядов.

1.2.1. Математические модели временных рядов

Исходя из вышесказанного, можно заключить, что предсказание значения, выдаваемого процессом $x(t)$ по предыдущим данным об этом процессе в предыдущие дискретные моменты времени и представленные значениями $x(n - T)$, $x(n - 2T)$, ..., $x(n - mT)$, где T – период снятия сигнала; m – порядок прогнозирования. Другими словами, временные ряды обладают свойством автокорреляции.

Автокорреляция – это статистическая взаимосвязь между последовательностями величин одного ряда.

Чтобы посчитать автокорреляцию, используется корреляция между временным рядом и ее сдвинутой копией от величины временного сдвига.

Также временные ряды могут обладать такими компонентами, как тренд, сезонность и ошибка (случайная компонента).

Тренд – это компонента, описывающая долгосрочное изменение уровня ряда.

Сезонность – компонента, которая описывает циклические изменения уровня ряда.

Ошибка (белый шум) – непрогнозируемая случайная компонента, описывает нерегулярные изменения в данных, необъяснимые другими компонентами.

Если временной ряд является стационарным, то его можно легко анализировать и прогнозировать.

Стационарность – это свойство временного ряда, которое означает, что его средние и стандартные отклонения не меняются со временем.

Нестационарный временной ряд может иметь тренд (постоянный рост или падение), цикличность (повторение циклов) или сезонность (повторение определенных событий в разное время года).

При простом анализе движения ВС, вряд ли можно говорить о какой-либо сезонности, цикличности или тренде. Хотя тренд может присутство-

вать на определенных участках временного ряда, как взлет или посадка. В таком случае можно говорить о не стационарности ряда.

Для прогнозирования временных рядов можно использовать специальные модели, такие как, авторегрессионная модель, скользящее среднее, и ARIMA.

Авторегрессионные модели (AR-модели) – модели, которые используют прошлые значения ряда для прогнозирования его будущих значений [16].

Скользящее среднее (MA-модели) – модели, которые используют прошлые значения ошибок – разницу между фактическими значениями ряда и его прогнозируемыми значениями для прогнозирования будущих значений.

ARIMA – модель, комбинирующая авторегрессионные и скользящие средние модели. ARIMA позволяет моделировать данные, не являющиеся стационарными, как это не требуется для AR и MA моделей. I (для «интегрированного») указывает, что значения данных были заменены разницей между их значениями и предыдущими значениями. ARIMA включает три параметра: параметр авторегрессии p , параметр скользящего среднего q и параметр интегрирования d . Подход ARIMA к временным рядам заключается в том, что в первую очередь оценивается стационарность ряда.

1.2.2. Нейронные сети

Для прогнозирования временных рядов широко и успешно применяются нейронные сети. Нейронные сети, благодаря своей архитектуре и механизму обучения, позволяют находить сколь угодно сложные решения задач, требующих решение в виде полинома больших степеней. При этом коэффициенты этого полинома корректируются автоматически во время обучения, в зависимости от квадрата разности предсказанных значений и реальных, приближаясь к минимуму ошибки, метод наименьших квадра-

тов встроен в сам процесс обучения нейронных сетей и автоматизирован, поэтому решено использовать нейросетевые модели.

С помощью нейронных сетей возможно динамически описывать математические модели полета объектов, основываясь не только на предыдущих наблюдениях анализируемого объекта, но и на наблюдениях других объектов подобного типа.

Более того, нейросетевые модели способны находить прогноз у нестационарных временных рядов с очень низкой автокорреляцией.

Исходя из вышесказанного, целесообразно выбрать нейронные сети в качестве инструмента для прогнозирования временных рядов.

Нейросетевая модель, обученная на основе коррекции ошибок без учителя, может быть использована для решения этой проблемы, так как обучающие примеры получаются непосредственно от самого процесса. На рисунке 3 функция $x(n)$ выполняет роль желаемого отклика [17, 18].

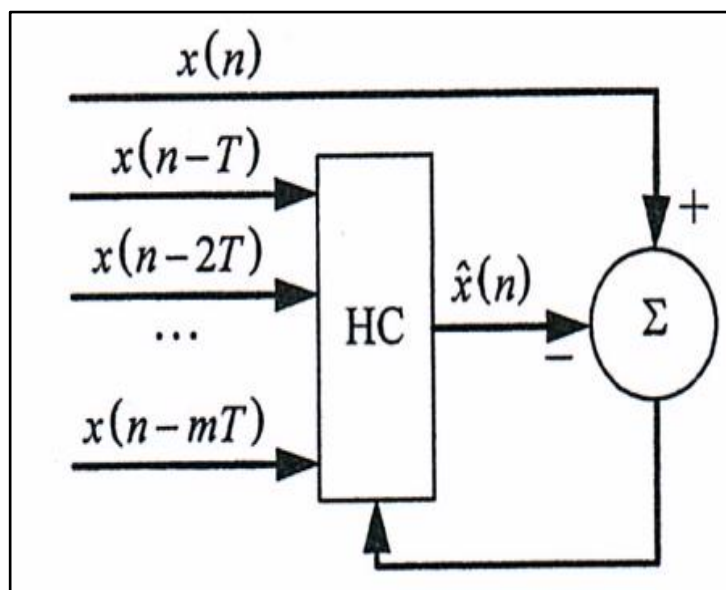


Рисунок 3 – Модель нейросети с обратной связью (отклик)

Здесь результат прогнозирования на один шаг вперед, сгенерированный НС в момент времени обозначается как n . Сигналы $x(n-T)$, ..., $x(n-mT)$ являются значениями прогнозируемого параметра в предыдущие моменты времени и подаются на вход нейронной сети, ввиду свойства

автокорреляции временного ряда. Сигнал ошибки определяется как разность между $x(n)$ и $\hat{x}(n)$.

Этот сигнал обратной связи используется для перенастройки параметров нейронной сети. Нейронные сети такого вида называются рекуррентными (RNN – Recurrent Neural Network).

Также возможно использование другой архитектуры. Рекуррентная нейронная сеть имеет недостаток в виде проблемы «затухающего градиента», когда при больших размерах временного ряда, между аномалиями (нетипичное поведение) присутствует большой временной период и их влияние на прогноз нивелируется – нейронная сеть теряет способность связывать информацию.

LSTM (Long short-term memory) нейронная сеть невосприимчива к большим временным лагам, обходит это ограничение и успешно применяется для прогнозирования временных рядов.

Также для решения поставленной задачи можно использовать модели, как GRU и CNN.

GRU (Gated Recurrent Unit), также является одним из видов рекуррентных нейронных сетей. В отличие от LSTM, в GRU нет отдельного долгосрочного состояния ячейки, как в LSTM. Вместо этого, она обновляет скрытое состояние напрямую, поэтому по структуре и параметрам GRU проще LSTM, но не так тонко фильтрует нужную информацию. Можно утверждать, что GRU больше подходит для относительно небольших временных рядов.

После обучения необходимо проверить корректность сети и ее адекватность. Для этого необходимо сравнить две траектории: прогнозируемую нейронной сетью и полученную из реальных данных, либо полученную из имитационных моделей, как в нашем случае. Для критерия разграничения траектории выберем наибольшее значение модуля разности соответствующих точек в интервале прогнозирования.

1.3. Обзор работ близких к тематике исследования

В своей статье «Прогнозирование траектории движения судна при помощи нейронной сети» [6] авторы исследовали возможность использование нейронной сети в качестве регулятора или компонента регулятора для управления движением корабля по заданной траектории. Задача прогнозирования траектории движения является составной частью задачи управления движением судна по заданному курсу, которую поставили перед собой авторы статьи. Была использована рекуррентная нейронная сеть (RNN), обученная на искусственно сгенерированных данных. Тестирование показало, что обученная нейронная сеть смогла добиться даже большей точности, чем традиционно используемый для этих целей раннее фильтр Калмана.

Особенностью их подхода является применение «пространственно закреплённой нейронной сети», когда нейронная сеть построена таким образом, чтобы каждый отдельный нейрон соответствовал определенной точке на плоскости или области пространства, в котором осуществляется движение объекта. Эта сеть также является рекуррентной, но не интерпретирует движение, как временной ряд изменения векторов скорости, а как координатную сеть, ячейки которой взаимосвязаны. Можно сказать, что этот подход является идейным симбиозом рекуррентной и сверточной нейронных сетей. Эксперимент показал хорошие результаты. Сам подход имеет преимущество в сокращении вычислений, но и имеет недостаток в виде дискретности прогнозов, зависящей от разрешающей способности выходного слоя сети.

В статье «Trajectory prediction for moving objects using artificial neural networks» [19] авторы изучали использование нейронных сетей для построения траектории движения роботов в замкнутом пространстве. Задача предсказания траектории движения является составной частью задачи построения траектории. Новизна подхода состоит в том, что в качестве входных параметров для нейронной сети выступают не только координаты по-

ложения робота, но и значения его скорости, угла поворота и ускорения, полученные от сенсоров. При этом, в качестве математической модели движения авторы выбрали свой кубический полином, а для оптимизации вычислений, использовались не сами значения параметров, а их нормализованная разность от предыдущих значений. В качестве нейронной сети они использовали перцептрон с тремя входами и одним выходом, состоящий из двух скрытых слоев с 20 нейронами в каждом. В статье подробно описаны процесс обучения и результаты работы сети.

В статье «Generative adversarial networks for spatio-temporal data: A survey» [20] автор изучал применение генеративно состязательных нейронных сетей (GAN) и провел всесторонний обзор последних разработок GAN для обработки пространственно-временных данных. В статье кратко описано применение популярных архитектур для обработки пространственно-временных данных и общепринятые методы оценки производительности пространственно-временных приложений с использованием таких сетей.

В статье «Generative adversarial networks for trajectory prediction and group detection in crowds» [21] авторами была представлена новая система глубокого обучения для прогнозирования траектории движения человека и определения принадлежности к социальной группе в толпе. Задача прогнозирования траектории является составной задачей, которую поставили перед собой авторы. В рамках исследования был разработан генеративный состязательный конвейер, который сохраняет пространственно-временную структуру окружения, в котором находится пешеход, позволяя извлекать соответствующие атрибуты, описывающие его социальную идентичность. Модель показала хорошие результаты и достаточно часто правильно определяла группы людей.

2. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

2.1. Модель вариантов использования

Цель разрабатываемого автором приложения – дать возможность пользователю получить прогноз траектории движения ВС из уже имеющейся траектории, оценить и сохранить его. Момент, с которого нужно получить прогноз, и количество прогнозируемых точек должны устанавливаться пользователем, как и метод прогнозирования.

На рисунке 4 представлена диаграмма вариантов использования приложения.

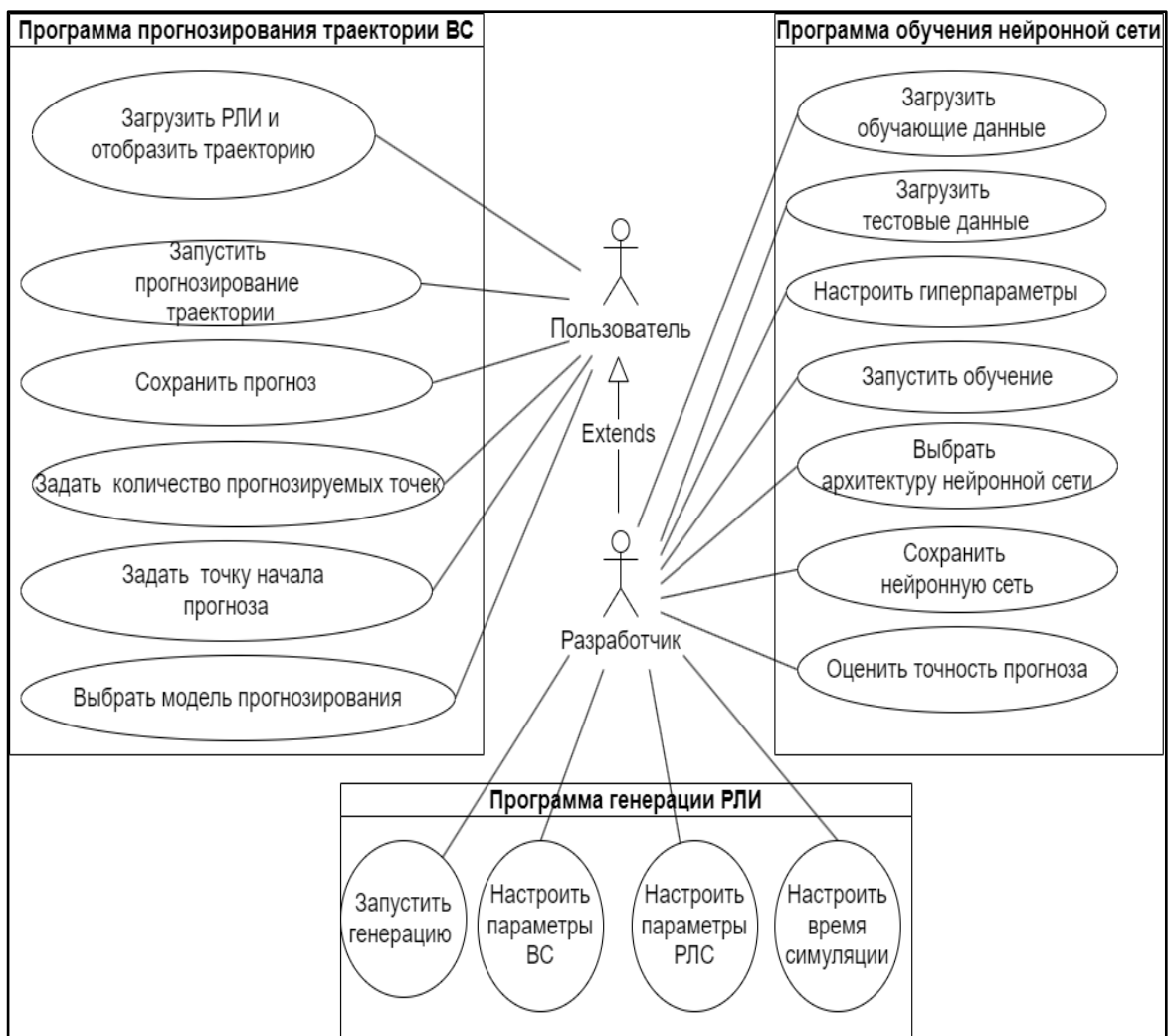


Рисунок 4 – Диаграмма вариантов использования

2.2. Сценарии вариантов использования

Генерация РЛИ

Разработчик записывает параметры РЛС, параметры ВС и время симмуляции в конфигурационный файл. После запуска программы (запуск генерации) через командную строку, запускается симмуляция полета судна и процесс документирования РЛС, и искусственные радиолокационные данные сохраняются в файл.

Обучение нейронной сети

Разработчик выбирает архитектуру нейронной сети, настраивает гиперпараметры сети, загружает обучающую и тестовую выборки с радиолокационной информацией и запускает обучение нейронной сети.

По окончании обучения, разработчик оценивает точность обученной модели: если она удовлетворительная, то он сохраняет нейросетевую модель с весами в виде файла.

Прогнозирование траектории ВС

Диспетчер нажимает кнопку «Файл», появляется окно файловой системы и выбирает там *xlsx*-файл для загрузки. После загрузки на экране отображается траектория полета ВС с траекторными точками. После загрузки файла становятся доступными поля для записи количества точек прогнозирования, и с какой точки начать прогноз. Пользователь выбирает модель для прогнозирования из списка моделей ниже нажатием на радиокнопку, задает номер точки, с которой будет проведен прогноз и количество прогнозируемых траекторных точек и запускает прогнозирование. После прогнозирования, на экране отображается спрогнозированная траектория красного цвета с красными траекторными точками. На панели управления, после прогноза отображается оценка точности прогноза несколькими метриками. Если полученный прогноз удовлетворителен, диспетчер сохраняет прогнозы нажатием на кнопку «Сохранить прогноз» в виде *xlsx*-файла вместе с отчетом о точности прогноза с метриками в виде текстового файла.

2.3. Интерфейс приложения

2.3.1. Требования к интерфейсу приложения и макет интерфейса

Данное приложение должно иметь пользовательский интерфейс, позволяющий:

- загрузить `xlsx`-файл с данными для отображения траектории и прогнозирования;
- задать количество прогнозируемых меток;
- задать индекс траекторной точки ряда, с которого начать прогноз;
- запустить программу прогнозирования;
- визуализировать траектории по загруженным и спрогнозированным траекторным точкам положения воздушного судна;
- загрузить `xlsx`-файл со списком спрогнозированных отметок того же формата, что и входной файл;
- отобразить результаты точности прогноза в виде метрик;
- загрузить отчет с результатами точности;
- отобразить итоговое количество траекторных точек на экране;
- выбрать модель для прогнозирования.

Отображение доступных для выбора моделей прогнозирования осуществляется по специальному конфигурационному файлу, содержащему названия моделей, пути к ним и их параметры по умолчанию. Выбор предоставляется в виде окна со списком.

В рамках выпускной квалификационной работы был разработан макет пользовательского интерфейса приложения для прогнозирования. На рисунке 5 показан планируемый результат вызова функции прогнозирования, где синей линией обозначена реальная траектория, а красной обозначена спрогнозированная.

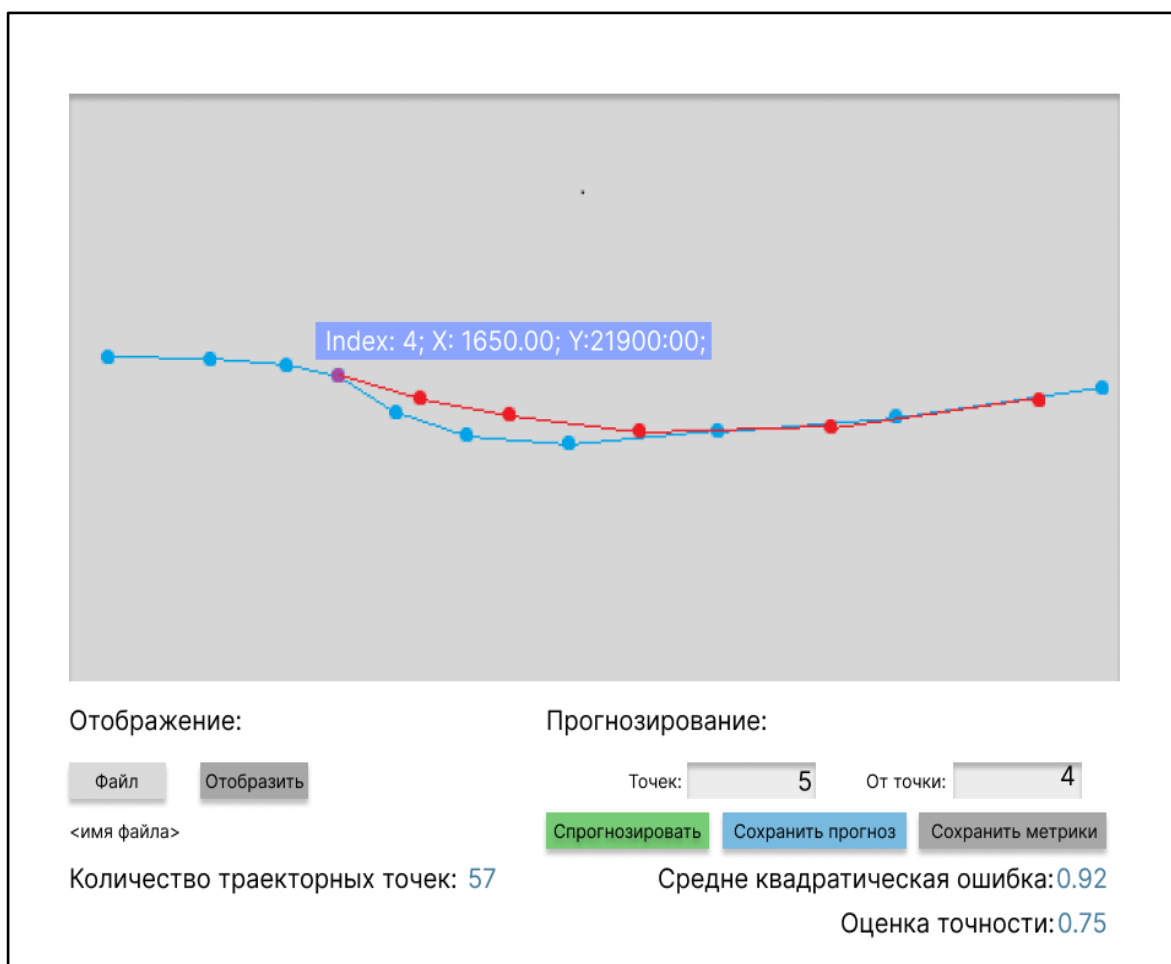


Рисунок 5 – Макет пользовательского интерфейса приложения

2.3.2. Описание интерфейса

В разрабатываемом интерфейсе приложения будут присутствовать перечисленные ниже компоненты.

