# 航行模拟器文档

## 一、需求分析

本系统旨在通过分析航行器的速度、方向角偏差、质量数量级对比和天气条件，预测航行器之间是否会发生碰撞，系统使用先进的深度学习模型，结合实时数据，提供准确的碰撞预测结果。该系统旨在面向广泛的用户群体，包括商业、政府和教育机构等，使他们都可以受益，以提高整体海上航行的安全性和效率，下对这些机构的具体需求进行分析

* 航运公司和船舶运营商：

这些用户需要确保他们的船只安全航行，避免碰撞事故，以减少损失和责任。系统可以为他们提供实时的碰撞预测，帮助他们做出及时的航行调整。

* 海上搜救组织：

这些组织需要预测和识别可能发生碰撞的区域，以便快速响应和救援。系统可以提供碰撞预测，帮助他们提前部署资源。

* 保险公司：

保险公司需要评估海上航行的风险，以便为船舶保险提供准确的保费。系统可以提供碰撞预测数据，帮助他们评估风险和定价。

* 船舶制造商和设计公司：

他们需要了解船舶在实际航行中的表现，以改进设计和提高安全性。系统可以提供碰撞预测和分析报告，帮助他们优化船舶设计。

* 海上物流公司：

这些公司需要确保货物安全、及时地到达目的地。系统可以提供碰撞预测，帮助他们规划航线和调度船只。

**二、工作流程**

该模型描述了航行碰撞的概率预测，预测利用物理方法合成了数据，并利用随机森林方法进行模型训练。包括多边形点阵的变换（将变速曲线运动规划为抛物线求最近点）,并采用蒙特卡洛估算重叠⾯积计算损失程度。

