同濟大學

TONGJI UNIVERSITY

毕业设计（论文）

课题名称　基于多源数据融合的台风强度和路径预报

副 标 题

学　　院　　软件学院

专　　业　　软件工程

学生姓名　　梁厚

学　　号　　2051840

指导教师　　穆斌

日　　期　　2024 年 6 月 6 日

基于多源数据融合的台风强度和路径预报

装

订

线

摘 要

台风是最危险的自然灾害之一，它们总是发生在西太平洋和西南太平洋，每年都会给太平洋沿岸带来经济和人身安全威胁。因此，相关领域的许多学者致力于寻找一种有效的方法来分析和预测台风路径，以防止灾害。

本研究构建了一个基于机器学习的多源数据融合模型，用于预测台风路径和强度。该模型主要分析太平洋地区的大气变量（如台风中心经度、纬度、气压和风速），并训练出一个高性能的预测模型。在此基础上，通过台风相似度计算模型，将路径和强度预测结果结合起来评估模型性能。

与以往模型相比，本文提出的模型利用了更多样化的台风历史资料，从而显著提高了预测精度。该模型能够同时输出台风的强度和路径，有助于更早掌握台风动向，协助相关部门进行预防工作。由于模型输出涵盖多个维度的数据，本文在传统的 MSE 和 MAE 评估方法基础上，加入了基于多维度数据融合的台风相似度计算指标，进一步比较预测台风与实际历史台风。

关键词：台风强度预测，台风路径预测，机器学习，LSTM

Typhoon Intensity and Track Forecast Based on Multi-Source Data Fusion

装 订 线

ABSTRACT

Typhoons are one of the most dangerous natural disasters, occurring frequently in the Western Pacific and Southwestern Pacific regions. They pose threats to both the economy and personal safety along the Pacific coastlines every year. Therefore, many scholars in relevant fields are dedicated to finding an effective method to analyze and forecast typhoon paths, aiming to prevent disasters.

This study develops a machine learning-based multi-source data fusion model for predicting typhoon paths and intensity. The model primarily analyzes atmospheric variables in the Pacific region, such as the longitude and latitude of the typhoon center, central pressure, and wind speed, to train a high-performance prediction model. Based on this, the outputs for path and intensity are combined using a typhoon similarity calculation model to evaluate the model's performance.

Compared to previous models, the proposed model utilizes more diverse historical typhoon data, significantly improving prediction accuracy. This model can simultaneously output typhoon intensity and path, aiding in earlier detection of typhoon movements and assisting relevant departments in disaster prevention. Since the model's outputs cover multiple dimensions, this study incorporates a typhoon similarity calculation index based on multi-dimensional data fusion in addition to traditional MSE and MAE evaluation methods, facilitating a more comprehensive comparison between predicted typhoons and actual historical typhoons.

Key words： typhoon intensity prediction, typhoon path prediction, machine learning, LSTM

目 录

1 引 言 1
1.1 研究背景及意义 1
1.2 国内外研究现状 1
1.3 本文所作的工作 2
2 相关理论与技术 3
2.1 SHIPS 环境参数 3
2.2 台风预测方法 3
2.2.1 CNN 模型 3
2.2.2 LSTM 模型 6
2.2.3 ConvLSTM 模型 8
2.3 台风路径相似性评估的定量方法 9
2.3.1 相似度计算的台风数据结构和符号定义 9
2.3.2 台风相似度计算步骤 10
2.3.3 相似距离归一化方法 11
3 基于多源数据融合的台风强度和路径预测模型实验 13
3.1 数据集分析以及预处理 13
3.1.1 数据集分析 13
3.1.2 数据集预处理 19
3.2 基于 CNN 和 LSTM 模型的台风强度和路径预测 20
3.2.1 ConvLSTM_CNN 模型训练 20
3.2.2 ConvLSTM_CNN 模型训练实验步骤 24
4 实验结果和讨论 29
4.1 关于路径预测的模型输出性能表现 30
4.2 关于强度预测的模型输出性能表现 32
4.3 基于预测路径和强度综合的台风相似度的讨论 34
4.3.1 台风相似度指标有效性的验证实验 35
5 结论和展望 42
5.1 结论 42
5.2 展望 42
参考文献 44
谢 辞 46

装 订 线

1 引言

1.1 研究背景及意义

台风作为最具破坏性的自然灾害之一，对沿海地区造成重大威胁，特别是在西太平洋和西南太平洋地区。这些极端天气事件带来了广泛的经济损失、生命损失和社会活动的中断。因此，相关领域的许多学者致力于寻找一种有效的方法来分析和预测台风路径和强度，以防止灾害。

传统的台风预报方法主要依赖于单一数据源的气象数据，这可能导致在预测这些风暴的强度和轨迹时出现不准确和局限性。因此，对于能够整合多个数据来源的更先进方法的需求越来越迫切。

本文研究的基于多源数据融合的台风强度和路径预报可以在一定程度上提高台风预报的准确性和可靠性，帮助相关部门更及时地做出灾害应对和减灾措施。其次，通过整合不同来源的数据，可以更全面地了解台风的形成和发展过程，为深入研究气候变化等问题提供重要参考。此外，这种方法还有助于推动机器学习和数据科学在气象领域的应用和发展，促进科学研究和技术创新的进步。

1.2 国内外研究现状

以往的台风路径和强度预报有几种流行的技术类型，包括：① 动力学方法，也称为数值预报方法（Numerical Forecast Method）。它主要使用动力学模型，利用数学方程来模拟台风运动。这些动力学方程往往过于复杂，需要超级计算机的数值方法获得近似解，因此成本较高；② 统计方法。它密切关注天气资源的统计数据而不是物理运动。这种方法的优点是比动态建模方法消耗更少的计算资源，同时可以应用于大多数国家和地区；③ 统计-动力学方法。这类方法结合了统计方法和动态建模方法。统计数据常常用于计算动力力学方程的更重要的初始值。预测结果主要取决于动态建模方法。

近年来，统计方法因其客观性、小计算量的特性受到广泛关注。统计方法的关键问题在于大量历史资料数据的获取以及如何定义或推导出良好的统计模型。

在历史资料数据的获取方面，训练数据常常需要用到各个气象中心提供的历史台风数据，包括二维的台风数据、卫星云图等。常用的公开数据集有，由中国气象局（CMA）维护的二维台风数据集、由欧洲中期天气预报中心（ECMWF）维护的 3D 台风数据集 EAR-5，以及在热带气旋图像强度回归数据集网站上的台风强度卫星图像（TCIR）。CMA 数据集提供了自 1949 年以来 WNP 盆地中每 6 小时热带气旋的位置和强度，提供包括时间、经度、纬度、风速、中心气压的台风数据。EAR-5 数据集提供了自 1979 年以来收集的 14 种全球大气再分析数据。EAR-5 的空间分辨率约为 31 公里，时间分辨率为每小时，数据通常以标准的气象学变量（如温度、湿度、风速等）以及衍生的气象学产品（如降水、云量、地面气压等）的形式提供。TCIR 收集了 2003 年至 2017 年期间全球 1379 个热带气旋的数据集，包括每 3 小时一次的卫星红外图像（IR）、水汽（WV）和被动微波降雨量（PMW）热带气旋图像，水平分辨率为 0.07 度纬度/经度。

统计模型的设计方面，2016 年，Moradi[1]等人提出了一种稀疏的循环神经网络（RNN），

用于自动提取二维台风的非线性特征。此外，他们还使用了动态时间规整（Dynamic Time Warping）来寻找与目标台风类似的路径。为了克服动态时间规整方法需要台风单调性的限制，Alemany[2]等人提出了一个全连接的循环神经网络，即长短期记忆（LSTM），用于构建二维台风的非线性特征。该工作的创新之处在于将台风的中心位置编码成人工网格点系统，以减少误差传播。对于建模台风的三维特征，即表示其整个三维结构的不同压力层的重新分析数据堆叠，提出了许多时空深度学习方法。Liu[3]等人设计了一个卷积神经网络（CNN），利用重新分析数据构建极端天气事件（包括热带气旋）的三维结构。然后，Kim[4]等人利用卷积 LSTM（ConvLSTM）模型执行相同的任务。ConvLSTM 相对于 CNN 的优势在于前者可以捕捉时间和空间之间的相关性。

本研究提出一种深度学习的模型，将 CNN 与 ConvLSTM 相结合，分析传统的预测因子和卫星观测数据，用于预测台风在 +24 小时的最大风速（V_{max}）和经纬度的位置信息。

1.3 本文所作的工作

本研究在现有数值预报、统计学习以及机器学习台风预测方法的基础上，结合台风历史数据和大气海洋数据的时空特性，构建多源数据融合的深度学习模型进行台风强度和路径预测，并且创新性地引入了台风路径相似性评估的定量方法，使用模型预测的强度和路径数据结合真实值进行综合计算，得出台风相似度的具体数值。其中台风强度指的是 +24 小时内的最大风速；路径则为台风中心的经度和纬度。

主要研究内容及流程包括：

（1）针对台风相关的大气数据、卫星云图和最佳路径数据，本文执行了以下优化步骤：首先，进行了数据的综合分析和预处理。在预处理阶段，对图片和文本数据进行了组合，并将数据集划分为训练集、验证集和测试集。此外，按照台风编号对数据进行了切割，并将其按照特定格式储存至代码中。针对预测的关键字段包括“强度”、“经度”和“纬度”，本研究进行了归一化处理。更多数据处理的具体步骤，请参考论文中的“数据分析以及预处理”部分。

（2）采用了多变量融合的深度学习模型来预测台风的强度和位置，并设置了训练轮次对模型进行训练。

（3）将训练好的模型保存，并且运用于测试集之上，和实际的观测值进行对比。根据台风强度预测的结果指标（均绝对方误差）相关计算公式进行台风强度预测结果的解读。

（4）根据训练的模型以及台风强度预测的结果，使用 Yangchen Di[5]等人提出的台风路径相似性评估的定量方法对预测台风和实际台风的相似性进行度量。同时对该方法进行了验证实验。

2 相关理论与技术

2.1 SHIPS 环境参数

SHIPS（Statistical Hurricane Intensity Prediction Scheme）是一个用于预测热带气旋（如飓风和台风）强度变化的统计模型。它结合了统计学方法和物理学原理，通过分析历史数据和当前的环境条件来预测未来热带气旋的强度变化。SHIPS 模型在美国国家飓风中心（NHC）和其他气象机构中广泛应用，用于支持热带气旋预报。

SHIPS 模型依赖于一组关键的环境参数，这些参数对热带气旋的强度变化有显著影响。SHIPS 模型使用的环境变量包括：

（1） 垂直风切变（Vertical Wind Shear）：指的是不同高度上的风速和风向的差异。强风切变通常会阻碍对流气流的组织和发展，从而限制强降水和风暴的形成。

（2）海表温度（Sea Surface Temperature，SST）：海表温度对于热带气旋和其他对流系统的发展至关重要。较高的海表温度通常会提供更多的能量，促进对流活动的增强。

（3） 垂直稳定度指数（Vertical Stability Index，VWS）：它衡量了大气在垂直方向上的稳定性。较低的稳定度通常会促进对流活动的增强。

（4） 湿度（Moisture）：高湿度有助于提供更多的水汽供给，从而支持强降水事件的发生。

（5） 剪切风速（Shear Vectors）：这指的是大气中的风速在空间中的变化情况。较强的剪切通常会阻碍对流系统的垂直发展，影响强降水的形成。

SHIPS 模型通过分析这些环境参数以及它们之间的相互作用，来评估未来强对流天气事件的可能性和强度。尽管 SHIPS 模型有其预测的局限性，包括：模型过于依赖于历史数据，可能无法完全捕捉到异常或极端天气事件以及预测结果对输入参数的准确性敏感，数据质量和观测精度会影响预测效果，但本文认为其在环境变量的选取上仍然具有相当大的借鉴意义，故本文参考了与 SHIPS 模型选取的环境变量作为输入的一部分进行强度和路径的预测。

2.2 台风预测方法

2.2.1 CNN 模型

神经网络（Neural Networks）是一类模仿人脑神经元工作原理的计算模型，通过层层递进的方式处理和学习数据中的复杂模式和特征。神经网络由多个节点（也称为“神经元”或“单元”）组成，这些节点按照一定的层次结构连接在一起。通常，神经网络包含输入层、隐藏层和输出层：

（1） 输入层：接受外界输入的数据，每个节点代表一个输入特征。

（2） 隐藏层：位于输入层和输出层之间，由若干节点组成，负责对输入数据进行特征提取和非线性变换。神经网络的强大能力主要源自隐藏层的深度和复杂性。

（3） 输出层：产生最终的预测结果或分类结果，每个节点对应一个输出值。

神经网络的每个连接都有一个权重，反映了节点间的关系强度。通过训练过程，神经网络不断调整这些权重，以最小化预测误差。训练过程中，常用的优化算法包括梯度下降法和反向

传播算法。传统神经网络的结构如图 2.1 所示。

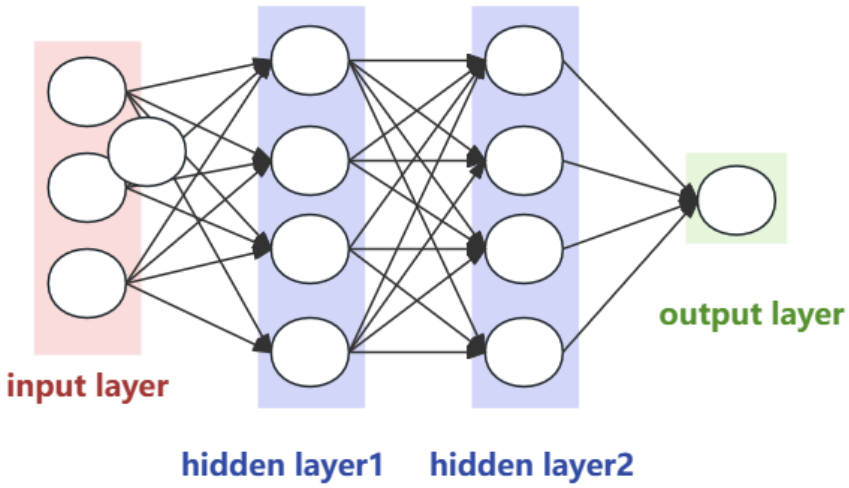

图 2.1 传统神经网络的结构

尽管神经网络在处理非线性问题和复杂模式识别方面具有显著优势，但它们在处理高维数据（如图像、音频和视频）时可能表现不佳。这主要是因为传统神经网络（如前馈神经网络）在处理高维数据时需要大量的参数，导致计算效率低下和过拟合问题。

卷积神经网络（Convolutional Neural Network，CNN）是一种在计算机视觉领域取得了巨大成功的深度学习模型。它们的设计灵感来自于生物学中的视觉系统，旨在模拟人类视觉处理的方式。在过去的几年中，CNN 已经在图像识别、目标检测、图像生成和许多其他领域取得了显著的进展，成为了计算机视觉和深度学习研究的重要组成部分。

CNN 的主要特点和组成部分如下：

（1） 卷积层（Convolutional Layer）：CNN 的核心部分。卷积层将输入图像与卷积核进行卷积操作。然后通过应用激活函数（如 ReLU）来引入非线性。这一步使网络能够学习复杂的特征。

（2） 池化层（Pooling Layer）：池化层用于减少特征图的空间尺寸，并降低模型对位置的敏感性。最常见的池化操作是最大池化，即在特定区域内取最大值作为输出。

（3） 激活函数（Activation Function）：激活函数通常被应用于卷积层之后，用于引入非线性特性。常用的激活函数包括 ReLU（Rectified Linear Unit）、Sigmoid 和 tanh。

（4） 全连接层（Fully Connected Layer）：在 CNN 的最后阶段，全连接层将卷积层和池化层的输出展平，并连接到一个或多个全连接层中。全连接层通常用于将卷积层和池化层提取的特征映射转换成输出类别的概率分布。

（5） 损失函数（Loss Function）：在监督学习任务中，损失函数用于衡量模型预测与真实标签之间的差异。常见的损失函数包括交叉熵损失函数（Cross-Entropy Loss）用于