Кнопка «Файл»

Используется для загрузки *xlsx*-файла с РЛИ для прогнозирования. При нажатии, должно открываться окно выбора файла в проводнике. После загрузки файла его имя будет отображено в надписи ниже. На экране появляется траектория с траекторными точками, окрашенная синим цветом.

Строка «Количество траекторных точек»

Отображает количество траекторных точек содержащихся в *xlsx*-файле. Необходимо пользователю для ориентации и понимания, с какой

точки можно начать прогнозирование. Появляется после нажатия на кнопку «Отобразить».

Поле с надписью «Точек»

Используется для хранения количества прогнозируемых траекторных отметок. Пользователь сам записывает в нее целое число. Если поле пустое или хранит отрицательное число, то по умолчанию при прогнозировании программа будет прогнозировать столько же точек, сколько было обнаружено в загружаемом файле, т.е. сгенерирует вторую траекторию той же длины. Если в поле будет вещественное число, то программа вычислит ближайшее к нему целое положительное число и будет использовать его в вычислениях.

Поле с надписью «От точки»

Позволяет пользователю выбрать, от какой точки начать прогнозирование. Нумерация точек начинается с 1. По умолчанию, если поле пустое равно 0 или отрицательное, то прогноз будет вестись от первой точки. Если в поле помещено вещественное число, то программа вычислит ближайшее к нему целое положительное число и будет использовать его в вычислениях.

Кнопка «Спрогнозировать»

Кнопка становится доступной после нажатия на кнопку «Отобразить» и при нажатии запускает программу прогнозирования с параметрами, указанными в полях «От точки» и «Точек».

Кнопка «Сохранить прогноз»

Кнопка становится доступной после нажатия на кнопку «Спрогнозировать» и при нажатии сохраняет спрогнозированные метки в файл формата xlsx.

Кнопка «Сохранить метрики»

Кнопка становится доступной после нажатия на кнопку «Спрогнозировать» и при нажатии сохраняет результаты сравнения реальной и спрогнозированной траекторий по метрикам в текстовый файл.

3. ПРОЕКТИРОВАНИЕ

3.1. Диаграммы компонентов и требования к входным данным программы генерации синтетической РЛИ

На рисунке 6 приведена диаграмма компонентов приложения.

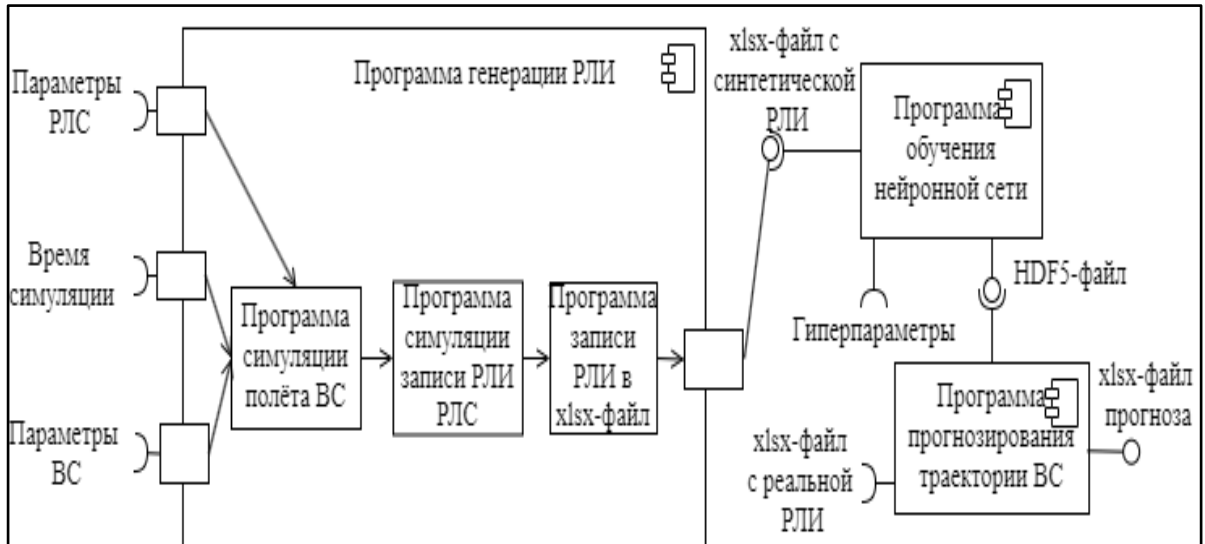


Рисунок 6 – Диаграмма компонентов

Программа генерации радиолокационной информации (РЛИ) состоит из трех модулей: модуля симуляции полета ВС и модуля симуляции документирования РЛС, модуль, записывающий РЛИ в файл .xlsx. Программа не имеет графического интерфейса и является отдельной утилитой, поэтому параметры симуляции должны приниматься из специального конфигурационного файла.

Для модуля симуляции ВС будут приняты следующие параметры:

- количество имитируемых ВС (траекторий);
- время симуляции в секундах;
- начальная координата X ;
- начальная координата Y ;
- начальная координата Z ;
- время начала отсчета в секундах;
- максимальное ускорение самолета;
- максимальное замедление самолета;

- начальное ускорение самолета;
- минимальная скорость самолета, при которой он не опускается;
- максимальная скорость самолета, которую он может развить;
- максимальный угол поворота самолета за период;
- начальный угол самолета;
- вероятность изменения ускорения самолета;
- вероятность изменение угла самолета.

Для модуля симуляции РЛС будут приняты следующие параметры:

- вероятность смещения метки по времени;
- вероятность потери метки;
- вероятность дублирования метки;
- вероятность погрешности в измерении координат метки;
- максимальная погрешность измерения координаты X ;
- максимальная погрешность измерения координаты Y ;
- начальная координата Z ;
- СКО измерения координаты Y ;
- начальная координата Z ;
- СКО измерения координаты X .

Для построения системы траекторных точек необходимо последовательно формировать новые точки координат со смещением, соответствующим проекции скорости судна на оси X , Y и Z . Для вычисления расстояния в качестве временной интервала принимается период обзора.

Ниже приведены формулы для вычисления координат на осях X , Z , Y (4)–(7):

$$X_{new} = X_{old} + V_0 \cdot \cos(\beta) \cdot t + \frac{(k \cdot \cos(\alpha)) \cdot t^2}{2}, \quad (4)$$

$$Y_{new} = Y_{old} + V_0 \cdot \sin(\beta) \cdot t + \frac{(k \cdot \sin(\beta)) \cdot t^2}{2}, \quad (5)$$

$$Z_{new} = Z_{old} + V_0 \cdot \sin(\beta) \cdot t + \frac{(k \cdot \sin(\gamma)) \cdot t^2}{2}, \quad (6)$$

$$V_{new} = V_{old} + k \cdot t. \quad (7)$$

где α – угол относительно оси Oх;

β – угол относительно оси Oу;

γ – угол относительно оси Oz;

t – время обзора;

V_0 – начальная скорость;

k – ускорение;

Y_{old} и Y_{new} – старая и новая координата Y;

X_{old} и X_{new} – старая и новая координата X;

Z_{old} и Z_{new} – старая и новая координата Z;

V_{old} и V_{new} – старая и новая начальная скорость.

При использовании этих формул в нашем алгоритме построения системы траекторных точек скорость воздушного судна в каждой точке изменяется и используется как начальная для следующей.

Параметры: ускорение и угол к оси Oх должны быть случайными числами при следующих ограничениях:

- в соответствии с техническими характеристиками стандартного пассажирского судна, его максимальное ускорение может быть не более $6,5 \text{ м/с}^2$, а максимальное отрицательное ускорение при равно замедленном движении – не более $3,5 \text{ м/с}^2$;

- максимальное изменение угла курса воздушного судна относительно оси Oх за период обзора – не более 5° ;

- максимальная развиваемая скорость и минимальная скорость движения воздушного судна для нахождения в воздухе;

- вероятность изменения ускорения и угла курса – 0,1.

При соблюдении данных условий можно построить траекторию, достаточно приближенную к траектории полета реального воздушного судна.

На рисунке 7 представлена внутренняя структура компонента «Программа обучения нейронной сети».

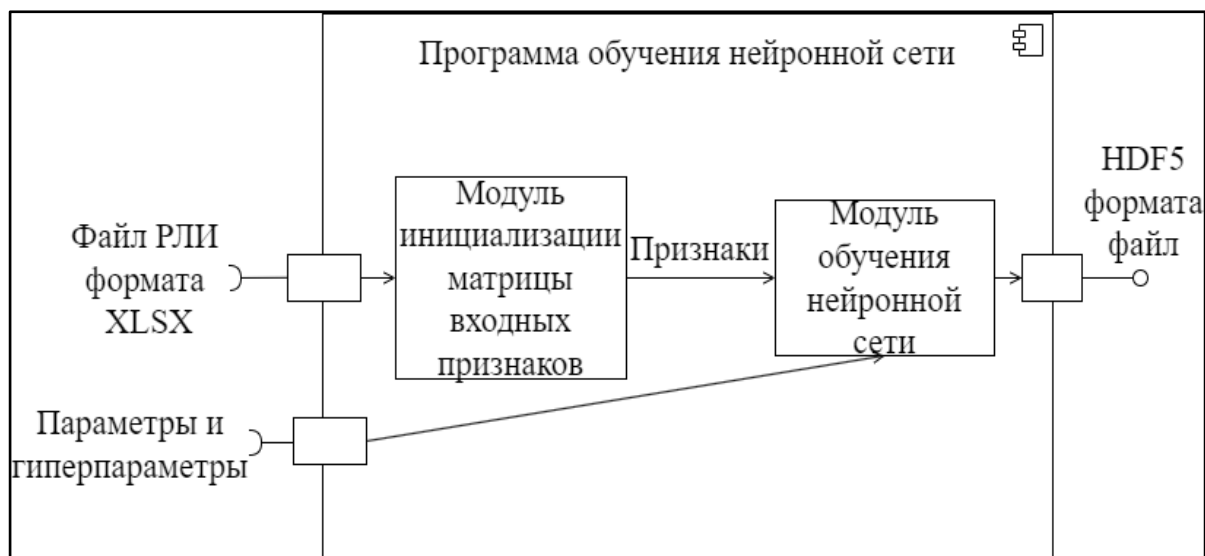


Рисунок 7 – Внутренняя структура компонента «Программа обучения нейронной сети»

На рисунке 8 представлена внутренняя структура компонента «Программа прогнозирования траектории ВС».