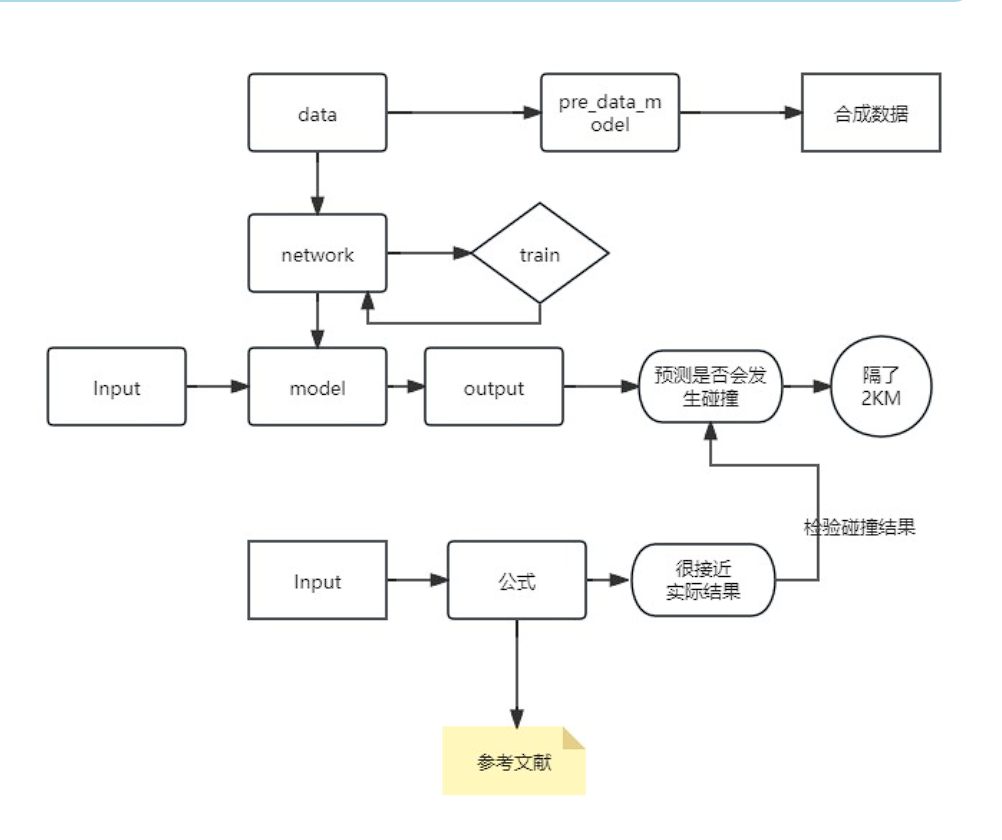


图2.1 工作流程图

## 三、功能说明

主要功能包括以下三个方面

**数据输入：**输入航行器的速度、方向角偏差、质量数量级对比和天气条件。

**碰撞预测：**基于输入数据，使用深度学习模型预测是否会发生碰撞。

**结果展示：**展示碰撞预测结果，并提供详细的分析报告。

### 1、用户输入参数

* **船体长度：**

影响船舶的总体积和在水中的阻力，也关系到船舶的操控性和转向半径。

* **船体宽度**：

与船体长度一起决定船舶的横截面积，影响船舶的稳定性和操控性。

* **船体高度**：

影响船舶的浮力和稳定性，特别是在风浪中的抗倾覆能力。

* **漏水速度**：

在发生碰撞后，如果船体受损，漏水速度将影响船舶的稳定性和沉没速度。

* **逃离速度**：

这些参数代表不同紧急情况下的逃生速度，例如在火灾、碰撞或其他紧急情况下，人员撤离的速度。

* **最大感知距离**：

指船舶的雷达或其他传感器能探测到其他物体的最远距离。

* **感知误差限**：

指传感器在探测过程中可能出现的最大误差范围。

* **人体反映时间**：

指人在接收到紧急情况信号后，做出反应所需的平均时间。

* **船体速度**：

指船舶当前的速度，这对于预测船舶的轨迹和可能的碰撞点至关重要。

* **对方物体速度**：

指接近船舶的其他物体（如另一艘船或障碍物）的速度。

* **距离初始值**：

指两艘船舶或船舶与其他物体之间的初始距离。

* **船只偏向角**（**0°径直碰撞）**：

指船舶与其他物体的相对角度为0度，即直接面对对方，这种情况下碰撞的风险最高。

* **时间误差限度：**

指在预测碰撞时，允许的时间误差范围。

* **船舶1初始位置**：

用户可以输入船舶1的初始X坐标和Y坐标。

* **船舶1速度**：

用户可以输入船舶1在X轴和Y轴方向上的速度。

* **船舶1尺寸**：

用户可以输入船舶1的长度和宽度。

* **船舶2初始位置**：

用户可以输入船舶2的初始X坐标和Y坐标。

* **船舶2速度**：

用户可以输入船舶2在X轴和Y轴方向上的速度。

* **船舶2尺寸**：

用户可以输入船舶2的长度和宽度。

* **转向距离:**

用户可以输入船舶在一定距离内转向的距离。

* **转向方向:**

用户可以选择船舶的转向方向，包括左转和右转。

### 2 、船舶运动模拟

* **移动**：船舶根据输入的速度和时间步长进行移动。
* **转向**：当两艘船舶之间的距离小于设定的转向距离时，船舶会根据设定的转向方向进行转向。

### 3、碰撞检测

* **蒙特卡罗算法**：应用使用 检测两艘船舶之间的碰撞。如果检测到碰撞，应用会在图表中标记碰撞点，并在文本中显示“Collision detected!”。

### 4、存档和加载

* **存档**：用户可以点击“Save”按钮，将当前的船舶状态（包括位置、速度、长度、宽度、转向距离和转向方向）保存到 ship\_state.json 文件中。
* **加载**：用户可以点击“Load”按钮，从 ship\_state.json 文件中加载之前保存的船舶状态，恢复仿真。