分类任务。

其中卷积操作是卷积神经网络（CNN）中的核心计算过程，用于提取输入数据中的局部特征。卷积操作通过在输入数据上滑动一个滤波器（或称为卷积核），在每个位置上计算滤波器与输入数据的局部区域的点积，从而生成特征图。

下图展示了一个卷积神经网络中卷积层的具体操作，包括输入（Input Volume）、两个卷积核（Filter W0 和 Filter W1）、偏置（Bias）、以及输出（Output Volume）的计算过程。

输入是一个多维数组（如图像数据），表示为 I，大小为 $H \times W \times C$ $(7 \times 7 \times 3)$，表示有三个通道（例如，RGB 图像中的三个颜色通道）。图中有两个卷积核（Filter W0 和 Filter W1），每个卷积核的大小为 $3 \times 3 \times 3$。这意味着每个卷积核有三个通道，每个通道的大小为 $3 \times 3 \times 3$。卷积核用红色方块表示，每个方块内的数字表示卷积核的权重。

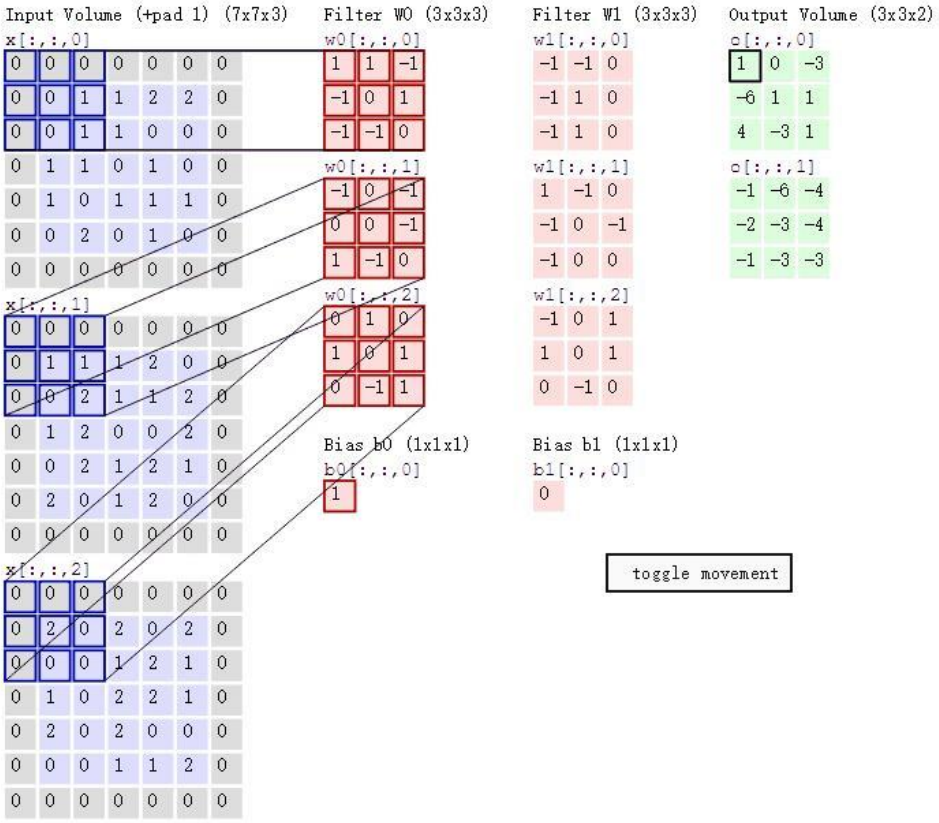

图 2.2 卷积层的具体操作

每个卷积核都有一个对应的偏置项（Bias），Bias b0 对应 Filter W0，Bias b1 对应 Filter W1。偏置项是一个标量，分别为 1 和 0。输出体积（Output Volume）是卷积操作的结果。图中展示了一个大小为$3 \times 3 \times 2$ 的输出体积，表示有两个通道，每个通道的大小为3×3 。输出体积用绿色方块表示，每个方块内的数字表示卷积结果。

CNN 模型通过多个卷积层和池化层交替堆叠，逐渐提取输入图像的高级特征。训练过程中，

通过反向传播算法更新模型参数，使模型能够逐渐学习到更有效的特征表示。在训练充分后，CNN 模型可以对新的未见过的图像进行准确的分类或预测。

2.2.2 LSTM 模型

在传统神经网络中，模型一般不会关注上一时刻的处理对下一刻的影响，也不会关注可用于下一时刻的信息，而只会关注当前时刻的处理。但是递归神经网络（Recurrrrent Neural Networks，RNN）不同，它可以使神经网络处理当前信息时记住上一刻的信息，因此其主要用途是挖掘数据中的时序信息来处理和预测序列数据，目前它被广泛应用于语音识别、机器翻译等方面。

传统的循环神经网络（RNN）的结构如图 2.3 所示。它接收来自两个方面的输入，一是外界的输入X_t，代表当前时刻的输入数据；二是上一个时刻的隐藏层的状态 h_{t-1}，代表网络在前一个时刻的记忆。这两部分输入经过权重矩阵的线性变换后，通过激活函数得到当前时刻隐藏层的状态 h_t，同时也会输出给下一个时刻作为记忆传递。

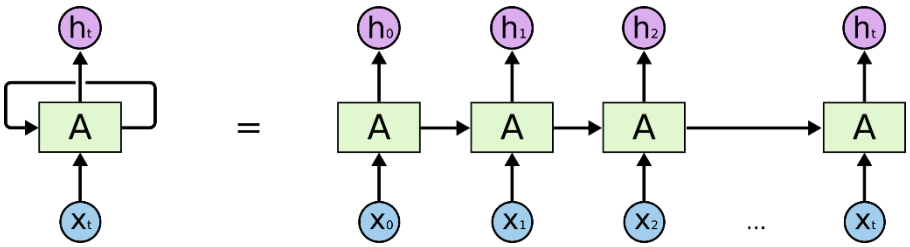

图 2.3 传统的循环神经网络（RNN）结构图

LSTM（Long Short-Term Memory，长短期记忆）是一种循环神经网络（RNN）的变体，专门设计用于处理和预测时间序列数据，如文本、语音、视频等。相比标准的 RNN，LSTM 在处理长期依赖关系时表现更加出色，能够有效地捕捉和记忆长时间间隔内的信息，从而在许多序列建模任务中取得了较好的效果。LSTM 模型的核心是由一系列的门控单元组成，包括遗忘门（forget gate）、输入门（input gate）和输出门（output gate），以及细胞状态（cell state）。这些门控单元通过学习来控制信息的流动，以便长期依赖关系得以保留或遗忘。LSTM 的结构如图 2.4 所示。

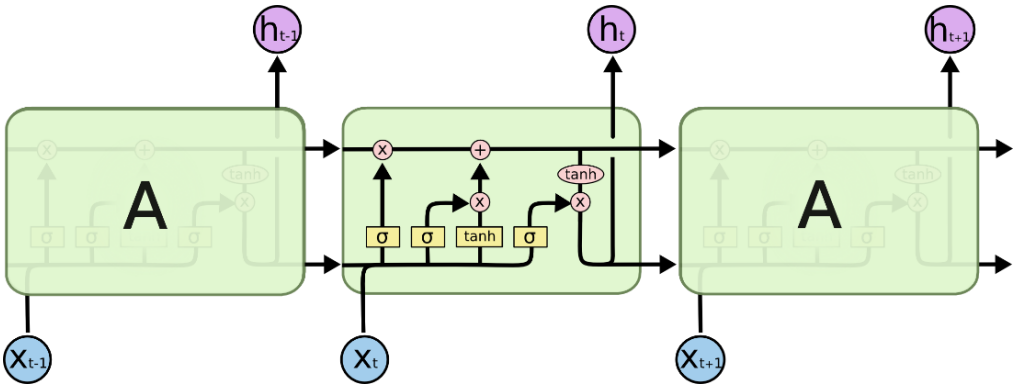

图 2.4 LSTM 结构示意图

其中符号说明见图 2.3。

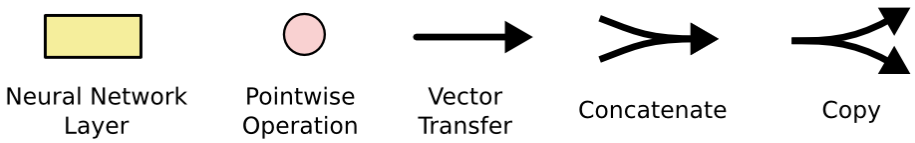

图 2.5 LSTM 结构图中的符号说明

LSTM 神经网络的关键是细胞状态（cell state）对应图 2.4 中最上方的水平线。细胞状态有些类似于传送带，它沿着整个链条直线向下运行，只有一些轻微的线性交互，信息很容易保持不变地流动。LSTM 模型通过被称为“门”的结构删除或者添加信息到细胞状态。门是一种选择性地让信息通过的方法，由一个 sigmoid 神经网络层和一个逐点乘法运算组成。sigmoid 层输出介于 0 和 1 之间的数字，描述了每个组件应该通过的程度。值为零表示“不让任何东西通过”，而值为一表示“让一切通过”。一个 LSTM 有三个这样的门，用于保护和控制细胞状态。

LSTM 的第一步是决定要从单元状态中丢弃哪些信息。这个决定是由一个叫“遗忘门” S 做出的。它着眼于 h_{t-1} 和 x_t，并输出一个介于 0 和 1 之间的向量，它们通过一个 sigmoid 函数计算得出。

$$f_t = \sigma\left(W_f \cdot [\mathrm{h_{t-1}}, \mathrm{x_t}] + \mathrm{b_f}\right) \tag{2.1}$$

下一步是确定要在单元格状态中存储哪些新信息。这分为两部分：①称为“输入门”的 S 型层决定了将更新哪些值 i_t；② tanh 层创建新候选值的向量 $\widetilde{C_t}$，可以添加到状态中。在下一步，LSTM 将结合两者来创建状态更新。

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.2}$$

$$\widetilde{\mathrm{C}}_\mathrm{t} = \tanh\left(\mathrm{W_C} \cdot \left[\mathrm{h_{\{t-1\}}}, \mathrm{x_t}\right] + \mathrm{b_C}\right) \tag{2.3}$$

接着更新上一步的单元格状态 C_{t-1} 进入下一步的单元格状态 C_t。通过将遗忘门的输出与细胞状态相乘，以决定哪些信息需要被遗忘，并将输入门的输出与新的候选值向量相乘，以确定

哪些信息需要更新到细胞状态中。最后根据当前的细胞状态和输出门的输出值来生成隐藏状态（Hidden State）。

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{2.4}$$

$$h_t = o_t \times tanh(C_t) \tag{2.5}$$

这样，LSTM 网络能够有效地处理长序列信息，并且能够选择性地记住和遗忘信息，从而更好地捕捉序列数据中的长期依赖关系。

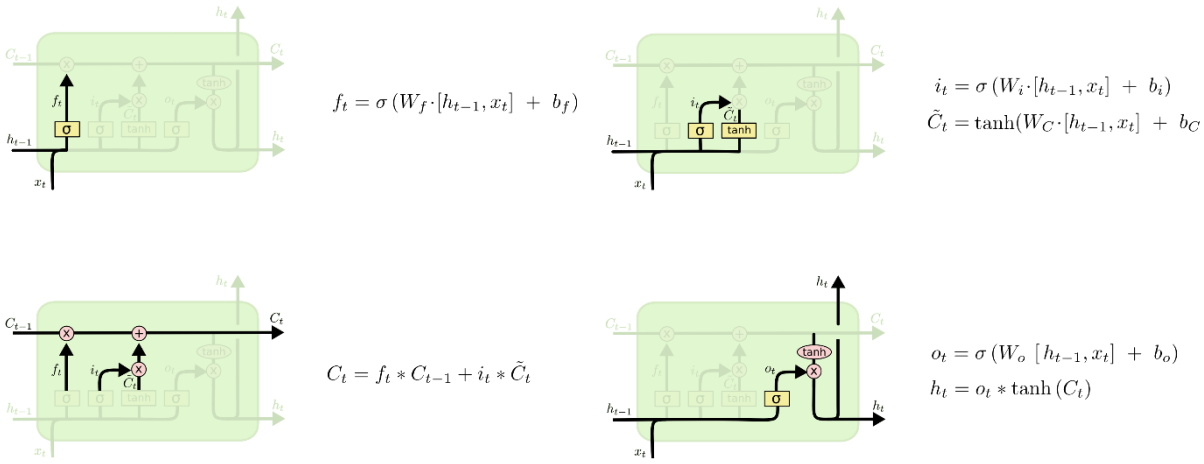

图 2.6 LSTM 原理说明

2.2.3 ConvLSTM 模型

ConvLSTM 是一种结合了卷积神经网络（CNN）和长短期记忆网络（LSTM）的深度学习模型，主要用于处理序列数据，如视频数据、时间序列数据等。它在 LSTM 的基础上引入了卷积操作，使得模型能够有效地捕捉序列数据中的空间特征。ConvLSTM 主要应用于视频分析、动作识别、天气预测等需要同时考虑时间和空间关系的任务。相比于传统的 LSTM 模型，ConvLSTM 能够更好地捕捉序列数据中的空间特征，因此在某些任务中表现更优异。

ConvLSTM 的核心结构包括：

（1）输入门（Input Gate）：输入门控制新输入信息如何影响 LSTM 的状态。它通过卷积操作提取输入数据的空间特征，再结合时间维度的信息，决定哪些新信息应该被添加到细胞状态中。

（2）遗忘门（Forget Gate）：遗忘门决定 LSTM 的状态中哪些信息需要被遗忘。通过卷积操作处理前一时刻的隐藏状态和当前输入数据，计算出一个遗忘系数，应用于当前细胞状态，以选择性地遗忘不重要的信息。

（3）输出门（Output Gate）：输出门确定 LSTM 的状态输出信息。通过卷积操作处理细胞状态和输入数据，结合时间序列中的历史信息，生成当前时刻的输出信息。

（4）状态更新：状态更新结合卷积操作更新细胞状态。新信息通过输入门进入细胞状态，

遗忘门选择性地移除不重要的历史信息，最终更新后的细胞状态包含了当前时刻的重要信息。

具体公式如下所示：

$$f_t = \sigma\left([h_{t-1}, x_t, C_{t-1}] * W_f + b_f\right) \quad 2.6$$

$$i_t = \sigma([h_{t-1}, x_t, C_t] * W_i + b_i) \quad 2.7$$

$$\widehat{C_t} = tanh([h_{t-1}, x_t] * W_c + b_c) \quad 2.8$$

$$C_t = f_t \bigodot C_{t-1} \bigoplus i_t \bigodot \widehat{C_t} \quad 2.9$$

$$o_t = \sigma([h_{t-1}, x_t, C_t] * W_o + b_o) \quad 2.10$$

$$h_t = o_t \bigodot tanh(C_t) \quad 2.11$$

在这些公式中 * 表示卷积操作，σ 表示 sigmoid 激活函数，W 和 b 表示权重和偏置参数。

总而言之，相比于传统的 LSTM 模型 ConvLSTM 有更加优秀的空间特征捕捉能力（由于积操作能够有效提取输入数据中的空间特征，使得 ConvLSTM 不仅能捕捉时间序列中的时间特征，还能捕捉输入数据的空间结构），高效处理高维数据能力（在处理高维数据（如视频、气象数据）时，ConvLSTM 能够通过卷积操作减少参数数量，提高计算效率和模型性能）以及多任务适用性（由于其强大的空间和时间特征捕捉能力，ConvLSTM 在视频分析、动作识别、天气预测、交通流量预测等多种任务中表现优异）。

2.3 台风路径相似性评估的定量方法

2.3.1 相似度计算的台风数据结构和符号定义

由于本文提出的模型将输出台风强度和路径两种台风数据，综合这两种数据的预测的准确度以评估模型性能的需求将变得十分迫切，而传统的 MSE 和 MAE 方法在多个不同维度的输出的评估上表现的粒度太粗糙，故本文决定采用文基于动态时间规整算法的台风相似度分析模型综合模型的两种输出评价预测台风和实际台风的相似度。下面将进一步介绍相似度函数的计算原理。