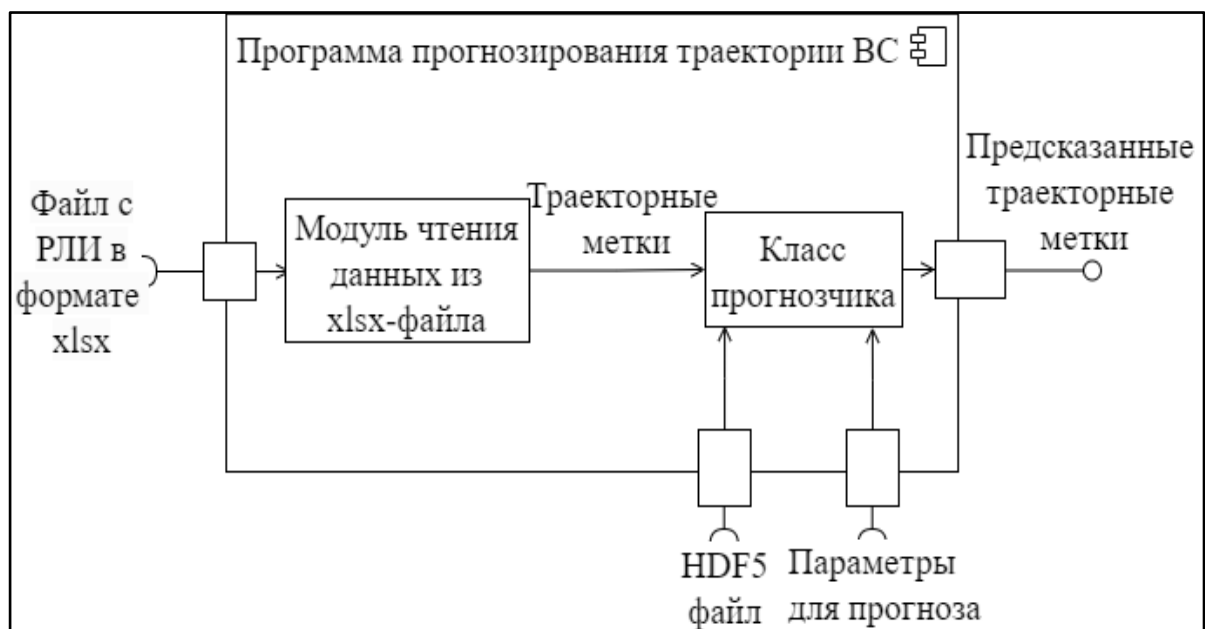


Рисунок 8 – Внутренняя структура компонента «Программа прогнозирования траектории ВС»

3.2. Требования к входным и выходным данным прогнозирования

Программа прогнозирования в качестве входных данных принимает файл формата .xlsx, выбранный пользователем в меню моделей метод прогнозирования, а также два входных числа: номер траекторной точки, с которой начинается прогнозирование и количество траекторных точек, которые необходимо спрогнозировать (число n). Так как программа является утилитой и не имеет графического интерфейса, параметры обучения загружаются из конфигурационного файла.

В файле с входными данными должна содержаться таблица, состоящая из столбцов: X , Y , Z , время и номер траектории, к которой принадлежит соответствующая метка. Каждая запись в этой таблице описывает траекторную точку движения воздушного судна на карте.

Поля X , Y , Z должны хранить координаты положения воздушного судна в пространстве, а поле «время» – время, когда эти координаты были зафиксированы.

Поле с номером траектории необходимо для различия ВС и их траекторий при множественном отображении, а также в обучающей выборке для программы обучения нейронной сети. Метки выдаются с определенной частотой, которая является индивидуальной физической характеристикой для каждой модели радиолокационной системы, которая генерирует радиолокационную информацию.

В качестве выходных данных приложение должно отобразить на экране и вернуть список из n спрогнозированных меток того же формата, что и входные данные с метками, если будет вызвана функция сохранения прогноза.

Также должно присутствовать отображение результатов оценки точности прогноза, вычисленных по нескольким метрикам. Должна быть возможность загрузить протокол с метриками точности прогноза в виде текстового файла.

4. РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ

4.1. Программа генерации синтетической РЛИ

В рамках выпускной квалификационной работы было необходимо разработать и отладить алгоритм программы симуляции полета ВС в части генерации траекторных отметок с заданным временным периодом.

Реализация программы

На рисунке 1 приложения приведена блок-схема алгоритма генерации траекторных меток.

Входные данные

В качестве входных параметров функция принимает следующие переменные:

- T – время симуляции в секундах;
- `start_hout` – часы времени начала симуляции;
- `start_minutes` – минуты времени начала симуляции;
- `start_second` – секунды времени начала симуляции;
- `delta_t` – период обзора радиолокационной системы в секундах;
- `start_boost` – начальное ускорение воздушного судна м/с^2 ;
- `start_speed` – начальная скорость воздушного судна в м/с ;
- `X_start` – начальное положение координат по оси X в метрах;
- `Y_start` – начальное положение координат по оси Y в метрах;
- `start_angle` – начальный угол относительно оси X в градусах;
- `max_angle_change` – максимальный угол поворота курса воздушного судна за период обзора в градусах;
- `max_speed` – максимальная в м/с , которое может развивать воздушное судно;
- `max_pos_boost` – максимальное положительное ускорение для воздушного судна в м/с^2 ;
- `max_neg_boost` – максимальное отрицательное ускорение для воздушного судна в м/с^2 ;

- `min_speed` – минимальная скорость в м/с, при котором воздушное судно может оставаться в воздухе;
- `initial_speed` – начальная скорость в симуляции;
- `initial_angle` – начальный угол поворота в симуляции;
- `max_angle` – максимальное отклонение от курса;
- `prob_boost_change` – вероятность изменения ускорения;
- `prob_boost_angle` – вероятность изменения угла;
- `max_height_speed` – максимальная скорость подъема;
- `middle_height` – средняя высота полета самолета;
- `min_speed` – минимальная скорость в м/с, при котором воздушное судно может оставаться в воздухе;
- `prob_timeshift` – вероятность смещения метки по времени;
- `prob_missing` – вероятность пропуска метки;
- `prob_dubbling` – вероятность дублирования метки;
- `prob_inaccuracy` – вероятность погрешности в измерении координат;
- `inaccuracy_wide_x` – средняя погрешность измерения координаты X (из СКО из характеристик РЛС);
- `inaccuracy_wide_y` – средняя погрешность измерения координаты Y (из СКО из характеристик РЛС);
- `inaccuracy_wide_z` – средняя погрешность измерения координаты Z (из СКО из характеристик РЛС);
- `count_of_traces` – количество траекторий.

В качестве выходных данных функция должна возвращать массив картежей, каждый из которых описывает соответствующую траекторную отметку воздушного судна и хранит три числа – координаты X, Y и Z. Также в каждой метке хранится идентификатор траектории и время, когда метка был зафиксирована.

Реализация интерфейса

На рисунке 9 приведено изображение разработанного интерфейса.