## 四、使用说明

通过左侧“导航栏”选择所需功能，进入相应页面,下对各个页面的具体使用进行说明。

### 1、主页

主页展示了本系统简介、主要功能、集中功能展示与开发区域。

****

图4.1 主页示意图

### 2、注册页面

用户可以使用qq邮箱进行注册



图4.2 注册页面示意图

### 3、登录页面

注册后，用户可以通过qq邮箱登录账号



图4.3 登录页面示意图

### 4、航行条件分析与碰撞模拟

**1）输入参数：**

左侧选择天气条件和船舶类型，并选择航行速度和船只数量。

中间部分输入最大感知距离、感知误差限、人体反映时间、船体速度、对方物体速度、距离初始值、样本数量、船只偏向角（0°径直碰撞）、时间误差限度等参数，参数确认后点击“确认输入”，参数确认后会显示“输入参数已确认”。

****

图4.4 航行条件分析与碰撞模拟参数输入示意图

**2）输出预测结果：**

点击“输出预测结果”后，会输出预测的碰撞概率和DeepSeek分析结果（分析输入参数和预测结果的原因）。

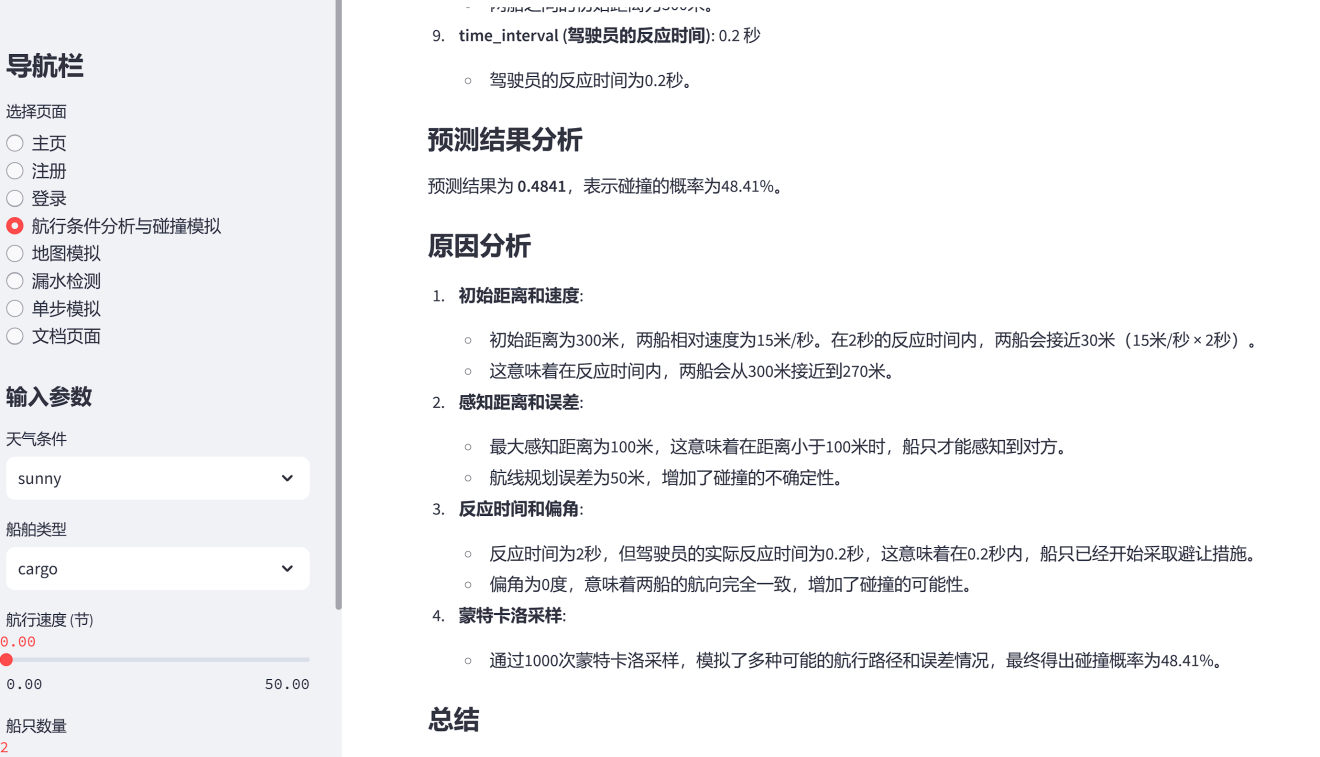
****

图4.5 航行条件分析与碰撞模拟预测结果示意图

### 5、地图模拟

根据“航行条件分析与碰撞模拟”板块输入的参数，进行地图动态模拟。使用不同颜色显示海岸线、岛礁、浅水区和船只轨迹。

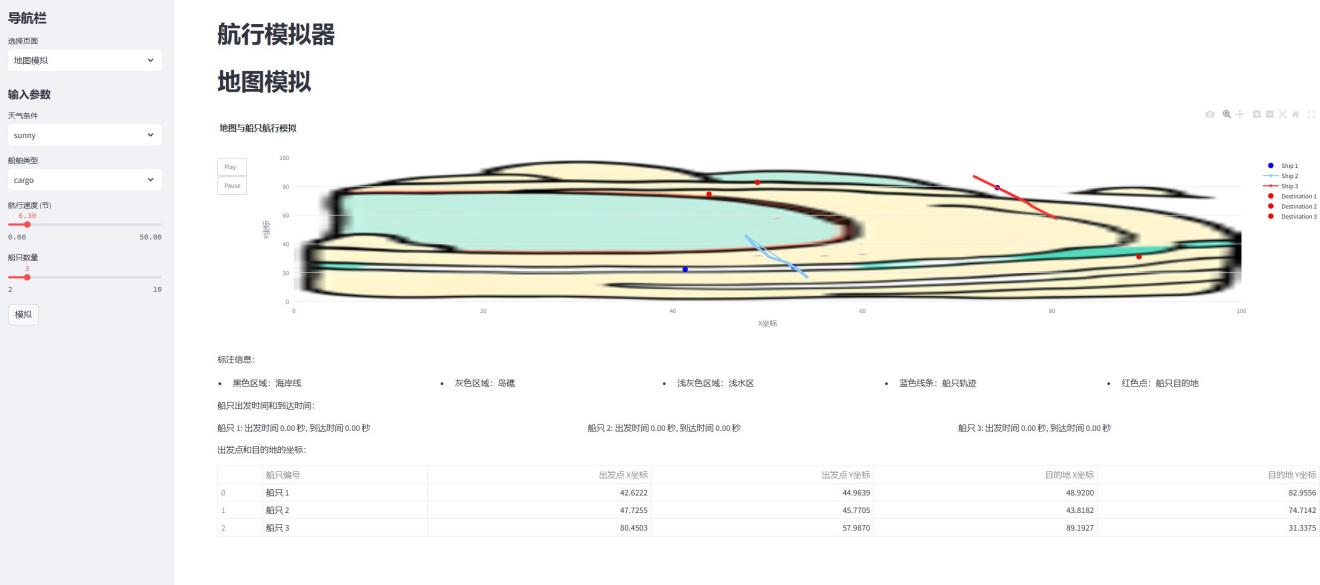


图4.6 地图模拟页面示意图

### 6、漏水检测

**1）输入参数：**

进行船体漏水模型可视化，输入船体参数（船体长度、船体宽度、船体高度）、漏水点参数（漏水速度）和逃离速度（逃离速度1、逃离速度2、逃离速度3）。