首先假设两个历史台风时间序列表示为：

$$\mathrm{B} = \{\mathrm{b}_1, \mathrm{b}_2, \mathrm{b}_3, \ldots, \mathrm{b}_\mathrm{n}\} \quad (2.12)$$

$$C = \{c_1, c_2, c_3, \ldots, c_m\} \quad (2.13)$$

其中 B 被命名为长度为 n 的基础时间序列，C 被命名为长度为 m 的比较时间序列。B 和 C 的每个分量，即 b_i 和 c_i，是台风在特定时刻关于各种属性的向量，由一系列键值对组成。台风数据结构如图 2.7 所示。其中 Lon 表示台风中心位置的经度，Lat 表示台风中心位置的维度，Vmax 表示最大风速（强度）。

$$
\begin{aligned}
&\{\\
&\quad "typhoon_1" : [\\
&\quad [Lon_{1,1}, Lat_{1,1}, Vmax_{1,1}],\\
&\quad [Lon_{1,2}, Lat_{1,2}, Vmax_{1,2}],\\
&\quad \cdots\\
&\quad [Lon_{1,l_1}, Lat_{1,l_1}, Vmax_{1,l_1}]\\
&\quad],\\
&\quad "typhoon_2" : [\\
&\quad [Lon_{2,1}, Lat_{2,1}, Vmax_{2,1}],\\
&\quad [Lon_{2,2}, Lat_{2,2}, Vmax_{2,2}],\\
&\quad \cdots\\
&\quad [Lon_{2,l_2}, Lat_{2,l_2}, Vmax_{2,l_2}]\\
&\quad],\\
&\quad \cdots\\
&\quad "typhoon_n" : [\\
&\quad [Lon_{n,1}, Lat_{n,1}, Vmax_{n,1}],\\
&\quad [Lon_{n,2}, Lat_{n,2}, Vmax_{n,2}],\\
&\quad \cdots\\
&\quad [Lon_{n,l_n}, Lat_{n,l_n}, Vmax_{n,l_n}]\\
&\quad]\\
&\}
\end{aligned}
$$

图 2.7 台风相似度评估数据结构说明

同时，两个多维向量 p_1 和 p_2 之间的距离函数定义为：

$$dis(p_1, p_2) = \sum_{i=1}^{n} f_i (p_{1i} - p_{2i})^2 \tag{2.14}$$

其中 p_{1i} 是向量 p_1 的第 i 个分量；p_{2i} 是向量 p_2 的第 i 个分量。f_i是第 i 部分权重的正系数，并且 $\sum_{i=1}^{n} f_i = 1$；n 是此向量的维度。

2.3.2 台风相似度计算步骤

相似度计算具体步骤如下：

（1） 构造一个关于距离函数的 $n \times m$ 的矩阵 Dis，如：

$$\begin{bmatrix} dis(\mathrm{b}_1, \mathrm{c}_1) & dis(\mathrm{b}_1, \mathrm{c}_2) & \cdots & dis(\mathrm{b}_1, \mathrm{c}_\mathrm{m}) \\ dis(\mathrm{b}_2, \mathrm{c}_1) & dis(\mathrm{b}_2, \mathrm{c}_2) & & dis(\mathrm{b}_2, \mathrm{c}_\mathrm{m}) \\ \vdots & & \ddots & \vdots \\ dis(\mathrm{b}_\mathrm{n}, \mathrm{c}_1) & dis(\mathrm{b}_\mathrm{n}, \mathrm{c}_2) & \cdots & dis(\mathrm{b}_\mathrm{n}, \mathrm{c}_\mathrm{m}) \end{bmatrix} \tag{2.15}$$

（2） 根据规则从左上角到右下角填充元素，构造一个关于代价函数的$n \times m$ 的积累矩阵 Acc。构造方法如图 2.8 所示。

（3） 台风时间序列中的元素是线性单位的多维向量。考虑到其中的所有元素的代价函数矩阵是二维的，取最后一个元素的平方根作为相似距离，使得结果的单位为线性，即 $\sqrt{Acc_{n,m}}$。

$$Acc_{i,j} = \begin{cases} Dis_{1,1} & \text{if } i = j = 1 \\ Acc_{1,j-1} + Dis_{1,j} & \text{if } i = 1,\ j > 1 \\ Acc_{i-1,j-1} + Dis_{i,1} & \text{if } i > 1,\ j = 1 \\ min\{Acc_{i-1,j-1}, Acc_{i-1,j}, Acc_{i,j-1}\} & \text{if } i > 1,\ j > 1 \end{cases}$$

图 2.8 Acc 矩阵计算原理

装

订

线

2.3.3 相似距离归一化方法

在台风相似归一化中文章采用 mtanh 函数作为归一化方法。tanh 函数是一种非线性激活函数，将输入值映射到 (−1,1) (−1,1) 的范围内，能够有效地压缩输入数据的范围，减少异常值的影响。这种特性在处理具有复杂模式的气象数据时尤为重要，因为它可以帮助平滑和规范化数据。

况且台风等气象现象的数据往往具有中心对称性。例如，风速或气压等数据在台风眼周围可能会有类似的分布特性。tanh 函数的对称性有助于保持这种特性，使得归一化后的数据仍然能反映台风的本质特征。

该函数作为一种修正的双曲正切函数被 Soboleva 和 Beskorovainyi [6]（2008） 引入，最初用于决策中 的多目标优化。该函数定义为：

$$mtanh(x;a,b,c,d) = \frac{e^{ax} - e^{-bx}}{e^{cx} + e^{-dx}} \tag{2.16}$$

它可以被视为双曲正切函数的通用形式，因为当 a = b = c = d = 1 时，它将变成正常的 tanh 函数。该函数作为一种激活函数被广泛应用于人工智能领域，以增加神经网络的非线性。鉴于 mtanh 函数可以更好地继承 tanh 函数地特征，当 4 个参数 a、b、c 和 d 均等于一个正实数，记为 q 时，将相似距离归一化为 (0,1] 范围内的数字的简化函数定义为：

$$t(x;p) = \frac{p^x - p^{-x}}{p^x + p^{-x}} \tag{2.17}$$

其中，$p = e^q > 1$，并且 p 称为简化参数。为了得出一个清晰易懂的指标来评估两个台风时间序列之间的相似性，定义了一个称为相似度百分比的修正函数：

$$f(x;p) = [1 - t(x;p)] \times 100\% \tag{2.18}$$

本文取 p = 1.005 来计算台风的相似度。Python 代码实现见图 2.9。

```python
def calculate_similarity(pred, truth, method="mse"):
    n = len(pred)
    m = len(truth)
    Dis = tf.zeros((n, m), dtype=tf.float32)
    Acc = tf.zeros((n, m), dtype=tf.float32)

    for i in range(n):
        for j in range(m):
            if method == "mse":
                Dis = tf.tensor_scatter_nd_update(Dis, [[i, j]], [weighted_mse(pred[i], truth[j])])
            elif method == "mae":
                Dis = tf.tensor_scatter_nd_update(Dis, [[i, j]], [weighted_mae(pred[i], truth[j])])

    for i in range(n):
        for j in range(m):
            if i == 0 and j == 0:
                Acc = tf.tensor_scatter_nd_update(Acc, [[i, j]], [Dis[i, j]])
            elif i == 0 and j > 0:
                Acc = tf.tensor_scatter_nd_update(Acc, [[i, j]], [Acc[i, j - 1] + Dis[i, j]])
            elif i > 0 and j == 0:
                Acc = tf.tensor_scatter_nd_update(Acc, [[i, j]], [Acc[i - 1, j] + Dis[i, j]])
            else:
                Acc = tf.tensor_scatter_nd_update(Acc, [[i, j]], [min(Acc[i - 1, j - 1], Acc[i - 1, j], Acc[i, j - 1]) + Dis[i, j]])

    d = tf.sqrt(Acc[-1, -1])
    p = 1.005
    # Normalization
    numerator = p ** d - p ** (-d)
    denominator = p ** d + p ** (-d)
    distance = numerator / denominator

    return 1 - distance
```

图 2.9 相似度计算 Python 代码实现

3 基于多源数据融合的台风强度和路径预测模型实验

3.1 数据集分析以及预处理

3.1.1 数据集分析

为了整合传统预测模型和卫星观测来进行强度和路径预测，本文使用来自两个来源的数据：SHIPS 开发数据库[7]和在线发布的用于强度回归的基准 TC 图像数据集（TCIR，Chen，Chen 和 Lin， 2018）[8]。SHIPS（Statistical Hurricane Intensity Prediction Scheme）开发数据库是一个广泛应用于热带气旋（台风、飓风等）强度预测的数据库。SHIPS 数据库汇集了多种气象和环境参数，这些参数被用于构建统计模型，以预测热带气旋的强度变化。SHIPS 数据库的优势在于其数据的多样性和时效性，为预测模型提供了丰富的输入特征。 SHIPS 数据库包括以下主要数据：① 环境变量：包括海面温度（SST）、大气温度、湿度、风切变（垂直风速变化）、大气稳定度等。这些变量是影响热带气旋强度变化的重要因素。通过监测这些变量，可以获得热带气旋发展的环境条；② 历史台风路径和强度数据：包括过去台风的路径、风速、气压等信息。这些历史数据为模型提供了学习和参考的基础；③卫星观测数据：包括红外（IR）、水汽（WV）和微波（PMW）图像等。这些数据为台风的实时监测和预测提供了重要支持。

其中，红外（IR）图像捕捉地球大气层发出的红外辐射，可用于观察云的高度、温度和云顶温度等信息。通常，云顶温度越低，云的高度越高，因此红外图像可用于识别云的位置和性质。水汽（WV）图像显示大气中水汽含量的分布情况，因为水汽对微波辐射具有较强的吸收能力。水汽图像可用于监测大气中水汽的垂直分布和变化，对天气系统的演变具有重要意义。微波（PMW）图像利用微波辐射来观测大气中的云、降水等信息。相比于红外和可见光，微波在云和降水中的穿透能力更强，因此微波图像对于云、降水的识别和监测具有独特的优势。

装 订 线

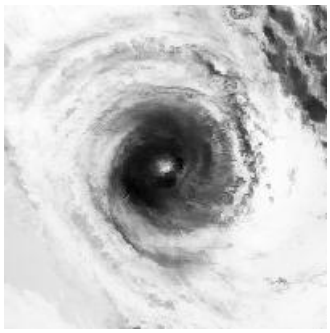

图 3.1 IR 图像-1

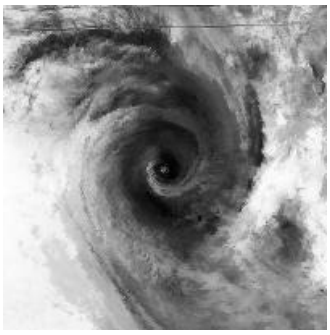

图 3.2 IR 图像-2

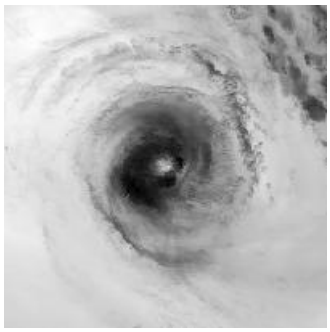

图 3.3 WV 图像 1

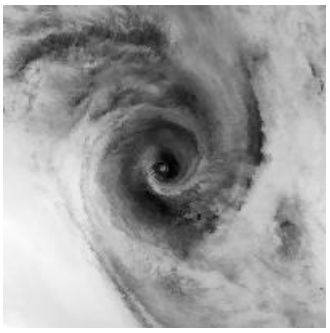

图 3.4 WV 图像 2

图 3.5 VIS 图像 1

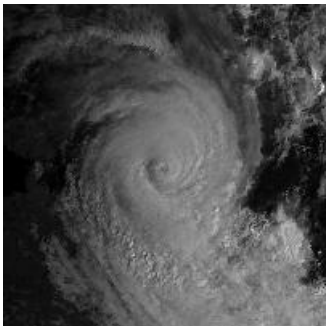

图 3.6 VIS 图像 2

装 订 线

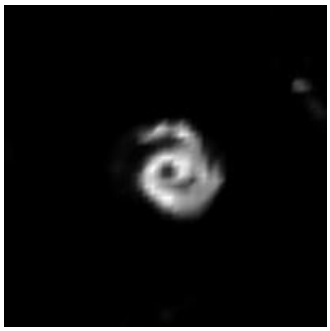

图 3.7 PMW 图像 1

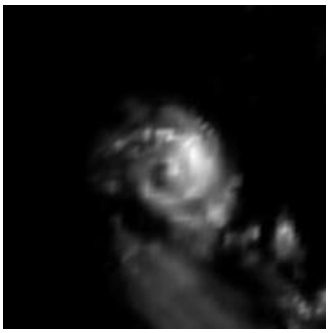

图 3.8 PMW 图像 2

热带气旋辐射数据（TCIR），是从两个开放数据源收集而来的：GridSat：一个全球红外窗口亮度温度的长期数据集，包括三个通道：IR、WV 和 VIS。该数据集自 1981 年以来每隔三小时收集大多数气象静止卫星的数据。分辨率为 7/100 度纬度/经度。CMORPH：将 2003 年至 2016 年的 CMORPH 降水率数据包含在 TCIR 中。CMORPH 提供相对较高的空间和时间分辨率的全球降水分析，使用了仅从低轨道微波卫星观测中获得的降水估计，并通过完全从静止卫星 IR 数据获取的空间传播信息来传输其特征。CMORPH 的分辨率为每三小时 0.25 度。

总而言之，TCIR 收集了 2003-17 年期间 1379 个全球 TC 数据，包括 3 小时卫星红外 （IR）、水汽（WV）和被动微波降雨率 （PMW）TC 图像，水平分辨率为 0.07 纬度/ 经度。示例图像如图 3.9 所示。在模型训练的过程中，单个图像的信息被存储为一个大小是$128 \times 128 \times 4$ 的矩阵。TCIR 数据集的统计信息如表 3.2 所示。

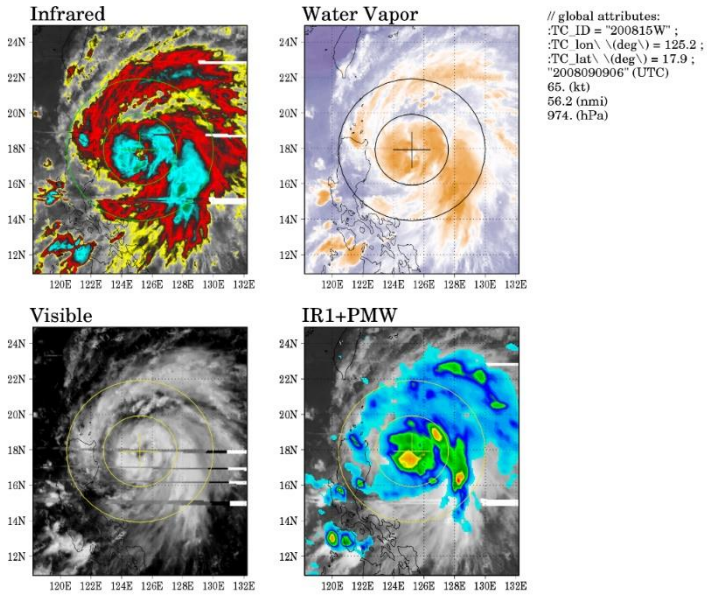

图 3.9 TC 数据集图像示例

表 3.1 TCIR 数据集统计信息

地区（Region）	TC 数量（#TCs）	段数（#Frames）
大西洋（Atlantic）	235	13707
西太平洋（West Pacific）	379	20061
东太平洋（East Pacific）	247	13615
中太平洋（Central Pacific）	19	1479
印度洋（Indian Ocean）	15	3205
南半球（Southern Hemisphere）	330	18434
总计	1285	70501