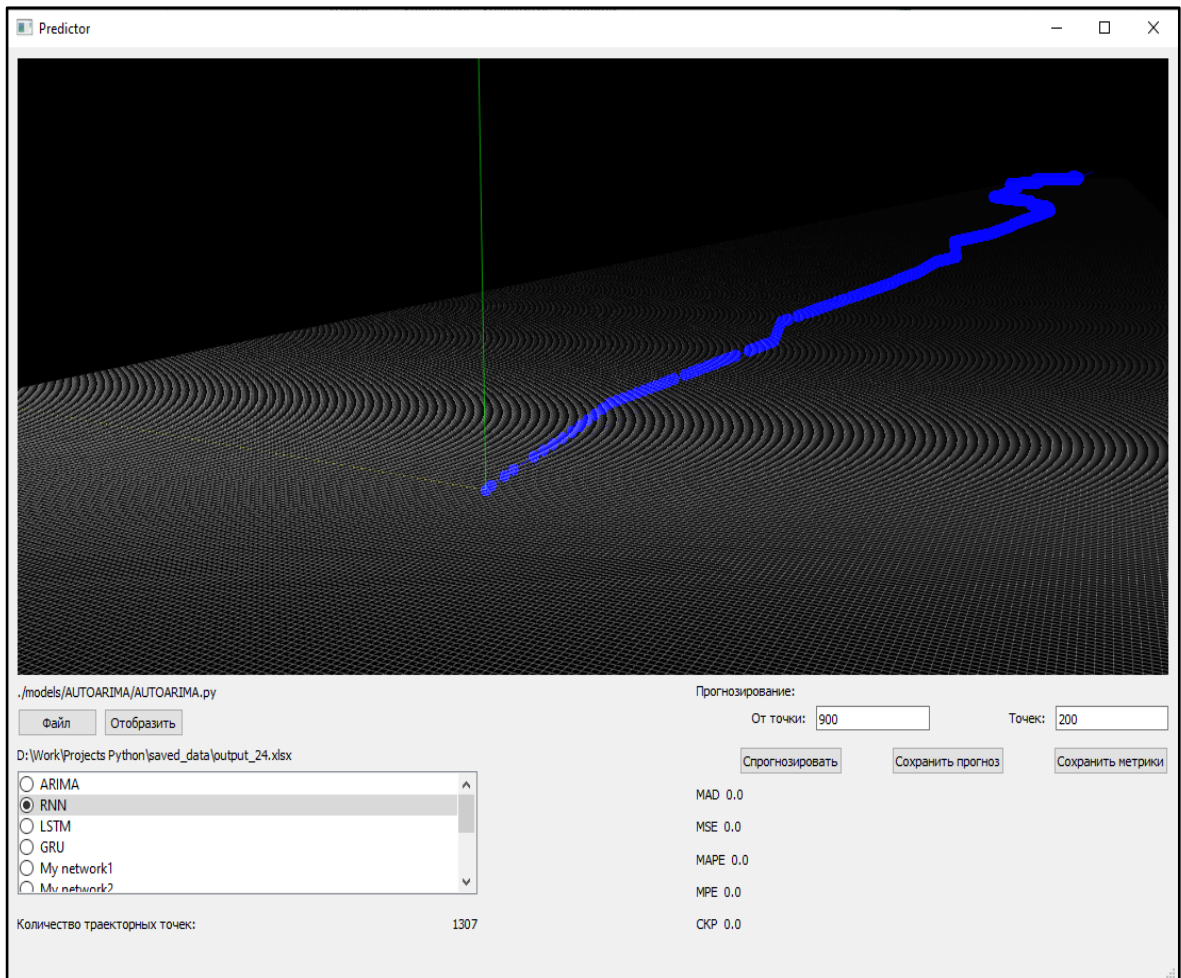


Рисунок 9 – Интерфейс приложения

4.2. Модели прогнозирования траектории

Модель LSTM

В листинге 1 приведен код инициализации модели LSTM [22].

Листинг 1 – Код программы прогнозирования временных рядов при помощи модели LSTM

```
regressorLSTM = Sequential()

regressorLSTM.add(LSTM(50, return_sequences = True,
                       input_shape = (X_train.shape[1],1)))
regressorLSTM.add(LSTM(50, return_sequences = False))
regressorLSTM.add(Dense(25))
regressorLSTM.add(Dense(1))
regressorLSTM.compile(optimizer = 'adam',
                      loss = 'mean_squared_error',
                      metrics = ["accuracy"])
regressorLSTM.fit(X_train, y_train,
```

```

batch_size = 1,
epochs = epochs_number)

```

Данная нейронная сеть имеет два полносвязных слоя по 25 и 1 нейронов, 2 скрытых слоя LSTM по 50 нейронов в каждом с функцией активации из гиперболического тангенса. LSTM слой имеет по умолчанию функцию активации гиперболический тангенс. В качестве оптимизатора используется Adam.

Модель RNN

В листинге 2 приведен код инициализации модели RNN [23].

Листинг 2 – Код программы прогнозирования временных рядов при помощи модели RNN

```

regressor = Sequential()
regressor.add(SimpleRNN(units = 50,
                        activation = "tanh",
                        return_sequences = True,
                        input_shape = (X_train.shape[1],1)))
regressor.add(Dropout(0.2))
regressor.add(SimpleRNN(units = 50,
                        activation = "tanh",
                        return_sequences = True))
regressor.add(SimpleRNN(units = 50,
                        activation = "tanh",
                        return_sequences = True))
regressor.add(SimpleRNN(units = 50))
regressor.add(Dense(units = 1,activation='sigmoid'))
lr_schedule = tf.keras.optimizers.schedules.ExponentialDecay(
    initial_learning_rate=0.01,
    decay_steps=10000,
    decay_rate=0.9)

regressor.compile(optimizer = SGD(learning_rate=lr_schedule,
                                momentum=0.9,
                                nesterov=True),
                  loss = "mean_squared_error")
regressor.fit(X_train, y_train, epochs = epochs_number, batch_size = 2)

```

Данная нейронная сеть имеет один слой исключения и 4 скрытых слоя RNN по 50 нейронов в каждом с функцией активации из гиперболического тангенса. Выходной слой имеет сигмоидальную функцию активации. В качестве оптимизатора используется обычный стохастический градиентный спуск (SGD).

Модель GRU

В листинге 3 приведен код реализации программы прогнозирования временных рядов при помощи модели GRU [18, 24].

Листинг 3 – Код программы прогнозирования временных рядов при помощи модели GRU

```
regressorGRU = Sequential()
regressorGRU.add(GRU(units=50,
                    return_sequences=True,
                    input_shape=(X_train.shape[1],1),
                    activation='tanh'))
regressorGRU.add(Dropout(0.2))
regressorGRU.add(GRU(units=50,
                    return_sequences=True,
                    activation='tanh'))
regressorGRU.add(GRU(units=50,
                    return_sequences=True,
                    activation='tanh'))
regressorGRU.add(GRU(units=50,
                    activation='tanh'))
regressorGRU.add(Dense(units=1,
                    activation='relu'))
regressorGRU.compile(optimizer=SGD(learning_rate=0.01,
                                   decay=1e-7,
                                   momentum=0.9,
                                   nesterov=False),
                    loss='mean_squared_error')
regressorGRU.fit(X_train,y_train,epochs=epochs_number,batch_size=1)
```

Данная нейронная сеть имеет один слой исключения и 4 скрытых слоя по 50 нейронов в каждом с функцией активации из гиперболического тангенса. Выходной слой имеет сигмоидальную функцию активации. В качестве оптимизатора используется обычный стохастический градиентный спуск (SGD – Stochastic Gradient Descent).

4.3. Тестирование

Тестирование работы программы генерации синтетической РЛИ

Было проведено тестирование программы симуляции движения воздушного судна.

Были заданы следующие параметры симуляции:

- T = 14400;
- delta_t= 10;

```
- X_start= 0;
- Y_start= 0;
- max_speed = 250;
- max_pos_boost = 6.5;
- max_neg_boost = 3.5;
- min_speed = 140;
- initial_speed = 125;
- initial_angle = 1;
- max_angle_change = 20;
- max_angle = 20;
- prob_boost_change = 0.1;
- prob_angle_change = 0.05;
- prob_timeshift = 0.005;
- prob_missing = 0.05;
- prob_dubbling = 0.005;
- prob_inaccuracy = 1;
- inaccuracy_wide_x = 10;
- inaccuracy_wide_y = 10;
- max_height_speed = 50;
- middle_height = 7500;
- Z_start = 0;
- count_of_traces = 5.
```

В листинге 4 представлено содержимое конфигурационного файла с данными для имитации.

Листинг 4 – Содержимое конфигурационного файла

```
[DataGenerator]
starting_val = 5
T = 14400
delta_t = 10
start_hour = 5
start_minutes = 32
start_second = 47
```

```
X_start = 0
Y_start = 0
max_pos_boost = 6.5
max_neg_boost = 3.5
initial_boost = 0
min_speed = 140
max_speed = 250
initial_speed = 125
initial_angle = 1
max_angle_change = 20
max_angle = 20
prob_boost_change = 0.1
prob_angle_change = 0.05
prob_timeshift = 0.005
prob_missing = 0.10
prob_dubbling = 0.005
prob_inaccuracy = 1
inaccuracy_wide_x = 10
inaccuracy_wide_y = 10
max_height_speed=50
middle_height=7500
Z_start=0
count_of_traces = 5
```

Результат работы программы приведен на рисунке 10.

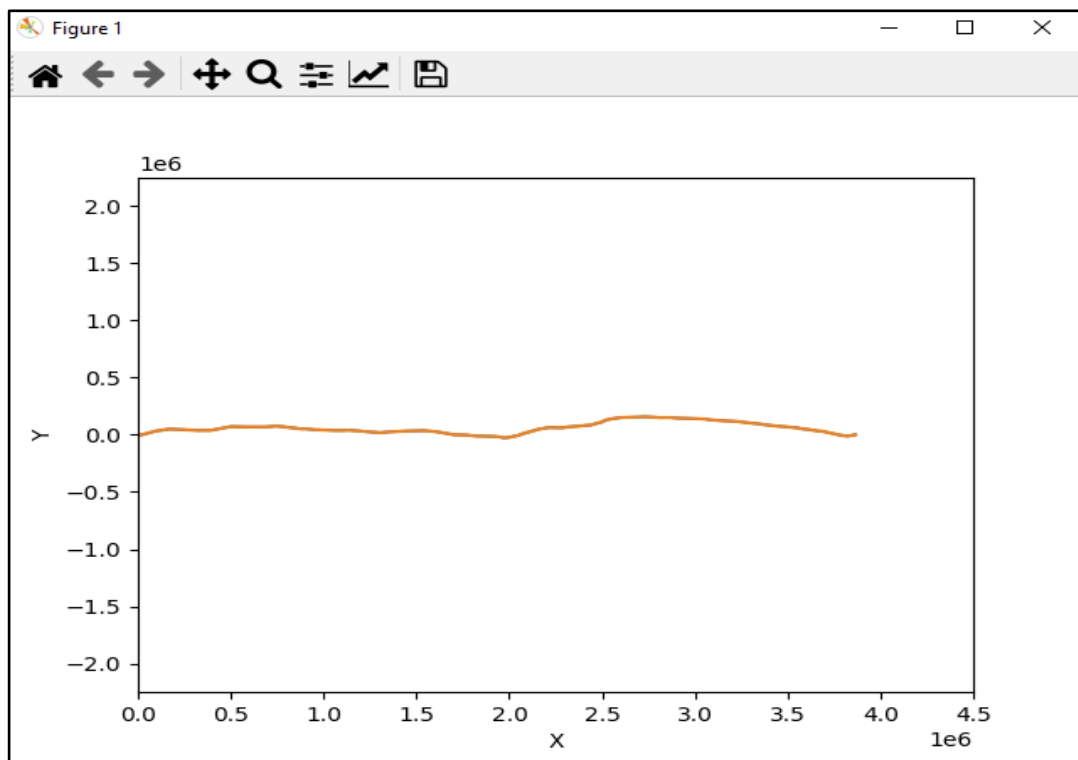


Рисунок 10 – Результат работы программы генерации траекторных меток

Методом визуальной оценки результата тестирования подтверждаем, что полученная траектория в достаточной степени соответствует реальной

траектории движения воздушного судна и может быть использована для формирования обучающей выборки для нейронной сети приложения.