****

图4.7 漏水检测参数输入示意图

**2）输出预测结果：**

点击“更新模拟”后，会输出模拟结果。

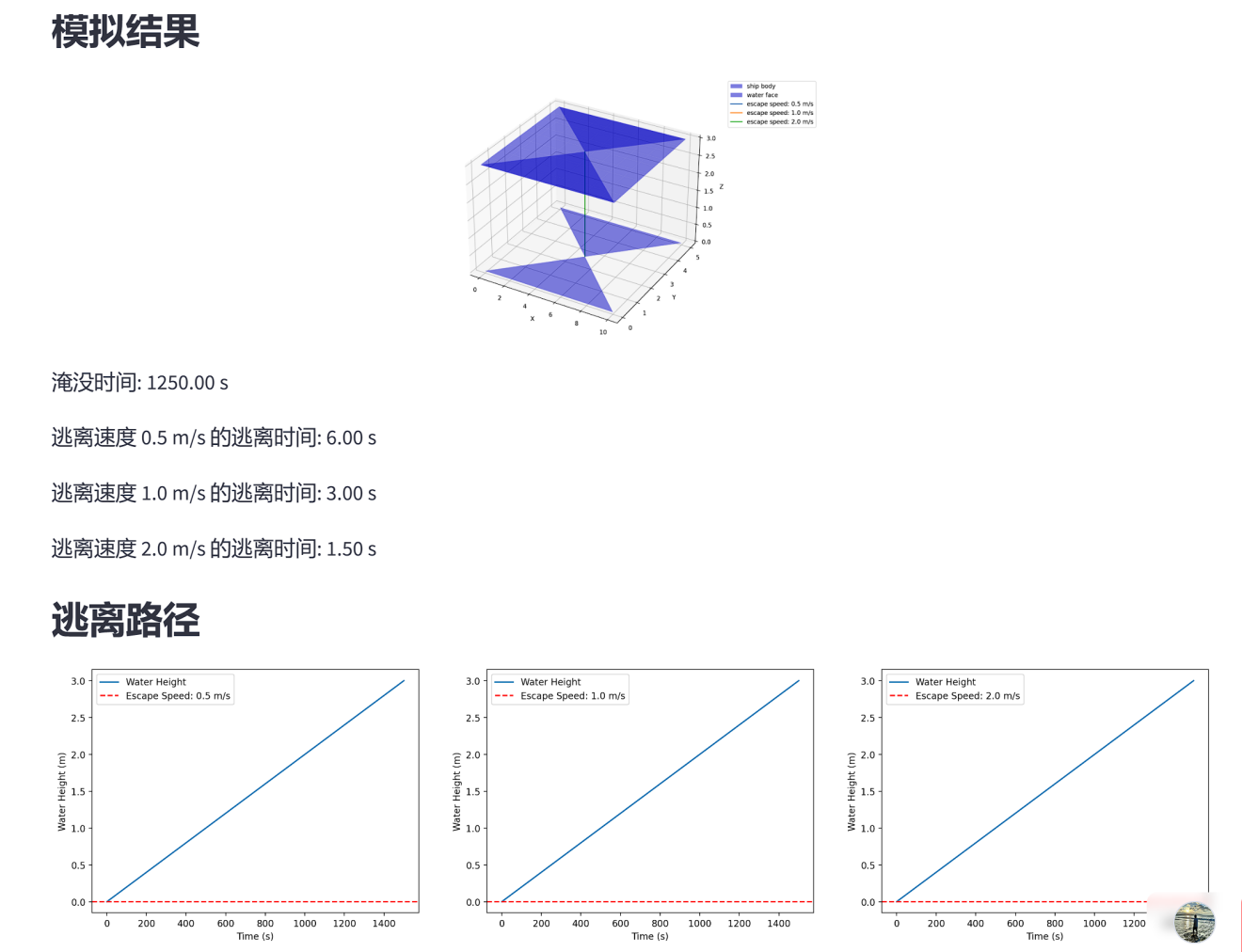
****

图4.7 漏水检测预测结果示意图

### 7、单步模拟

**1 ）输入参数**

1. 在侧边栏中输入船舶1和船舶2的初始位置、速度、长度和宽度。
2. 输入转向距离和转向方向。

**2 ）仿真操作**

1. 点击“Step”按钮，逐步推进仿真，观察船舶的运动轨迹和碰撞情况。
2. 点击“Clear”按钮，恢复到初始状态，重新开始仿真。

**3 ）存档和加载**

1. 点击“Save”按钮，将当前的船舶状态保存到 ship\_state.json 文件中。
2. 点击“Load”按钮，从 ship\_state.json 文件中加载之前保存的船舶状态，恢复仿真。