此外，还使用从 TCIR 数据集收集/导出的季后分析的 TC 相关信息，包括 TC 位置、平移速度、洋盆、到海岸线的距离、当地时间，以及最重要的最佳轨迹 和最大风速 Vmax，用作监督学习的标记数据。 TCIR 数据集收集了联合台风预警中心（JTWC）和修订后的大西洋飓风数据库（HURDATИ）的最佳轨迹数据。关于 SHIPS 环境参数参考了 DeMaria[9]等人的研究。该研究收集了先前研究建议的八个常用环境参数，包括涡去除后 200 hPa 散度（ДИ″）、TC 势强度（POT）、850– 700- hPa 相对湿度（RHLO）、切变相关参数 （SHRD、SHR_x、SHR_y 和 SHRG）和海面温度 （RSST）。季后分析的 TC 相关信息的变量如图 3.2 所示。

region	ID	time	Vmax	R34	RMW	MSLP	lon	lat	SHIPS_D200	SHIPS_RHLO	SHRD	SHRS	SHTS	SHRG	DIVC	U200	EPSS	ENSS	RSST	check	valid_profile	Vmax_best
WP	200401W	2004021100	15.936704	0.0000	0	1006.0	147.2	8.2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	True	15.0
WP	200401W	2004021103	17.512036	0.0000	0	1005.0	146.7	8.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	True	18.0
WP	200401W	2004021106	17.476355	0.0000	0	1004.0	146.2	8.8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	True	20.0
WP	200401W	2004021109	20.755124	0.0000	0	1004.0	145.6	9.2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	True	20.0
WP	200401W	2004021112	20.237560	0.0000	0	1004.0	145.1	9.6	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	True	20.0
WP	200401W	2004021115	24.749842	0.0000	0	1003.0	144.6	10.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	True	23.0
WP	200401W	2004021118	24.808107	0.0000	0	1002.0	144.0	10.4	53	80	94	90	66	246	71	-199	41	15	283	1	True	25.0
WP	200401W	2004021121	25.237652	0.0000	30	1002.0	143.3	10.8	60	80	88	86	79	234	68	-192	38	15	283	1	True	25.0
WP	200401W	2004021200	22.100362	0.0000	60	1002.0	142.7	11.1	67	80	82	82	91	222	65	-185	35	15	283	1	True	25.0
WP	200401W	2004021203	21.776451	0.0000	45	1001.0	142.0	11.3	48	79	105	77	86	237	56	-186	33	16	284	1	True	28.0
WP	200401W	2004021206	21.578376	0.0000	30	1000.0	141.3	11.6	28	78	128	72	81	252	46	-186	30	16	284	1	True	30.0

图 3.10 季后分析 TC 相关信息

图 3.10 中的每一行对应一条热带气旋（TC）数据。每个热带气旋都有一个唯一的编号（ID），每 3 小时记录一次其位置和强度信息。每一条记录中包含了以下变量：

（1） 地区（region）：台风发生的地区（对应表 3.1）。例如，图 3.10 中的 WP 指的是西太平洋。

（2） 编号（ID）：区别不同台风的唯一标识。例如，图 3.10 中的 ID 均为 200401W，说明这些记录均来源于同一个历史台风。

（3） 时间（time）：记录的时刻，格式为年月日时。例如，图 3.10 中的第一条记录时间

为 2004021100，标识此条记录的时刻为 2004 年 2 月 11 日 0 时。

（4） 台风的最大持续风速（Vmax）：台风中心周围最强劲的持续风速。这是指在台风的风眼（即中心）周围，风速的最高值。这里以每小时英里（mph）为单位。Vmax 是衡量台风强度的重要指标之一，通常用来分类台风的级别，比如根据 Saffir-Simpson 飓风风暴等级（针对大西洋和东太平洋的飓风）或台风风力等级（针对西太平洋的台风）。

（5） R34：指在台风中心周围的一个特定半径范围内，风速达到 34 节（约 39 英里/小时）的区域。这个参数通常用来描述台风的外围风场范围。R34 的大小可以提供有关台风的风力结构和外围风场的信息。

（6） 台风的最大风速半径（Radius of Maximum Winds RMW）：指在台风中心周围距离台风眼最近的地方的半径范围，在这个范围内台风的风速达到最大值。RMW 的大小对于理解台风的结构和强度分布至关重要。通常，RMW 越小，风暴的中心结构越紧凑，风速变化也更剧烈。而 RMW 越大，风暴的结构可能更为松散，风速的变化可能更加平缓。

（7） 台风的最低海平面气压（Mean Sea Level Pressure MSLP）：表示在台风中心周围海平面上的气压值。MSLP 是一个重要的气象参数，因为台风的强度和发展通常与其中心的气压有关。通常情况下，MSLP 越低，台风的强度越大，风速也越高。

（8） 台风中心的经度（lon）： 热带气旋中心当前所在位置的经度，单位为度。

（9） 台风中心的纬度（lat）： 热带气旋中心当前所在位置的纬度，单位为度。

（10） SHIPS_D200： SHIPS 模型中的 200hPa 环境湿度。在气象学中，200hPa 通常是指大气中处于高空层次的一个气压高度，对应于大约 12 公里的高度。这个高度通常位于对流层顶部附近。环境湿度是指在这个高度范围内，空气中水蒸气的含量。200hPa 环境湿度是指在 12 公里高度处的大气湿度情况。

（11） SHIPS_RHLO： SHIPS 模型中的低层相对湿度。低层相对湿度是指在较低大气层（通常指地面至几千米高度）中空气中的水蒸气含量与该温度下饱和水汽含量的比率。这个参数表示在台风的发展和移动路径上，低层大气中水汽的含量情况。

（12） SHRD：垂直风切变（Shear）。垂直风切变是指大气中不同高度之间的风速和/或风向的差异。在气象学中指的是垂直方向上大气中风速的变化情况。

（13） SHTS：海温（Sea Surface Temperature）。表示海洋表面的温度。

（14） SHRG：热传输（Heat Flux）。在气象学中，热传输通常指的是海洋和大气之间的热量交换。在台风环境中，热传输是指由海洋向大气传输的热量或能量。

（15） DIVC：静力下沉（Divergence）。在气象学中，静力下沉指的是大气中水平气流的分散或离开某一点的速率。它是空气垂直速度的水平分量。在台风环境中，静力下沉通常与气压分布和气流的变化有关。当空气在某一点处向外散开时，会产生负的静力下沉，这意味着该点的气压倾向于升高。静力下沉的分布可以影响到台风的形成、发展和强度，因为它与台风周围环流的结构和稳定性有关。

（16） 200hPa 环流（U200）：在 200hPa 高度处的水平风流场。

（17） 预测误差标准偏差（EPSS）；集合预测误差标准偏差（ENSS）；海表温度相对于平均值的标准化偏差（RSST）。

（18） 最佳 Vmax 估计（Vmax_best）：指在给定环境条件下，通过某种模型或算法计算出的最优的台风最大持续风速估计值。这个值通常是通过结合多种观测数据和预测模型来计算得出的。

通过整合 SHIPS 开发数据库和 TCIR 数据集，本文实现了传统统计模型和卫星观测数据的结合，提升了台风强度和路径预测的准确性。SHIPS 数据库提供了丰富的环境变量和历史台风数据，使得模型能够学习到台风发展的环境条件和历史模式。而 TCIR 数据集提供了高分辨率的卫星图像数据，为模型提供了实时的空间和时空信息。

此外本文还进行了对于“理论部分”提到的“台风相似性评估的定量方法”，利用从公开平台获取到的历史台风数据，对度量方法有效性和准确性进行验证。所以，本文还使用了来源于中国浙江省水利部台风轨迹实时发布系统 （http://typhoon.zjwater.gov.cn/default.aspx）提供的 2010-2020 年台风数据。数据经过采集和预处理后，转换为 JSON 格式的文件进行进一步分析，该文件由 272 个 键值对组成，如下图所示。每个键值对由一个键和一个值，编码为“key”： “value”，其中键是台风的索引，值是按时间排序的列表。台风数据结构如图 3.11 所示。

$$
\begin{aligned}
&\{\\
&\quad \text{"typhoon}_1\text{"}: [\\
&\qquad [\mathrm{Lon}_{1,1}, \mathrm{Lat}_{1,1}, \mathrm{P}_{1,1}, \mathrm{L}_{1,1}, \mathrm{V}_{1,1}],\\
&\qquad [\mathrm{Lon}_{1,2}, \mathrm{Lat}_{1,2}, \mathrm{P}_{1,2}, \mathrm{L}_{1,2}, \mathrm{V}_{1,2}],\\
&\qquad \ldots,\\
&\qquad [\mathrm{Lon}_{1,l_1}, \mathrm{Lat}_{1,l_1}, \mathrm{P}_{1,l_1}, \mathrm{L}_{1,l_1}, \mathrm{V}_{1,l_1}]\\
&\quad],\\
&\quad \text{"typhoon}_2\text{"}: [\\
&\qquad [\mathrm{Lon}_{2,1}, \mathrm{Lat}_{2,1}, \mathrm{P}_{2,1}, \mathrm{L}_{2,1}, \mathrm{V}_{2,1}],\\
&\qquad [\mathrm{Lon}_{2,2}, \mathrm{Lat}_{2,2}, \mathrm{P}_{2,2}, \mathrm{L}_{2,2}, \mathrm{V}_{2,2}],\\
&\qquad \ldots,\\
&\qquad [\mathrm{Lon}_{2,l_2}, \mathrm{Lat}_{2,l_2}, \mathrm{P}_{2,l_2}, \mathrm{L}_{2,l_2}, \mathrm{V}_{2,l_2}]\\
&\quad],\\
&\quad \ldots,\\
&\quad \text{"typhoon}_n\text{"}: [\\
&\qquad [\mathrm{Lon}_{n,1}, \mathrm{Lat}_{n,1}, \mathrm{P}_{n,1}, \mathrm{L}_{n,1}, \mathrm{V}_{n,1}],\\
&\qquad [\mathrm{Lon}_{n,2}, \mathrm{Lat}_{n,2}, \mathrm{P}_{n,2}, \mathrm{L}_{n,2}, \mathrm{V}_{n,2}],\\
&\qquad \ldots,\\
&\qquad [\mathrm{Lon}_{n,l_n}, \mathrm{Lat}_{n,l_n}, \mathrm{P}_{n,l_n}, \mathrm{L}_{n,l_n}, \mathrm{V}_{n,l_n}]\\
&\quad]\\
&\}
\end{aligned}
$$

图 3.11 台风数据结构图

3.1.2 数据集预处理

为了获取适合深度学习模型输入的数据，我们需要对 TCIR 数据集进行一系列的处理和筛选。首先将所有的图像数据和季后分析的台风信息存储在一个 H5 文件中（TCSA.h5）。对完整数据集的具体处理和筛选步骤如下：

（1） 划分数据集。将所有台风记录划分为 3 个部分：训练集（包含 2003 年至 2014 年的台风记录。这部分数据将用于模型的训练，是模型学习的主要依据）、验证集（包含 2015 年至 2016 年的台风记录。验证集用于在每个训练轮次结束之后评估模型的性能，调整模型参数以避免过拟合）和测试集（包含 2017 年的台风记录。测试集用于最终评估模型的性能，确保模型在未见过的数据上也能表现良好），以便每个训练轮次结束之后使用验证集和测试集评估模型的性能；

```python
def data_split(image_matrix, info_df, phase):
    if phase == 'train':
        target_index = info_df.index[info_df.ID < '2015000']
    elif phase == 'valid':
        target_index = info_df.index[(info_df.ID > '2015000') & (info_df.ID < '2017000')]
    elif phase == 'test':
        target_index = info_df.index[info_df.ID > '2017000']

    new_image_matrix = image_matrix[target_index]
    new_info_df = info_df.loc[target_index].reset_index(drop=True)
    return new_image_matrix, new_info_df
```

图 3.12 数据集划分函数

（2） 处理单个台风数据。根据台风记录的 ID，将同一台风的所有记录整合在一起，作为一个相对独立的数据单元。这样可以确保模型能够捕捉到每个台风的完整生命周期信息。格式化每个字段对应的数据并存储在 Python 的变量中。具体来说，将各类数据字段（如风速、气压等）进行标准化处理，以便模型更好地理解和学习。将所有的 SHIPS 环境变量统一放在“env_feature”字段中。这些环境变量是影响台风发展的关键因素，包括海洋温度、湿度、风切变等。

（3） 对每个台风的路径数据（经度和纬度）进行归一化操作，在单个台风与其归一化函数之间构建映射关系，归一化函数作为一个全局函数将存储在 .pkl 文件中。鉴于在首次迭代训练的模型缺少归一化步骤使得模型表现较差，在后续的训练轮次中加入训练数据的归一化的预处理。归一化函数如公式 3.1 所示，这个函数简单明了，计算方便，能够快速应用于大规模数据处理。对于每个特征，只需要计算其最小值和最大值，并进行简单的除法运算即可完成归一化。这种简便的计算方式能够大大提高数据处理的效率，同时它还保持了原始数据中各特征值之间的相对关系——通过将数据缩放到一个标准范围，极值（即最小值和最大值）的影响被有效地控制在一定范围内，避免了极值对模型训练造成的过大干扰。特别是在处理异常值或噪声数

据时，这种归一化方法能够显著提高数据的稳定性和模型的鲁棒性。将预处理完成的训练数据存储在 TFRecord 文件中，确保数据在 TensorFlow 环境下的高效读取和使用。

$$R = \frac{R_{min}}{R_{max} - R_{min}} \tag{3.1}$$

（4） 切割时间序列数据。由于 LSTM 的期望输入和输出均是关于某个台风的时间序列数据，所以需要对在第二步中处理完成的单个台风的所有记录进行切割。设定预测步长和编码步长：预测步长（estimate_distance）设定为 8，编码步长（encode_length）设定为 1。这意味着每次预测将基于前 8 个时间步的记录进行。过滤历史长度：过滤掉历史长度（history_len）不足的台风记录，即所有的记录数量少于特定值（min_history_len）的历史台风。其中，min_history_len 如公式 3.2 所示。这个公式确保每个台风记录至少包含足够的历史数据和预测步长，以便模型进行训练和预测。最后，对每个台风的记录进行时间序列切割，生成 LSTM 模型可以接受的输入格式。具体来说，每个台风的数据将被分割成若干个固定长度的子序列，每个子序列包含连续的时间步数据，用于训练 LSTM 模型。

$$\mathrm{min\_history\_len} = \mathrm{encode\_length} + \mathrm{estimate\_distance} + 2 \tag{3.2}$$

数据预处理是深度学习模型训练中的关键步骤。通过对 TCIR 数据集的处理和筛选，可以确保模型能够获得高质量、规范化的数据，从而提高模型的训练效果和预测性能。

通过上述一系列处理和筛选步骤，我们可以将 TCIR 数据集转换成适合深度学习模型输入的格式。这些步骤包括数据集划分、单个台风数据处理、时间序列数据切割等，确保模型能够获得高质量、规范化的数据，从而提高模型的训练效果和预测性能。数据预处理是深度学习模型训练中的关键步骤，通过合理的数据处理和筛选，可以有效提高模型的性能和泛化能力。

3.2 基于 CNN 和 LSTM 模型的台风强度和路径预测

台风强度预测是一个重要的气象学问题，对于防灾减灾和社会经济发展有着重要意义。近年来，随着深度学习技术的发展和应用，一些新的台风强度预测方法被提出。深度学习是一种基于人工神经网络的机器学习方法，能够从大量的数据中自动地提取特征和规律，从而实现复杂的任务。深度学习方法可以分为监督学习和无监督学习。其中监督学习是指利用已知标签的数据来训练模型，从而实现分类或回归等任务。而无监督学习是指利用无标签的数据来训练模型，从而实现聚类或生成等任务。

本文使用 CNN 、LSTM 和 ConvLSTM 模型相结合的架构，以大气数据和海表数据作为输入数据，模型输出 3 个预测值：台风强度、台风中心经度和台风中心纬度。

3.2.1 ConvLSTM_CNN 模型训练

本文提出的基于多源数据融合的台风强度和路径预测的机器学习模型的架构图如图 3.13 所示。[H × W × D]表示输出的特征图的高度、宽度和深度（通道数目）。辅助特征（Auxiliary Features），例如，TC 信息和 SHIPS 参数，与输出回归器中的特征连接。