Тестирование работы интерфейса приложения

Ниже приведен результат работы программы отвечающей за инициализацию и работу интерфейса. На рисунках изображены вызов проводника при нажатии кнопки «Файл» (рисунок 11) и отображение траектории при нажатии кнопки «Отобразить» (рисунок 12).

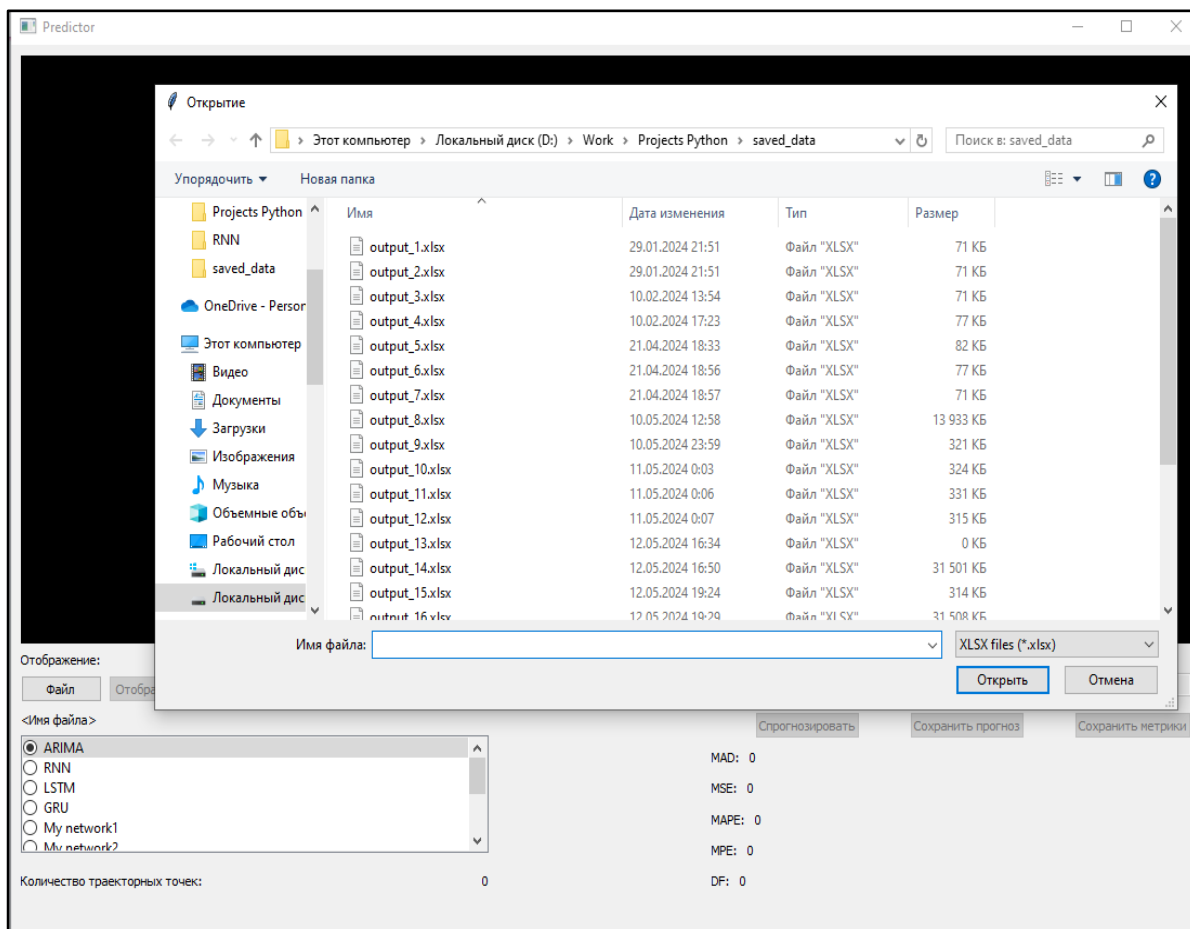


Рисунок 11 – Результат вызова функции выбора файла с РЛИ через кнопку «Файл»

Программа генерации РЛИ сама определяет, какие файлы с РЛИ уже есть в директории и сохраняет новые файлы в папке с новым именем.

После выбора файла на месте поля «имя файла» отображается путь к нему, а рядом с надписью «Количество траекторных точек» отображается текущее число траекторных меток в файле для ориентации пользователя.

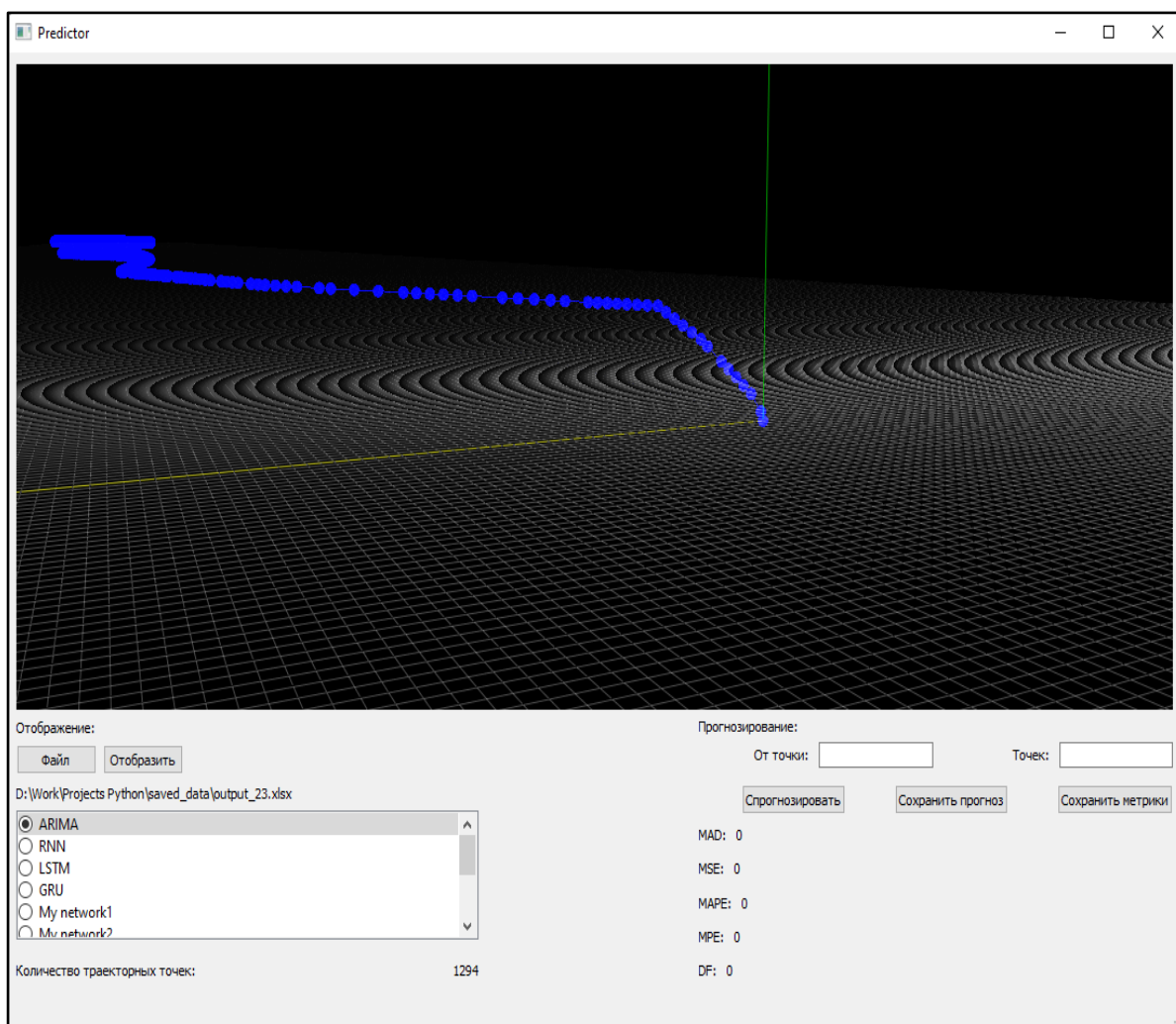


Рисунок 12 – Результат вызова функции отображение траекторных точек на экране через кнопку «Отообразить»

После нажатия кнопки «Отообразить» становятся доступными поля ввода номера начальной точки, с которого необходимо начать прогноз и поле для количества точек, которые нужно спрогнозировать. По умолчанию, если числа не будут введены в эти поля, прогноз будет вестись с последней точки траектории и на одну отметку вперед, а методом прогнозирования будет выбрана ARIMA.

Тестирование функции прогнозирования приложения

Для выбора метода прогнозирования траектории отображается список с доступными моделями. Этот список формируется автоматически из

конфигурационного файла, где записаны пути к файлам для запуска моделей.

На рисунке 13 представлен результат прогнозирования траектории при помощи модели AUTOARIMA.