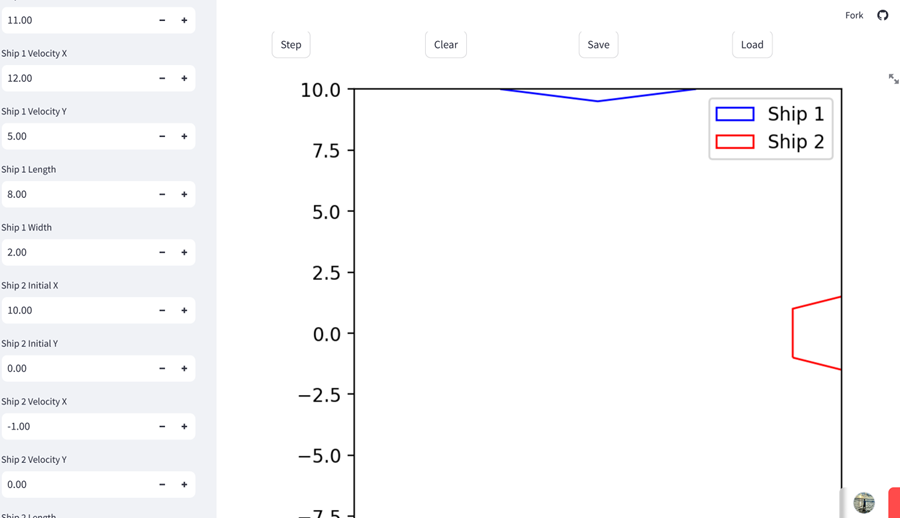


图4.8 单步模拟页面示意图

## 五、环境配置

整个系统基于python 3.12运行

|  |  |
| --- | --- |
| 序号 | 库名称 |
| Streamlit | 1 |
| Numpy | 2 |
| pandas | 3 |
| matplotlib | 4 |
| plotly | 5 |
| streamlit-authenticator | 6 |
| mysql-connector-python | 7 |
| pyvista | 8 |
| openai | 9 |

**六、文件结构**

│ .gitignore

│ requirements.txt

│

├─fig

│ coast.png

│ distribution.png

│ island1.png

│ island2.jpg

│ logo.png

│ ship.png

│ workflow.png

│

├─json\_copy

│ ship\_state.json

│

├─readme

│ distribution.png

│ readme.md

│ readme.pdf

│ workflow.png

│

└─src

│ app.py

│

├─database

│ dachuang\_list.sql

│ db\_config.py

│ log\_reg.py

│

├─model

│ │ server\_shell.sh

│ │

│ ├─data

│ │ collision\_data.csv

│ │ deletetorch\_framework.py

│ │ delete\_data.py

│ │ delete\_plot.py

│ │

│ ├─ship\_model

│ │ │ 3D\_moving.py

│ │ │ 3D\_ship.py

│ │ │ collision\_model.py

│ │ │ data\_generate.py

│ │ │ predict.py

│ │ │ train.py

│ │

│ └─temp

│ best\_collision\_model.pkl

│ collision\_data.csv

│ scaler.pkl

│ xgboost\_model.json

│

└─page

page\_box\_leaking.py

page\_deepseek\_api.py

page\_doc.py

page\_enter.py

page\_map\_simulation.py

page\_ship\_collision.py

page\_single\_step.py

图6.1 文件结构展示