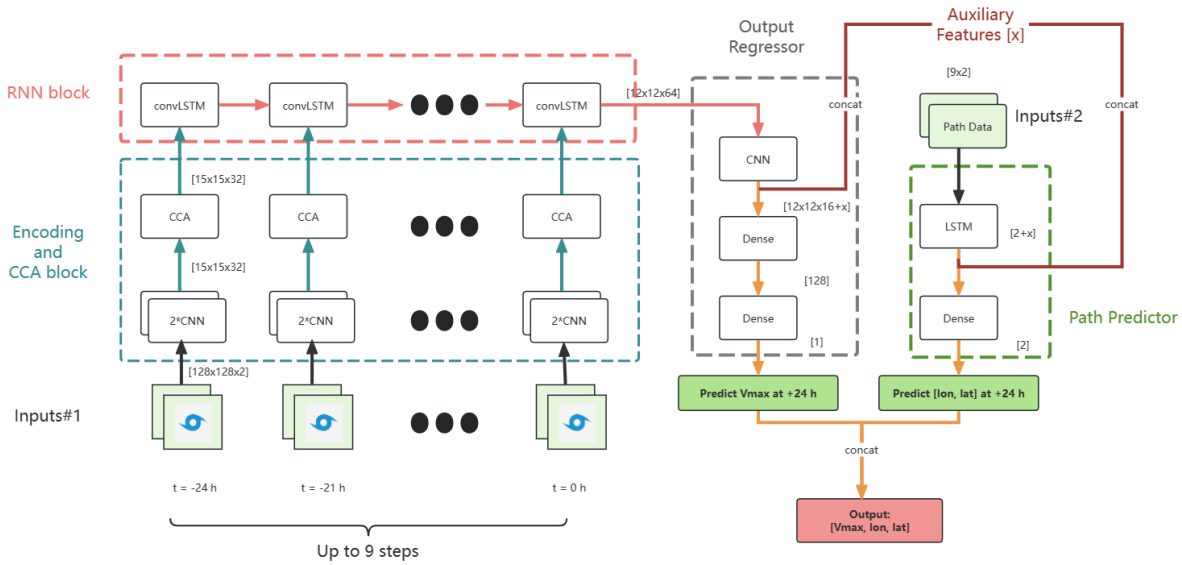

图 3.13 实验模型 ConvLSTM_CNN 架构图

下面将对模型的处理流程进行详细说明。

（1）对于 TC 强度的预测：首先，使用卷积层从卫星图像（图 3.3 输入部分， Inputs#1）中自主提取基本特征 （图 3.3，解码和 CCA 模块 Encoding and CCA block）。随后，在选择阶段（图 3.3 RNN 模块，图片最上方红色方框部分），部署了具有 ConvLSTM 的 RNN 块来对提取到的基本特征进行演化（feature evolution）。模型的最后部分（图 3.3，输出回归器 Output Regressor）由一个卷积层和两个密集层组成，进行特征到强度回归，预测 +24 h 时的 Vmax。正如之前的研究（例如，Bai 等人[10]）使用 IR 和 PMW 图像进行 TC 强度估计/预测。本研究中的模型使用 IR 和模拟 PMW（PMW*）图像作为模型输入，尺寸为$128 \times 128 \times 1\timest$，以 TC 位置为中心（图 3.3，输入）。本文使用的 PMW* 图像由 Chen[11]等人提出的混合 GAN-CNN 模型生成。值得注意的是，PMW* 图像可以从 IR 和 WV 图像中检索，从而用于实时预报（因为收集的 PMW 降雨率不是实时数据）。Chen 等人遵循 Olander 和 Velden 的概念开发了这种深度生成模型，即可以通过使用专门设计的色标绘制 IR 和 WV 图像之间的差异来模拟 PMW 图像。此外，在将特征传递到密集层之前，将环境/TC 相关预测器连接到展平的特征上，将其用作辅助特征。

此外，批量归一化（Ioe & Szegedy，2015）[12]将被应用到这些层。编码块（图 3.3 中间蓝色方框部分 Encoding and CCA block）的第二部分是 Bai [10]等人提出的跨通道注意（CCA）模块。 CCA 模块的灵感来自用于强度估计的 Dvorak 类型的方案（Olander 等人，2021 年[13]；Olander 和 Velden，2019 年[14]）。执行此操作是为了突出与 TC 强度相关的关键云特征（例如，眼睛、中心密集的阴云图案和剪切引起的不对称）。具体来说，CCA 模块计算关键二维加权掩码（two-dimensional importance weighting mask），然后通过 Hadamard 乘积将其应用于从先前编码

卷积层从而获得更多的特征图。范围从 0 到 1 的关键二维加权掩码被实现为具有用于归一化的 sigmoid 函数的两层 CNN。因此，CCA 模块有助于对隐藏图中模型应更多被关注的位置进行总体重要性评估。卷积 LSTM（ConvLSTM，Shiet al.，2015）单元用 于分析编码块提取的演化特征。 ConvLSTM 算法是与 CNN 结合的 LSTM 模块，用卷积运算代替所有矩阵乘法来对数据的空间和时间方面进行建模。在 RNN 块之后，大小为 [12 × 12 × 64] 的特征被传递到最终输出回归器中（图 3.3，灰色虚线框 Output Regressor）。卷积层对这些特征进行变换以减少它们的维度，并且与环境/TC 相关的预测器被连接为辅助特征。最后，两个完全连接的密集层将特征回归到最终预测 Vmax at+24 h，并采用 dropout （模型将随意丢弃 20%的数据）来减轻潜在的过度拟合。

同时，由于预测 RI 的难度部分在于其稀有性，因此应用加权均方误差 （MSE） 损失来强制模型对强化事件赋予更高的权重：

$$w = tanh\left(\frac{\delta \mathrm{V}_{\max} - 20}{10}\right) \times 1000 + 1000.1 \tag{3.3}$$

$$WeightedMSE = \frac{1}{N}\sum_{i=1}^{N} w \times (X_i - Y_i)^2 \tag{3.4}$$

其中，δV_{max} 是强化率，N 是训练数据的样本数（或批量大小），X 是+24 h 模型预测的 Vmax，Y 是目标标记数据。

（2）对于 TC 路径的预测：台风的路径数据（$[1 \times 2 \times t]$）作为 LSTM 模型的输入（图 3.3 输入部分， Inputs#2）。考虑台风位置的二维坐标表示，对于持续时间连续的台风，其路径可以用一组时间序列表示（输入数据，对应图 3.3 右上角 Input#2）：

$$[x_1^1, x_2^1]^T, [x_1^2, x_2^2]^T, \ldots, [x_1^t, x_2^t]^T \tag{3.5}$$

其中，$\left[x_1^i, x_2^i\right]^T$是台风坐标信息的二维向量表示，在本文中$x_1$表示经度，$x_2$表示纬度。i 表示其在序列中的索引，每两个相邻元素的时间差为 3h。输出和输入一样是一个二维向量构成的序列。输入序列和相应的输出序列构成一个样本。在图 3.3 中，输入数据首先通过一个 LSTM 节点，然后，和强度预测一样，与环境 TC 相关的预测器被连接为辅助特征，通过一个密集层输出二维位置信息的时间序列。

最后将预测的台风强度和路径组合到一起，共同作为模型的输出。模型的部分代码如图 3.14 所示。

装 订 线

Python ▾ 取消自动换行 | 复制

```python
class Model(tf.keras.Model):
    def __init__(self):
        super().__init__()

        self.input_norm = layers.BatchNormalization()
        self.image_encoder_layers = [
            layers.Conv2D(filters=16, kernel_size=4, strides=2, activation='relu'),
            layers.BatchNormalization(),
            layers.Conv2D(filters=32, kernel_size=3, strides=2, activation='relu'),
            layers.BatchNormalization()
        ]

        self.cross_channel_attention = [
            layers.Conv2D(filters=32, kernel_size=3, strides=1, padding='same', activation='relu'),
            layers.BatchNormalization(),
            layers.Conv2D(filters=1, kernel_size=2, padding='same', strides=1),
            # layers.Softmax()
        ]

        self.rnn_block = layers.ConvLSTM2D(
            filters=64, kernel_size=4, dropout=0.0,
            recurrent_dropout=0.0, return_sequences=False
        )

        self.lstm_layer = layers.LSTM(64, return_sequences=True)

        self.rnn_output_encoder = layers.Conv2D(filters=64, kernel_size=1, strides=1, activation='relu')
        # self.output_layers = [
        #     layers.Dense(units=128, activation='relu'),
        #     layers.Dropout(rate=0.2),
        #     layers.Dense(units=1),
        # ]

        self.int_output_layers = [
            layers.Dense(units=128, activation='relu'),
            layers.Dropout(rate=0.2),
            layers.Dense(units=1),  # 3 for intensity, latitude, and longitude
        ]

        self.pat_output_layers = [
            layers.Dense(units=128, activation='relu'),
            layers.Dropout(rate=0.2),
            layers.Dense(units=2),  # 3 for intensity, latitude, and longitude
        ]
```

图 3.14 ConvLSTM_CNN 模型部分 Python 代码

3.2.2 ConvLSTM_CNN 模型训练实验步骤

实验所用的环境为 Google Colab 提供的云计算资源“运行时类型”选择为 A100GPU，并连接到 Google Drive 装载训练数据。

实验参数的设置如表 3.2 所示：

表 3.2 实验参数设置

参数名称	值
batch_size	50
encode_length	1
estimate_distance	8
rotate_type	‘no’
input_image_type	[2]
evaluate_freq	1
class_weight	-1
learning_rate	0.0005
max_epoch	30

其中：

（1）evaluate_freq 表示在训练过程中每隔多少个训练周期就进行一次评估。在每次评估中，使用验证数据集 datasets["valid"] 来评估模型在当前训练状态下的性能。本实验中 evaluate_freq 设置为 1，那么每个训练周期后都会进行一次评估。

（2）class_weight 是一个类别权重的设置。通过设置类别权重，可以调整模型对不同类别样本的重视程度。在本实验中设置一个权重为 -1 的类别权重，这意味着所有类别的权重都是相同的。

（3）learning_rate 学习率，用于控制参数更新的步长。

（4）max_epoch 最大训练周期数（限制训练过程的持续时间）。训练过程将在达到指定的最大训练周期数后停止，即使模型还没有完全收敛。

配置文件的设置如图 3.15 所示：

```yaml
experiment_name: 'ens21'

model: ConvLSTM_CCA_Pro

data:
  data_folder: 'TCSA_data'
  batch_size: 50
  encode_length: 1
  estimate_distance: 8
  rotate_type: 'no'
  input_image_type: [2]

training_setting:
  evaluate_freq: 1
  class_weight:
    - 1
  learning_rate: 0.0005
  max_epoch: 30
```

图 3.15 配置文件图

在模型编译方面，本模型使用 Adam 算法作为优化器，这是一种基于梯度下降的自适应学习率的优化算法，能够有效地提高模型的收敛速度和性能。使用 Huber 作为模型预测的损失函数，这是一种平滑的绝对误差损失函数，取 MAE 与 MSE 中的较小值，能够减少异常值的影响，也适用于当前问题的预测。指定使用平均绝对误差作为评估指标，这是一种衡量预测值和真实值之间差异的指标，越接近 0 表示效果越好。

本文使用 3.2.1 中提出的模型对数据集进行训练，指定训练 30 个周期，即对整个训练数据集进行 30 次前向传播和反向传播。在每个周期结束后对模型进行评估，并计算出验证损失值和验证评估值。每轮训练完成后会返回一个历史对象，它会记录每个周期的损失值和评估值。通过观察历史对象中记录的损失值和评估值变化趋势，我们可以判断模型是否收敛或过拟合，并根据情况调整模型参数或训练策略。实验界面如图 3.16 所示。

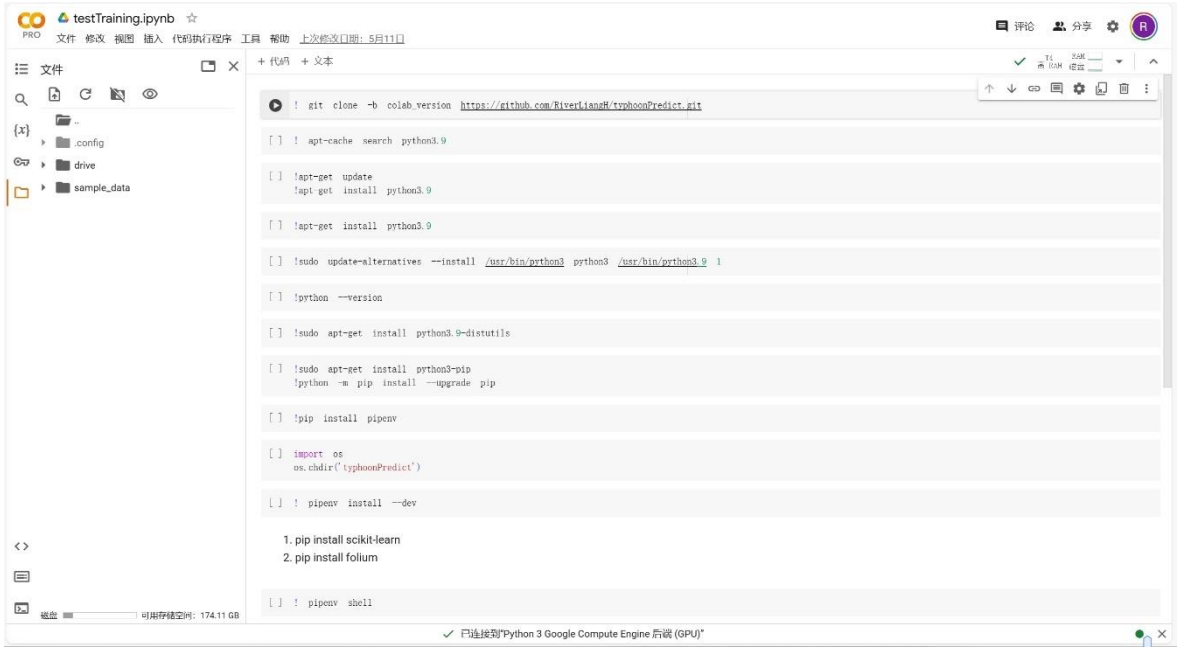

图 3.16 训练实验界面

实验一共耗时 12 h。使用了 Google Colab Pro 的云计算资源，系统 RAM 51.0 GB，GPU RAM 15.0 GB，磁盘 201.2 GB。数据集文件存放在 Google Drive 上，实验开始时，首先从 github 仓库拉取实验代码，然后装载 Google Drive，即谷歌云盘中的数据集文件 TCSA.h5。在 Colab 中，python 代码的执行是基于.ipynb 文件，也就是 Jupyter Notebook 格式的 python 文件。这种笔记本文件与普通.py 文件的区别是可以分块执行代码并立刻得到输出，同时也可以很方便地添加注释，这种互动式操作十分适合一些轻量的任务。同时由于免费的计算资源的运行时长没有保证，购买了 100 个计算单元。

实验运行的计算资源配置如图 3.17 所示：

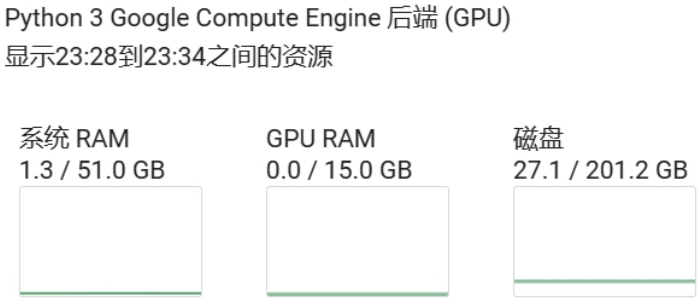

图 3.17 实验计算资源配置参数

由于 Google Colab 上的 Python 和 Python Package 都是预先安装好的，但是实验需要的

python 环境和系统预装未必一致，所以需要先在分配到的 GPU 资源上更换 python 环境。同时，本文使用 pipenv 工具来创建虚拟环境。具体命令如下所示：

```
! git clone -b colab_version https://github.com/RiverLiangH/typhoonPredict.git
! apt-cache search python3.9
! apt-get update
! apt-get install python3.9
! apt-get install python3.9
! sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.9 1
! python –version
! sudo apt-get install python3.9-distutils
! sudo apt-get install python3-pip
! python -m pip install --upgrade pip
! pip install pipenv
! pipenv run python main.py experiments/ens21.yml
```