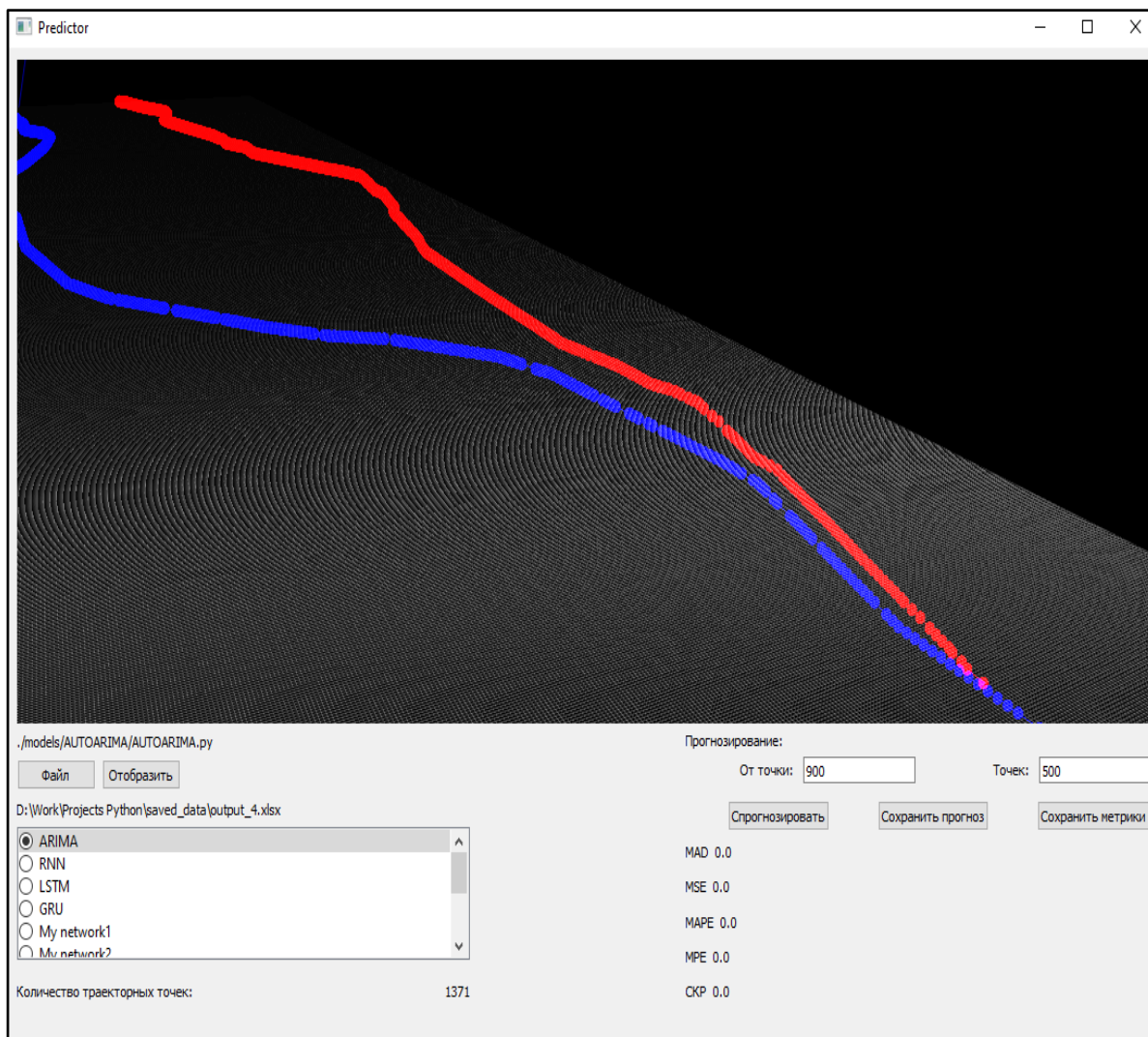


Рисунок 13 – Результат прогнозирования временного ряда с траекторными точками моделью AUTOARIMA

Траектория движения реального воздушного судна отображается синим цветом, а спрогнозированная моделью траектория помечается красным цветом.

На рисунке 14 представлен результат прогнозирования траектории при помощи модели RNN.

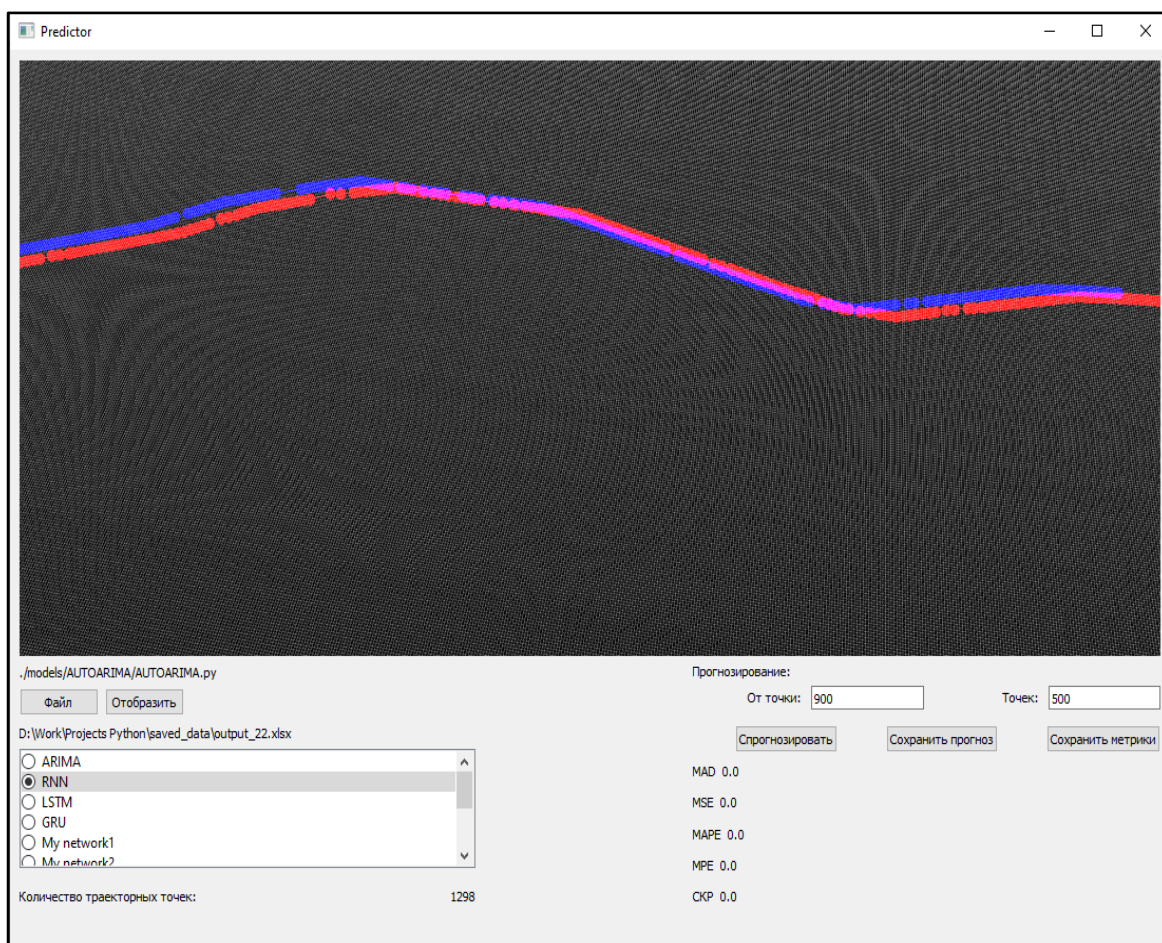


Рисунок 14 – Результат прогнозирования временного ряда с траекторными точками моделью RNN

После завершения прогнозирования, программа вычисляет и отображает регрессионные метрики, полученные при сравнении прогнозируемого и реального временных рядов: среднее абсолютное отклонение (MAD – Mean Absolute Deviation), среднеквадратичная ошибка (MSE – Mean Squared Error), средняя абсолютная процентная ошибка (MAPE – Mean Absolute Percentage Error), средняя процентная ошибка (MPE – Mean Percentage Error) и корень из среднеквадратичной ошибки (RMSE – Root Mean Squared Error).

По полученным предварительным прогнозам уже можно сделать вывод, что траектории, спрогнозированные нейронными сетями LSTM и RNN, ближе к реальной траектории и имеют меньшее отклонение, чем траектория, вычисленная моделью ARIMA.

5. ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ

Исследования эффективности рассматриваемых методов уже были проведены ранее, при анализе других временных рядов [25].

Модели были обучены на 20 эпохах, в качестве обучающей выборки использовалась сгенерированная РЛИ с 500 траекторий движения ВС, что соответствует среднему трафику пассажирского судна за полгода. В качестве метрики точности используется MAPE. Расчет MAPE представлен в формуле (8):

$$MAPE = \frac{1}{n} \cdot \sum \frac{|Y_i - \hat{Y}_i|}{Y_i}, \quad (8)$$

где Y_i – фактическое значение;

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

n – количество периодов.

На рисунке 15 изображена гистограмма с результатами.

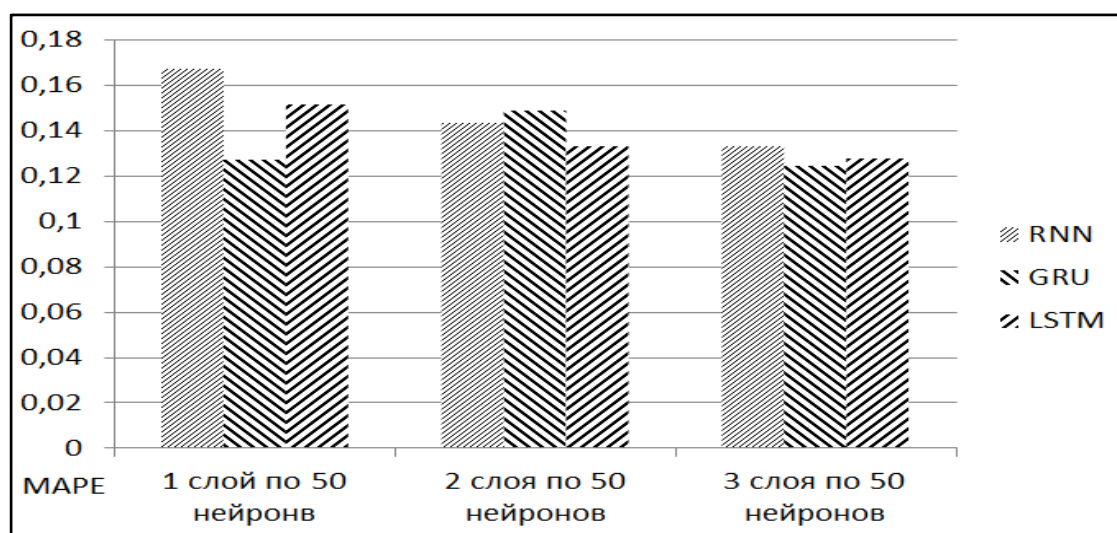


Рисунок 15 – Результаты вычислительных экспериментов

По полученным результатам можно сделать вывод, что GRU эффективнее LSTM, а тот эффективнее RNN. Также можно сделать вывод, что точность увеличивается с прибавлением слоев нейронов к моделям, однако при чрезмерно большом количестве слоев точность может наоборот снизиться. Вероятнее всего, при большом количестве нейронов сеть переобучается и не может выдавать точный прогноз.