最后一句是启动实验的命令，表明参数配置文件选取 experiments 文件夹下的 ens21.yml。配置文件见图 3.15。由于如果分配到的计算资源一定时间没有任何操作之后，会被系统自动释放，所以在实验过程中要及时保存生成的模型文件以及实验日志 Log，即将分配的实例空间中的文件拷贝到 Google Drive，拷贝的命令如下所示：

```
cp -r /content/typhoonPredict/logs /content/drive/MyDrive/trainLog/20240502_tcsa
cp -r /content/typhoonPredict/saved_models /content/drive/MyDrive/trainLog/20240502_tcsa
```

由于 Python 拥有丰富的机器学习库和框架，如 TensorFlow、PyTorch、scikit-learn 等，这些工具简化了模型的构建、训练和部署过程，使开发者能够快速实现复杂的机器学习任务，所以本文整体的编码语言选择了 python 语言，并且使用 TensorFlow 作为整体框架，并选择 TensorBoard 工具进行实验结果的可视化。

代码的主要模块如图 3.18 所示。其中“modules”包放置实现不同板块功能的 .py 文件，data_downloader.py 主要从公开的数据集网站 TCIR 下载模型训练所需要的数据集；data_handler.py 是数据预处理板块，这一板块输出 datasets 变量包含“train”“valid”“test”三个元素，将直接被应用于模型的训练，也就是说 data_handler.py 处理后的数据将直接与训练模型进行交互；experiment_helper.py 则用来存放实验的一些配置信息，包括，实验结果的存放位置等；model_constructor.py 则创建一个模型类 ConvLSTM_CNN 的实例，将其载入到模型训练程序中；model_trainer.py 是训练的主程序代码所在的文件；tfrecord_generator.py 是直接与数据集文件 TCSA.h5 进行交互的代码块；training_helper.py 则存放模型性能相关参数的计算代码，包括在上文提及的台风相似度，MAE 和 MSE。

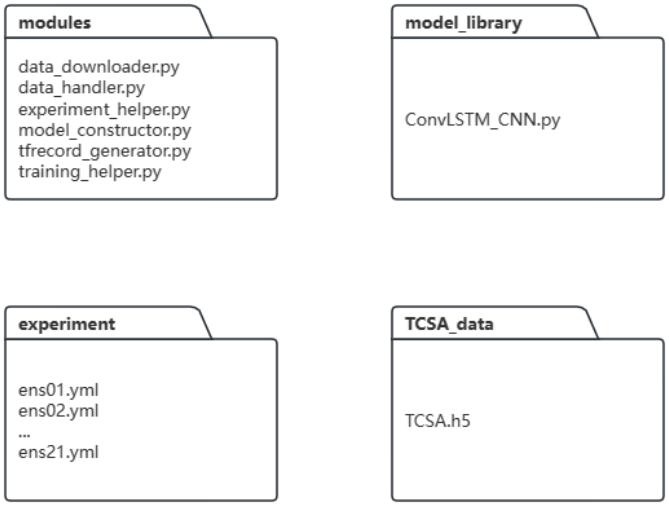

图 3.18 代码模块

装 订 线

4 实验结果和讨论

为了验证本文提出的模型对于台风强度和路径预测的有效性，本文在预处理后的数据集上进行了一系列模型预测实验。本文使用在数据预处理阶段中划分出的验证数据集（2015-2016）和测试数据集（2017）对每一训练周期得到的模型性能进行评估。

图 4.1 展示了训练的 30 个周期的系统日志。

```
2024-05-02 06:07:14.509896: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1850] Cannot dlopen some GPU libraries
Skipping registering GPU devices...
2024-05-02 06:07:14.513133: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
data_folder ../drive/MyDrive/typhoonPredict/TCSA_data
Executing epoch #1
Completed 1 epochs, do some evaluation
Executing epoch #2
Completed 2 epochs, do some evaluation
Executing epoch #3
Completed 3 epochs, do some evaluation
Executing epoch #4
Completed 4 epochs, do some evaluation
Executing epoch #5
Completed 5 epochs, do some evaluation
Executing epoch #6
Completed 6 epochs, do some evaluation
Executing epoch #7
Completed 7 epochs, do some evaluation
Executing epoch #8
Completed 8 epochs, do some evaluation
Executing epoch #9
Completed 9 epochs, do some evaluation
Executing epoch #10
Completed 10 epochs, do some evaluation
Executing epoch #11
Completed 11 epochs, do some evaluation
Executing epoch #12
Completed 12 epochs, do some evaluation
Executing epoch #13
Completed 13 epochs, do some evaluation
Executing epoch #14
Completed 14 epochs, do some evaluation
Executing epoch #15
Completed 15 epochs, do some evaluation
Executing epoch #16
Completed 16 epochs, do some evaluation
Executing epoch #17
Completed 17 epochs, do some evaluation
Executing epoch #18
Completed 18 epochs, do some evaluation
Executing epoch #19
Completed 19 epochs, do some evaluation
```

图 4.1 训练 30 周期系统日志

在每个训练周期结束之后，本文以 MAE（Mean Absolute Error）和 MSE （Mean Square Error），即绝对均方误差和平方均方误差作为模型的评估指标。MSE（Mean Squared Error）和 MAE（Mean Absolute Error）是两种常用的衡量模型的预测结果与实际结果之间的差异评估指标，MSE 是预测值与实际值之间差异的平方的平均值。计算公式如公式 4.1 所示：

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2 \tag{4.1}$$

其中，y_i 是实际值（ground truth），$\hat{y}_i$ 是预测值，n 是样本数量。MSE 的值越小，说明模型的预测结果越接近真实值。

MAE 是预测值与实际值之间差异的绝对值的平均值。计算公式如公式 4.2 所示：

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.2}$$

与 MSE 不同，MAE 不取平方，因此它对异常值的敏感性较低；而 MAE 计算出来的值越小，说明模型的预测结果越接近真实值。在选择 MSE 或 MAE 作为评估指标时，通常需要考虑具体的应用场景和对模型误差的敏感程度。例如，如果异常值对结果影响较大，可以选择 MAE ；如果需要更关注大误差的情况，可以选择 MSE 。本文同时计算了每次训练结束后验证集和测试集上的 MAE 和 MSE ，分别保存了在 MAE 和 MSE 上表现最优的模型。

损失函数则采用简单的对预测变量（强度、经度、纬度）的 MSE 的加权平均数，MSE 随训练周期（时间）的大小变化如图 4.2 所示。

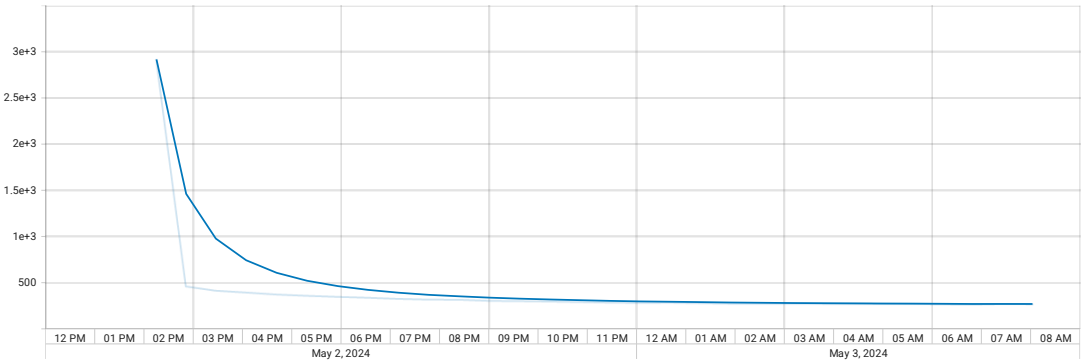

图 4.2 mean square error 损失函数的大小随训练周期数变化

4.1 关于路径预测的模型输出性能表现

模型在训练过程中，路径预测的表现随训练周期数（时间）的变化如图 4.3（MAE）和图 4.4（MSE）所示。

从图 4.3 中可以看出，随着训练周期的增加，模型路径预测的路径的损失值逐渐下降，评估值逐渐上升，说明模型的性能在不断提高。

同时也可以看出，模型在验证集上的损失值与在测试集上相差不大（颜色较深的曲线为验证集上的 MAE ，颜色较浅的为测试集上的 MAE），说明模型没有出现过拟合或欠拟合的现象。

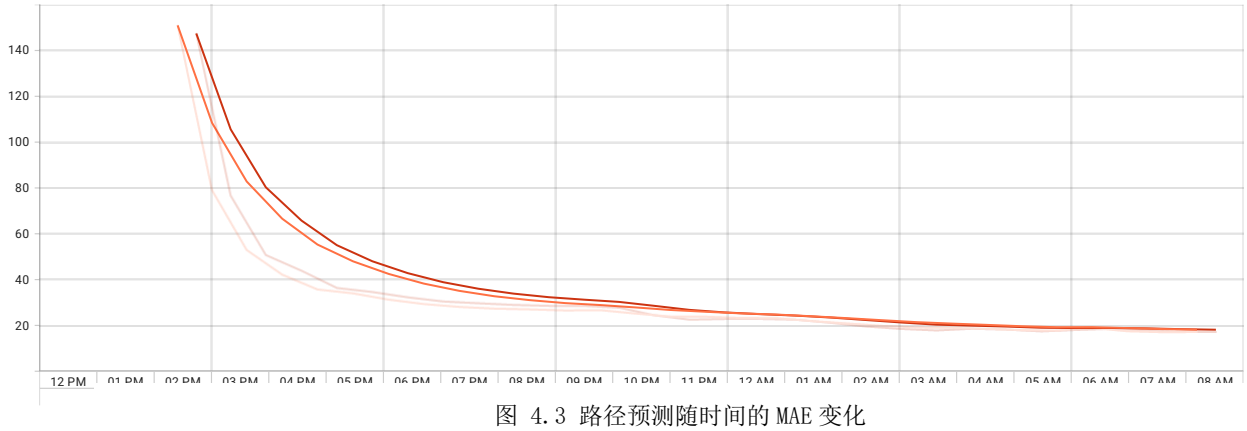

图 4.3 路径预测随时间的 MAE 变化

图 4.4 同。

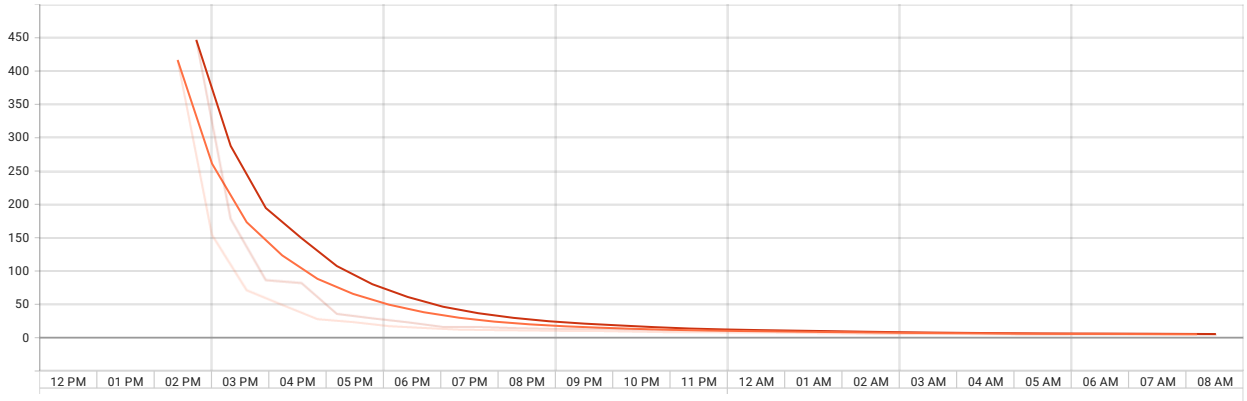

图 4.4 路径预测随时间的 MSE 变化

结合图 4.3 和图 4.4，MAE 和 MSE 的值均在 09PM 之后趋于稳定，意味着模型在该时间点之后对路径数据的预测性能转向较为稳定的状态。同时也表明了这两个性能的衡量指标具有一致性。其中编号为“200402W”的台风的预测路径和实际路径的对比图如图 4.5 所示。其中，红色路径是台风的真实数据，蓝色部分是模型的预测路径。

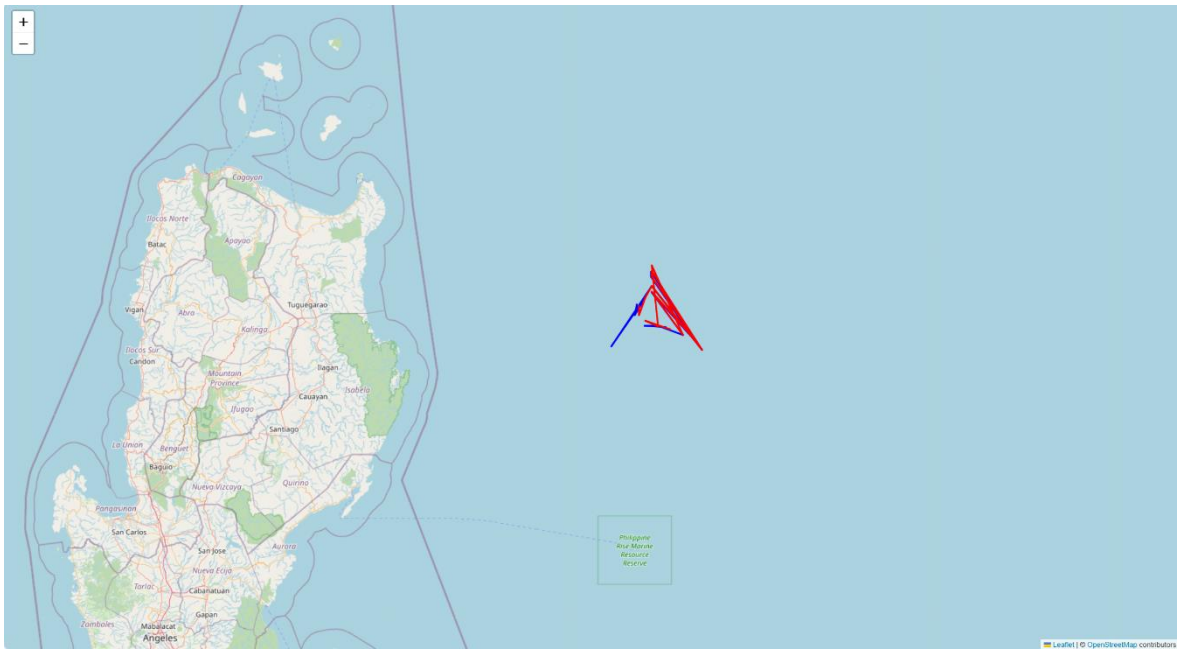

图 4.5 200402W 台风预测路径和实际路径对比

从图 4.5 可以看出，尽管模型在 MAE 和 MSE 指标上均有收敛，但是由于台风位置（经度、纬度）对于模型的精度要求较高（相差为 1 的经度在地图上已经是很远的距离了），所以尽管预测路径已经十分靠近实际路径，但是拟合度仍有待提高。

4.2 关于强度预测的模型输出性能表现

模型在训练过程中，强度预测的表现随训练周期数（时间）的变化如图 4.6（MAE）。从图中可以看出，模型关于台风强度 Vmax 的预测一直没有很好的收敛。在 10PM 左右出现最低点。其可能的原因是受到损失函数中路径预测部分 MSE 的影响。

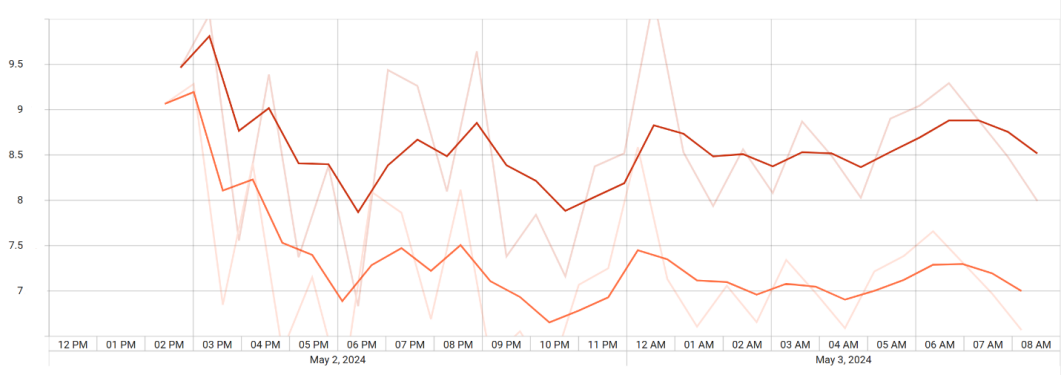

图 4.6 台风强度的 MAE 随时间的变化示意图

装 订 线

编号为“200402W”的台风的预测强度和实际强度对比图如图 4.7 所示。模型在大多数样本点上表现出较高的准确性。在第 0 到第 3 个样本点和第 6 到第 8 个样本点，预测值与真实值几乎完全一致。但在第 4 和第 5 个样本点，以及第 10 到第 12 个样本点，预测值与真实值之间存在一定的偏差，特别是在第 4 和第 11 个样本点，偏差较为明显。这可能是由于这些样本点的真实强度值存在较大波动，模型未能完全捕捉这些变化。

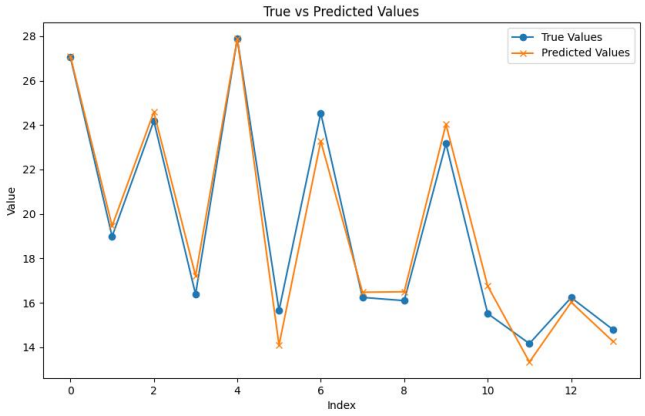

图 4.7 200402W 台风预测强度和实际强度对比

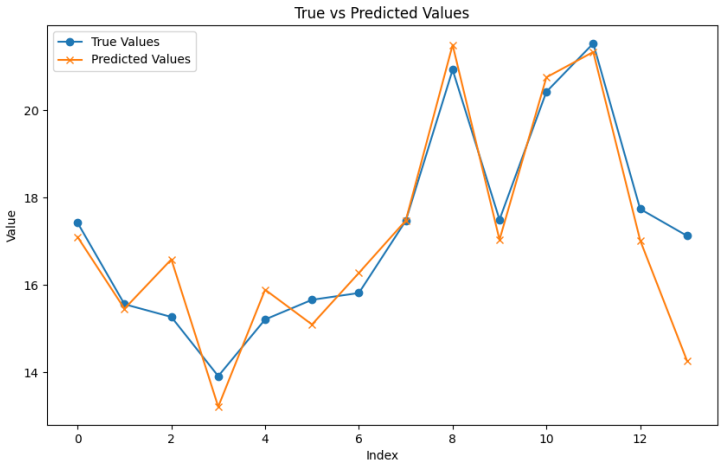

图 4.8 201303C 台风预测强度和实际强度对比

从图 4.9 中可以看出，模型在大多数情况下能够较好地跟踪真实值的变化趋势。在第 0 至第 6 个样本点，预测值与真实值几乎完全重合，表明模型在这些点上具有较高的准确性。然而，在第 7 到第 12 个样本点，尽管预测值与真实值的总体趋势相符，但存在一定的偏差，尤其是在

第 8 和第 9 个样本点，预测值明显低于真实值。这可能是由于这些样本点对应的强度变化较为剧烈，模型未能完全捕捉到这种快速变化。下图所示的预测数据和真实数据的比对图也存在相同问题。说模型在应对上升幅度较大或者波动幅度较大的 RI 时，性能表现不佳。模型对于极端天气下的台风强度预报仍存在较大的提升空间，其可能的原因是这些热带气旋的强度可能更多地受到热带气旋内部动力的影响，而不是受热带气旋与环境相互作用的影响。

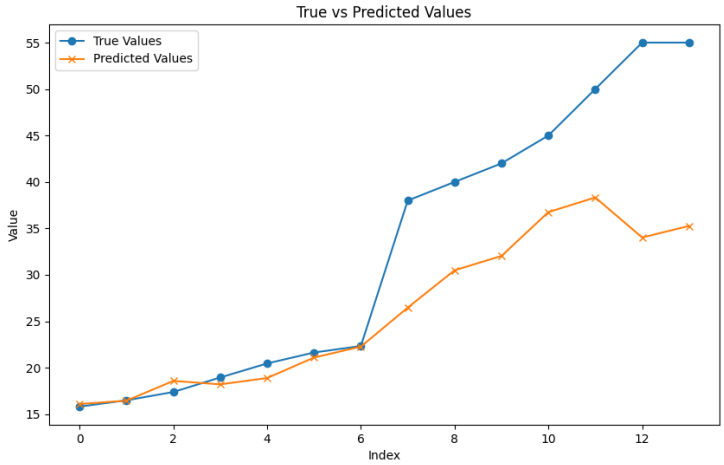

图 4.9 201402C 台风强度预测数据与真实数据对比图

综合来看，一方面强度的预测受到一定的路径预测的干扰，以及模型对于极端天气以及变化幅度较大的情况的预测能力的欠缺，所以在强度预测方面仍有待进一步的提升。

4.3 基于预测路径和强度综合的台风相似度的讨论

本文创新性地使用了台风相似度衡量的量化方法，结合台风强度和路径，给出了预测台风和历史台风的相似度比率。相似度越接近 100% 说明预测值越接近真实值，即，台风预测模型的性能更加优良。图 4.10 展示了台风相似度的随训练周期数（时间）的变化曲线（每个训练周期结束后取所有台风相似度的平均值）：

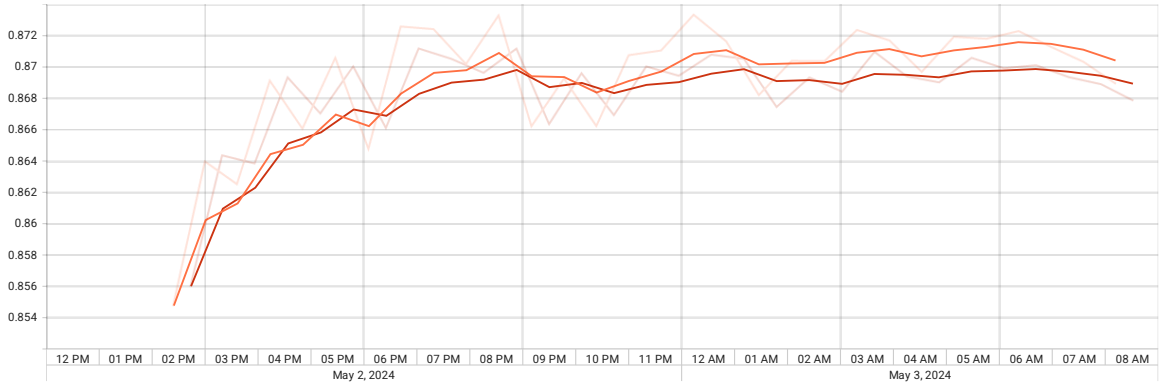

图 4.10 台风相似度的随训练周期数（时间）的变化曲线

台风相似度的计算方法见理论部分（2.3 台风路径相似性评估的定量方法介绍）。由图 4.10 可以看出，在 12 AM 之后，台风相似度的数值趋于稳定（达到 87% 以上），说明模型在该时间点之后对台风数据的预测性能转向较为稳定的状态。同时，模型在验证集上的相似度与在测试集上相差不大（图 4.10 中颜色较浅的曲线为测试集，颜色较深的曲线为验证集），说明模型没有出现过拟合或欠拟合的现象。

4.3.1 台风相似度指标有效性的验证实验

为了验证实验使用的相似度验证指标是有效的，本文使用一组真实的历史台风数据，根据“理论部分”介绍的相似度量化评估方法计算这几组台风关于时间的序列数据的相似度。

下面给出一个实际的台风相似度计算的实例。与本文目标模型的输出（强度、经度、纬度）不同，为了更好地说明相似度计算的总流程，验证实验中，本文将上文给出的仅有 3 个属性的台风结构替换成拥有更多测量参数的实际台风数据。台风时间序列选择台风 202,018 莫拉夫（Molave） 作为基准时间序列，2019 年或 2020 年发展的台风组成长度大小为 52 的比较时间序列集，其中每个台风依次选择作为比较时间序列。具体而言，2019 年发展的 29 个台风索引为 201901 至 201929，2020 年发展的 23 个台风索 引为 202001 至 202023。

同时，因为距离函数涉及的向量都是 5-D 的，验证实验引入了 n 个系数 f_i (i=1,2,3,..,n) 。其中 n = 5。这五个元素中，经度和纬度是必选项，其他三个元素是可选项，不是固定的，在实际使用中可以灵活调整。根据对台风的经验认识和相关论文的建议，本次实验强调了测地线要素（经纬度）的重要性，而减弱了动力要素（移动速度）的重要性。并且考虑到台风的存在作为天气情况在系统中，气象要素（中心气压和蒲福风级）的权重也大于动力要素，但小于测地线要素。综上，对这些系数进行赋值，如下表 4.1 所示。

表 4.1 系数赋值表

符号（Symbol）	含义（Meaning）	种类（Category）	值（Value）
f_1	Longitude	Geodesic	0.26
f_2	Latitude	Geodesic	0.26
f_3	Central Pressure	Meteorological	0.22
f_4	Expanded Beaufort Scale	Meteorological	0.14
f_5	Move Speed	Dynamical	0.12

表 4.2

台风编号（Typhoon code）	台风索引（Typhoon Index）	台风名字（Typhoon Name）
A	202018	Molave
B	202022	Vamco
C	202007	Higos
D	202004	Hagupit
E	202008	Bavi
F	201909	Lekima

表 4.1 的值代表在“距离”计算中的权重。根据表格选定的值，距离函数的展开形式为：

$$dis(p_1,p_2) = 0.26(Lon_1 - Lon_2)^2 + 0.26(Lat_1 - Lat_2)^2 + 0.22(P_1 - P_2)^2 + 0.14(L_1 - L_2)^2 + 0.12(V_1 - V_2)^2$$

和“理论部分”对台风相似度计算原理的介绍同样，归一化函数的 p 取值为 1.005。

计算基准时间序列与对比时间序列集中各台风的相似距离和相似百分比后，选取表 4.2 中列出的 6 个台风作为代表性台风，绘制在下图中。请注意台风 202018 的基准位置的时间序列，因为台风与台风自身之间的时间序列的相似距离为 0，相似度百分比为 100% ，即两个台风完全相同。相似度归一化的基本原理是，比较时间序列之间的相似距离越短基数越大，该对所占的百分比就越高，因为相似距离是衡量两个时间序列对比的正相关度量，并且归一化函数是严格单调递减的，这在图 4.11 中得 到了证实。图 4.12 使用墨卡托投影绘制了这 6 个台风的图。从地图上可以直观地看出基准台风 A 与台风 B 相似度高达 80%，说明这两个台风最为接近。而具有较长相似距离的北部轨迹台风 D、E 和 F 挤在一起并垂直于基准台风。台风 C 轨迹与基准台风相似但位置偏差较远。本实验代码采用 Python-C 混合风格编程，其中 Python 用于加载和预处理数据，C 语言旨在加速相似距离和百分比的计算。 Python 包 “time” 用于确定每种编程语言的运行时间。最终，以台风 A 的台风时间序列作为基准时间序列，与 2019 年 或 2020 年的所有其他 52 个时间序列组成对比时间序列集，经过 50 次重复实验，Python 端耗时 6.829 ms，C 语言平均耗时 60.730 ms，即瞬间计算出结果。

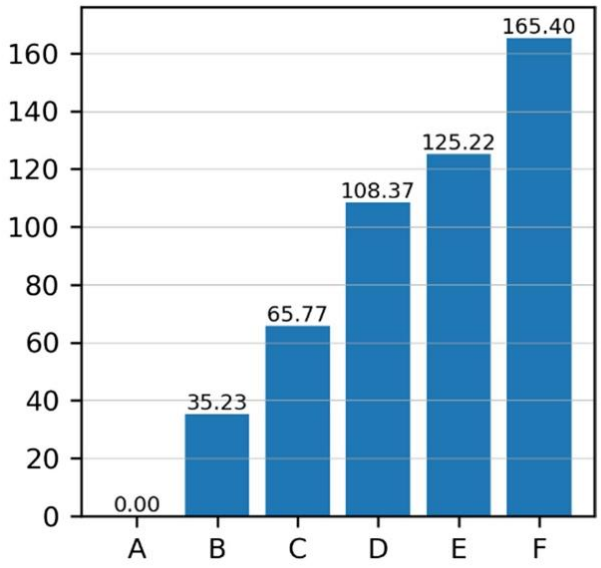

图 4.11 ABCDEF 台风相似度比对图

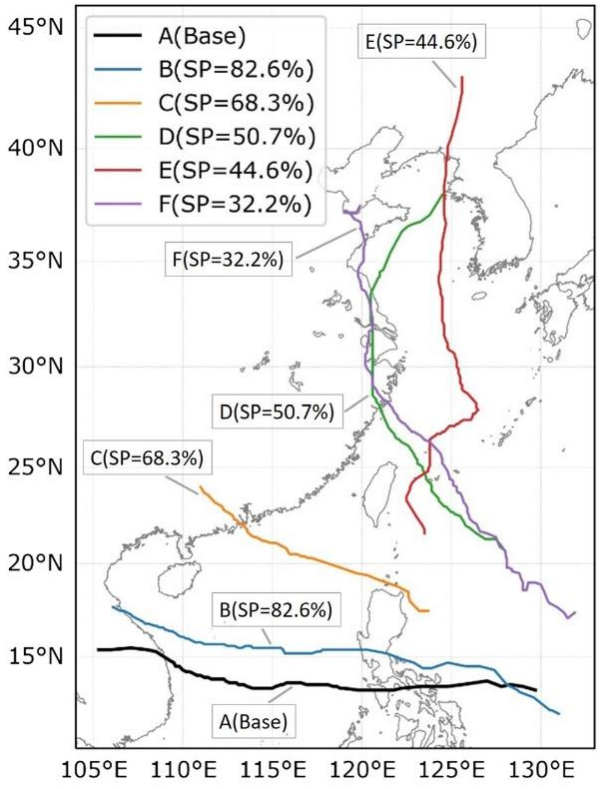

图 4.12 ABCDEF 台风在地图上的投影

装 订 线

在这个实验中有点矛盾的是，台风 E 的相似度比台风 F 更高，而 E 距离 A（基础台风）比 F 更远。原因是除了测地线之外，相似度还会受到其他因素的影响。在对历史台风的相似度评估和验证实验中，本文介绍了不同元素系数的显着性等级并赋予不同的值。这一过程的思想是，某个因素越重要，对该因素施加更显着区分的期望就越高。在不受气象和动力因素影响的情况下，本文进行了重新实验来纠正视觉矛盾。对比如表 4.3 所示，说明可以通过修改系数来定制元素的影响程度。此外，表 1 中提到的 5 个系数可以根据不同的用途赋予不同的值。

表 4.3

台风编号（Typhoon Code）		f_1	f_2	f_3	f_4	f_5	距离（Distance）	百分比（%）
Before	E	0.26	0.26	0.22	0.14	0.12	125.22	44.6
	F						165.40	32.2
After	E	0.5	0.5	0	0	0	151.46	36.2
	F						149.70	36.7

从气象学家的角度来看，气象学上的相似性比地理形态学上的相似性更显着，其分配如表 4.4 所示。选择台风 201909 利奇马作为基准时间序列，并将比较时间序列集扩大到涵盖所有台风 2018 年至 2020 年 （废弃台风除外）。在保持简化参数 p = 1.005 不变的情况下，选取相似距离最短的前 4 个台风（包括基础台风）作为代表性台风，其相似度信息如图 4 所示。