ЗАКЛЮЧЕНИЕ

В рамках данной выпускной квалификационной работы было разработано приложение для автоматизированной обработки радиолокационной информации и прогнозирования траектории движения воздушного судна при помощи нейронных сетей.

Для генерации синтетических данных была разработана утилита, имитирующая полет реального воздушного судна и процесс записи данных о положении воздушного судна радиолокационной системой.

В рамках данной выпускной квалификационной работы были выполнены следующие задачи:

- 1) разработана программа генерации синтетической радиолокационной информации по данным из симуляции;
- 2) спроектирован дизайн пользовательского интерфейса приложения для прогнозирования траектории полета воздушного судна;
- 3) разработано приложение для прогнозирования траектории полета воздушного судна, в части управления через пользовательский интерфейс;
- 4) разработаны нейронные модели для прогнозирования траектории движения воздушного судна по данным об уже пройденной траектории;
- 5) обучены, протестированы и отлажены нейросетевые модели;
- 6) встроены нейросетевые модели в приложение для прогнозирования;
- 7) протестировано и отлажено приложение для прогнозирования траектории полета воздушного судна в части точности выдаваемого прогноза;
- 8) проведены вычислительные эксперименты и проведены сравнения с другими методами прогнозирования.

Помимо практической пользы для прогнозирования траекторий, созданная система позволяет оценить эффективность различных методов и моделей прогнозирования траектории движения.

ЛИТЕРАТУРА

1. Кондратьева Т. Н. Прогнозирование тенденции финансовых временных рядов с помощью нейронной сети LSTM. // Вестник евразийской науки, 2017. – 61 с.
2. Hamilton J. D. A new approach to the economic analysis of nonstationary time series and the business cycle. // *Econometrica: Journal of the econometric society*, 1989. – С. 357–384.
3. Голубков А. В., Гаврилова М. П. Применение искусственных нейронных сетей в профилактической и клинической медицине (научный обзор). // *Профилактическая и клиническая медицина*, 2020. – С. 30–39.
4. Митрохин В. Е., Башков И. Н. Прогнозирование мобильного трафика из необработанных данных с использованием сетей LSTM. // *Радиотехника, электроника и связь*, 2019. – С. 117–123.
5. Аверкин А. Н., Ярушев С. А. Нейронные сети в семантическом анализе, 2016. – С. 319–322.
6. Сазонов А. Е., Дерябин В. В. Прогнозирование траектории движения судна при помощи нейронной сети. // *Вестник государственного университета морского и речного флота им. адмирала СО Макарова*, 2013. – С. 6–13.
7. Бакулев П.А. Радиолокационные системы. Учебник для вузов. // М.: Радиотехника, 2007. – 376 с.
8. Bowen E. G. Radar days. // CRC Press, 1998. – 220 с.
9. Жутяева Т.С., Зайцев М.Ф. Проектирование цифровых устройств обработки сигналов в обзорных РЛС. Учебное пособие по курсу «Радиолокационные методы исследования окружающей среды». // М.: МЭИ, 1998. – 78 с.
10. Черняк В. С. Многопозиционная радиолокация. // М.: Радио и связь, 1993. – 416 с.
11. Петров А. В., Яковлев А. А. Анализ и синтез радиотехнических комплексов. // М.: Радио и связь, 1984. – 248 с.

12. Кузьмин С. З. Основы теории цифровой обработки радиолокационной информации. // М.: Сов.радио, 1974. – 432 с.
13. Кремер И. Я., Нахмансон Г. С. Оптимальная обработка сигналов при когерентном многопозиционном приеме на фоне внутренних и внешних шумов. // Радиотехника и электроника, 1979. – С. 2478–2487.
14. Теоретические основы радиолокации. // М.: Сов.радио, 1978. – 607 с.
15. Ярлыков М. С. Применение Марковской теории нелинейной фильтрации в радиотехнике. // М.: Сов.радио, 1980. – 360 с.
16. Al-Chalabi H., Al-Douri Y. K., Lundberg J. Time Series Forecasting using ARIMA Model: A Case Study of Mining Face Drilling Rig. // 12th International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP 2018), Athens, Greece, November 18–22, 2018. – International Academy, Research and Industry Association (IARIA), 2018. – С. 1–3.
17. Татузов А.Л. Нейронные сети в задачах радиолокации. // М.: Радиотехника, 2009. – 432 с.
18. Yang Z., Pang C., Zeng X. Trajectory Forecasting Using Graph Convolutional Neural Networks Based on Prior Awareness and Information Fusion. // ISPRS International Journal of Geo-Information, 2023. – 77 с.
19. Payeur P., Le-Huy H., Gosselin C. M. Trajectory prediction for moving objects using artificial neural networks. // IEEE Transactions on Industrial Electronics, 1995. – С. 147–158.
20. Gao N. et al. Generative adversarial networks for spatio-temporal data: A survey. // ACM Transactions on Intelligent Systems and Technology (TIST), 2022. – С. 1–25.
21. Fernando T. et al. Gd-gan: Generative adversarial networks for trajectory prediction and group detection in crowds. // Springer International Publishing, 2019. – С. 314–330.

22. Siami-Namini S., Tavakoli N., Namin A. S. The performance of LSTM and BiLSTM in forecasting time series. // 2019 International conference on big data (Big Data), 2019. – C. 3285–3292.
23. Tokgöz A., Ünal G. A RNN based time series approach for forecasting turkish electricity load. // 2018 26th Signal processing and communications applications conference (SIU), 2018. – C. 1–4.
24. Wu L. A short-term load forecasting method based on GRU-CNN hybrid neural network model. // Mathematical problems in engineering, 2020. – C. 1–10.
25. Parizad A., Hatziadoniu C. Deep learning algorithms and parallel distributed computing techniques for high-resolution load forecasting applying hyperparameter optimization. // Systems Journal, 2021. – C. 3758–3769.

ПРИЛОЖЕНИЕ. Алгоритм программы генерации РЛИ

Ниже приведена блок-схема программы генерации траекторных меток (рисунок 1).

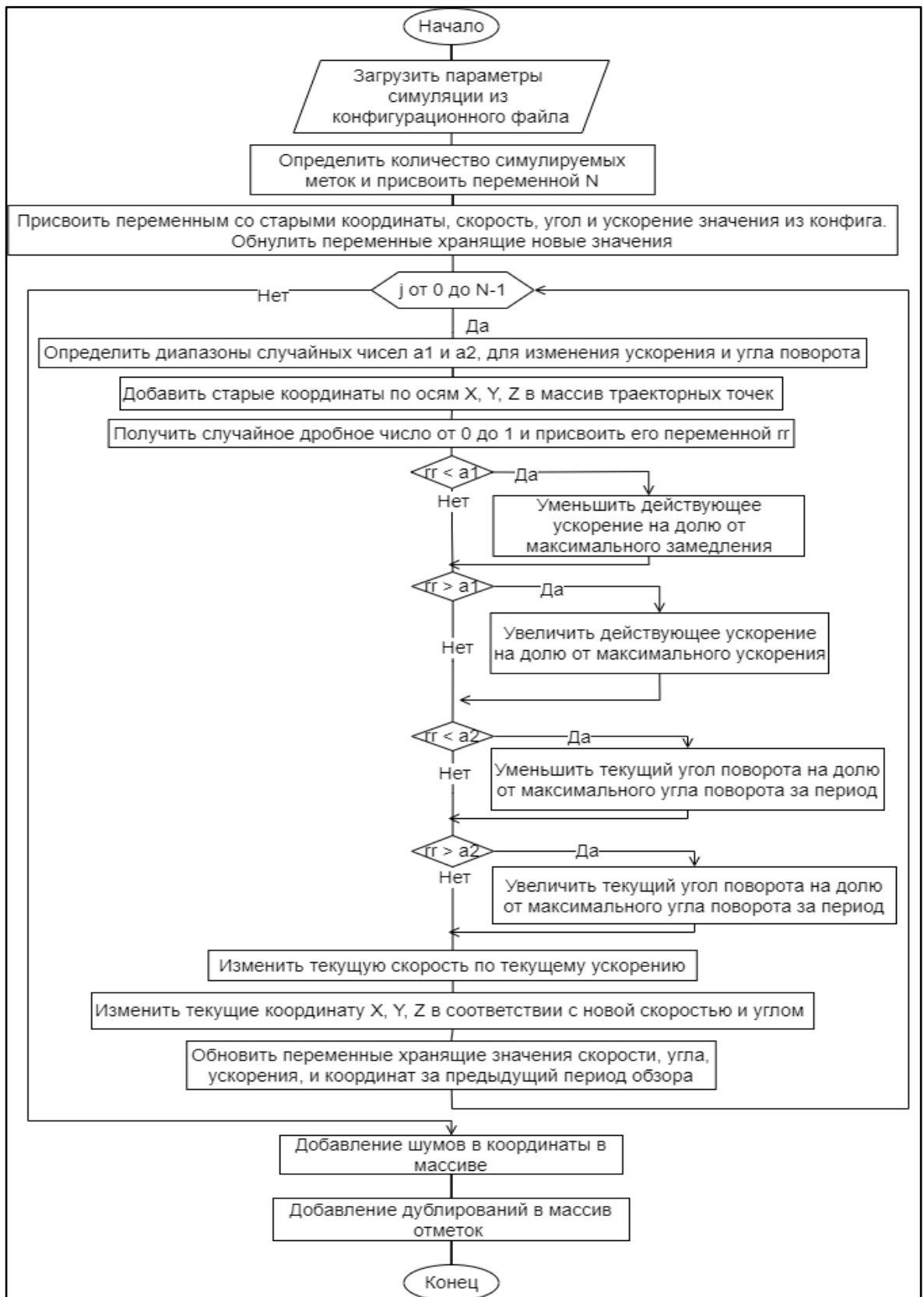


Рисунок 1 – Алгоритм программы генерации РЛИ