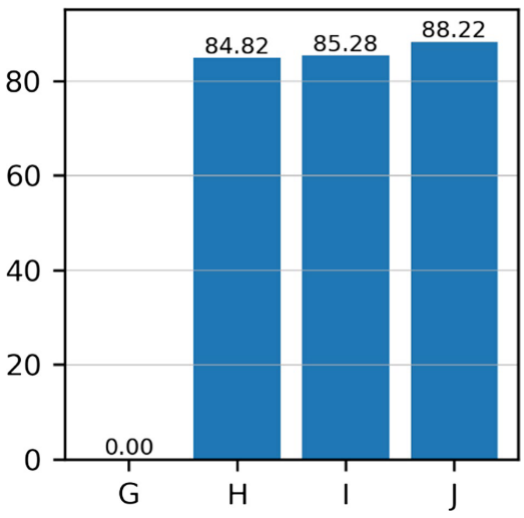

图 4.13 GHIJ 相似度比对图

其中，台风 201909 利奇马编码为“G”、台风 202010 海神编码为“H”、台风 201808 玛丽亚编码为“I”、台风 202009 美萨克编码为“J”。与前一个实验类似，基准台风的时间序列相似距离为零，相似百分比大小保持为 100%。这些台风的路径如图 4.14 所示。 地图上显示了最相似

台风的路径，并标注了它们的代码和相似百分比以及基准台风。缩写“SP” 的意思是“相似度百分比”。

计算参数的取值如下表所示：

表 4.4

符号（Symbol）	含义（Meaning）	种类（Category）	值（Value）
f_1	Longitude	Geodesic	0.16
f_2	Latitude	Geodesic	0.16
f_3	Central Pressure	Meteorological	0.26
f_4	Expanded Beaufort Scale	Meteorological	0.28
f_5	Move Speed	Dynamical	0.14

2019 年 8 月 4 日 17 时（UTC+8，下同），基准台风 G 在菲律宾东部海面生成，随后向西北方向移动，在 24 小时内迅速从强热带风暴增强为超强台风。8 月 10 日 1 时 45 分，它在中国浙江省温岭市沿海登陆，最大风速达 52 米/秒，强度为 16 级，中心最低气压为 930 百帕，登陆后逐渐减弱。2020 年 8 月 28 日 17 时，台风 J 在菲律宾东部海面生成，向西南方向移动。9 月 1 日 5 时，它增强为 2020 年的第一个超强台风，随后转向东偏西北，并开始减弱，于 9 月 3 日约 1 时 30 分在韩国庆尚南道釜山沿海登陆，风速为 42 米/秒，强度为 14 级，中心气压为 950 百帕。2020 年 9 月 1 日 20 时，台风 H 在西北太平洋生成，9 月 4 日 5 时增强为超强台风，并于 9 月 7 日约 7 时 30 分在韩国南部沿海登陆，风速为 40 米/秒，强度为 13 级，中心气压为 955 百帕。2018 年 7 月 4 日 20 时，台风 I 生成，随后其向西北方向移动，并持续增强，最终于 7 月 6 日 5 时增强为超强台风，并于 7 月 11 日 9 时 10 分在中国福建省连江县黄岐半岛沿海登陆，风速为 42 米/秒，强度为 14 级，中心气压为 960 百帕。

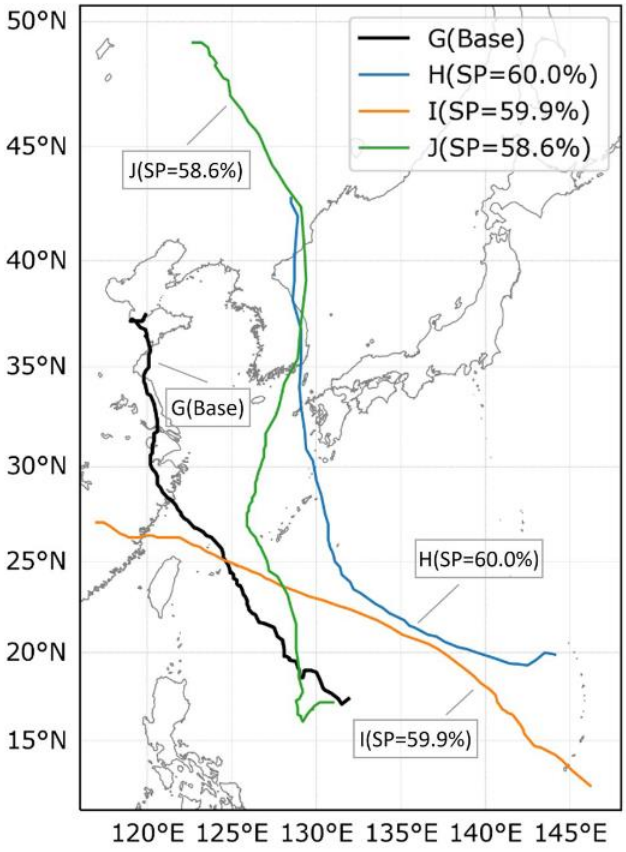

图 4.14 台风路径图

接着本文考察 p 的取值（在上文中取值为 p = 1.005）对相似距离的影响。很明显，与超强台风 201909 号（W）最相似的前三个台风都被分类为超强台风，这证明了新系数设置的普遍性。然而，图 5 中反映出一个不足之处：相似度百分比并没有明显的区分性，即相似度百分比的计算结果之间的差异不够显著，无法清晰地区分出各台风之间的相似度。可以观察到这几个台风相似度的最大差值为 1.4%（60.0%-58.6%），这种情况下，相似度百分比无法提供足够的信息来有效地区分不同台风的相似程度。

由于相似度百分比是通过公式 4.4 计算的，简化参数 p 是影响结果的决定性参数。图 4.15 展示了公式 4.4 在不同简化参数下的情况。下图中横坐标代表相似距离，纵坐标代表相似度。这个公式用于对相似距离的归一化操作。

$$f(x;p) = [1 - t(x;p)] \times 100\% \tag{4.4}$$

容易证明：

$$\forall x > 0, f(x;p_1) < f(x;p_2) \tag{4.5}$$

其中 $p_1 > p_2 > 1$。这个证明可以解释为：随着 p 的增加 $f(x;p)$的图像将向原点 (0, 0) “收缩”。另外， p 的功能是在不同范围内拉伸图像。由于切换简化参数 p 的目的是将相似距离分布在(0, 1] 范围内，结合原实验中(0, 500]范围内的数据，当 p < 1.4 时可以达到显着的效果。

由于学界对台风相似度的定义尚未明确、准确地确定。因此，本文通过实验验证的相似度量化计算方法有助于首先基于动态时间规整算法对两个历史台风之间的相似性进行量化和归一化。实验表明，该方法能够区分台风，并在一定参数规格下筛选出相似度相对较高的台风，当以台风 A 为基准台风时，台风 B 的相似度为 82.6%，台风 C 的相似度为 68.3%，这也验证了该方法的可行性 —— 能够处理各种类型的历史和正在进行的台风时间序列数据，同时保持明显的客观性和准确性。

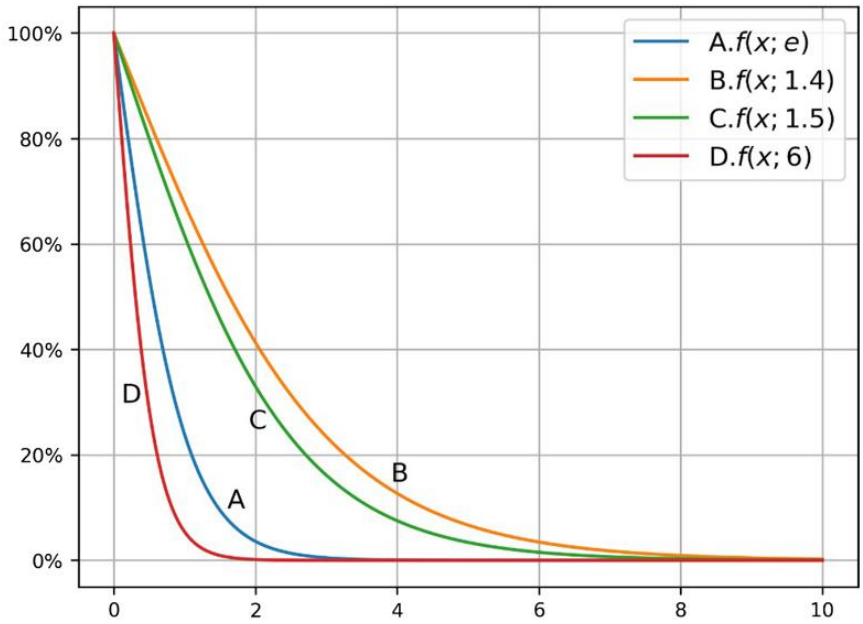

图 4.15 f(x;p) 函数对于不同的 p 的取值的图像

5 结论和展望

5.1 结论

本课题基于多源数据，包括图像数据（3 小时卫星红外 （IR）、水汽（WV）和被动微波降雨率 （PMW）TC 图像）和从 TCIR 数据集收集/导出的季后分析的 TC 相关信息（TC 位置、平移速度、洋盆、到海岸线的距离、当地时间，以及最重要的最佳轨迹和最大风速 Vmax），作为训练数据，集成不同的模型，包括 CNN 和 ConvLSTM 在内，提出了一种新颖的机器学习的模型架构来对台风强度 Vmax 进行预测。创新性地使用了对台风相似度评估的量化方法来对预测台风和历史台风进行相似度方面的评估，最终预测台风和实际台风的相似度能够达到 87%以上，说明模型有着相对较为良好的性能。

同时本文还进行了台风相似度量化方法有效性的验证实验，实验结果表明本文所使用的台风相似度的度量方法（基于动态时间规整算法）有助于对两个历史台风之间的相似性进行量化和归一化。该方法在寻找与正在发生的台风相似的台风以及支持台风预测方面显示出巨大的潜力。在基准实验之后的几个比较实验说明该算法涉及的参数可以根据 用户的需求进行高度定制。然而，在模型预测性能的验证实验中这些参数的权重均设置为大小相同，这里可以根据进一步的研究结果进行调整。即，本文的工作缺乏对物理学上的相似距离的合 理解释以及合理、有目的地分配参数的系统标准。未来需要制定一系列参数建 议，以适应不同环境和领域的使用。

一般来说，传统的 RI （Rapid Intensity）预测利用大气数据的统计方面的预测器。预测变量所承载的高级物理意义使人类能够相应地理解和设计相关性。然而，预测变量中细节的丢失可能是改进预测的瓶颈。本研究通过同时利用传统预测变量和卫星观测中自动提取的特征，探索深度学习用于强度和 RI 预测的可行性。受领域知识启发的处理（即加权损失、CCA 模块和用于对齐剪切向量的旋转） 有助于模型更好地从数据中学习。此外，针对 TC 强度预测提出了一种新颖的深度学习集成方法，因为考虑到触发 RI 的物理过程过于非线性，单一模型可能不是 TC 强度预测的最佳解决方案。所以本研究集成了不同的模型，提出了一种新颖的机器学习的模型架构。

综上所述，本研究的方法和结果对于热带气旋（TC）的监测、预测和应对具有重要的应用价值和广阔的前景：①提升热带气旋监测与预警能力。通过利用多源数据进行热带气旋相关信息的综合分析，可以提升对热带气旋的监测与预警能力。综合考虑卫星图像和 TCIR 数据集中的信息，可以更准确地掌握热带气旋的位置、移动速度、强度等关键特征，为相关部门提供更及时、准确的预警信息。②优化应对策略和资源调配。通过深入分析热带气旋的演变过程和轨迹，可以更好地理解其对不同地区的影响程度和可能带来的灾害类型。这有助于相关部门优化灾害应对策略，合理调配资源，最大程度地减少热带气旋可能造成的损失。

5.2 展望

本实验的结果在一些方面仍有提升的空间，包括：

（1）对比实验的不足：

由于计算资源的限制，本实验相对缺乏与其他预测模型的对比实验。为了更全面地评估所提出的模型的性能，未来可以考虑与其他经典或先进的台风路径预测模型进行比较。这将有助于确定提出模型的相对优势和不足之处，并为进一步改进提供指导。

（2）损失函数的改进：

本实验中，模型的损失函数为强度和位置数据的均方误差（MSE）的简单求和。然而，这两个方面的比率可能会相互干扰，导致某一方面的预测性能表现不够理想。未来的工作可以考虑采用更复杂的损失函数，如加权和或逐步训练的方式，以更好地平衡强度和位置的预测性能。

（3）地理信息的处理：

台风路径的预测对模型精度要求非常高，而地球表面的经度和纬度之间的差异可能导致预测路径与实际路径之间的较大偏差。为了解决这个问题，未来的研究可以考虑引入更精细的地理信息处理方法。例如，可以采用不同地区的台风进行区别处理，或者缩小台风路径预测的地图范围，只针对特定经纬度范围内的台风进行预测。这些方法可以提高预测的精度和准确性，使其更符合实际应用需求。

通过解决上述问题，可以进一步提升所提出模型的性能，并增强其在台风路径预测领域的应用潜力和可靠性。

此外，在台风预测领域（包括强度和路径预测）一些潜在的发展方向可能包括：①跨领域合作与数据共享。未来可以进一步加强与气象、海洋、环境等相关领域的合作与交流，共享数据资源和研究成果。通过跨领域的合作，可以更全面地理解热带气旋的形成和发展机制，提高热带气旋相关信息分析的准确性和可靠性。②引入更多数据源和新技术手段：随着遥感技术和数据采集技术的不断发展，未来可以引入更多的数据源和新的技术手段，如卫星雷达数据、地面观测数据、人工智能算法等。这将进一步丰富热带气旋相关信息的数据来源，提高数据的时空分辨率和精度，为研究提供更多的可能性和机遇。③多学科融合与交叉研究：热带气旋相关信息的分析涉及多个学科领域，如气象学、海洋学、地理学、计算机科学等。未来可以进一步推动多学科的融合与交叉研究，探索更多新的方法和技术手段，以应对气候变化和自然灾害带来的挑战，为人类社会的可持续发展做出更大的贡献。

综上所述，本研究的方法和成果对于提升热带气旋监测与预警能力，优化灾害应对策略，推动多领域合作与创新具有重要意义。未来的发展方向将是跨领域合作、引入新技术和方法、推动多学科融合，为构建更加安全、和谐的人类社会贡献力量。

参考文献

[1] Moradi Kordmahalleh M, Gorji Sefidmazgi M, Homaifar A. A sparse recurrent neural network for trajectory prediction of atlantic hurricanes[C]//Proceedings of the Genetic and Evolutionary Computation Conference 2016. 2016: 957-964.

[2] Alemany S, Beltran J, Perez A, et al. Predicting hurricane trajectories using a recurrent neural network[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 33(01): 468-475.

[3] Liu Y, Racah E, Correa J, et al. Application of deep convolutional neural networks for detecting extreme weather in climate datasets[J]. arXiv preprint arXiv:1605.01156, 2016.

[4] Kim S, Kim H, Lee J, et al. Deep-hurricane-tracker: Tracking and forecasting extreme climate events[C]//2019 IEEE winter conference on applications of computer vision (WACV). IEEE, 2019: 1761-1769.

[5] Di Y, Lu M, Chen M, et al. A quantitative method for the similarity assessment of typhoon tracks[J]. Natural Hazards, 2022: 1-16.

[6] Soboleva E V, Beskorovainyi V V. The utility function in problems of structural optimization of distributed objects[J]. Kharkiv University of Aair Force, 2008, 121.

[7] Bai C Y, Chen B F, Lin H T. Benchmarking tropical cyclone rapid intensification with satellite images and attention-based deep models[C]//Machine Learning and Knowledge Discovery in Databases: Applied Data Science Track: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part IV. Springer International Publishing, 2021: 497-512.

[8] Chen B, Chen B F, Lin H T. Rotation-blended CNNs on a new open dataset for tropical cyclone image-to-intensity regression[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2018: 90-99.

[9] DeMaria M, Mainelli M, Shay L K, et al. Further improvements to the statistical hurricane intensity prediction scheme (SHIPS)[J]. Weather and Forecasting, 2005, 20(4): 531-543.

[10] BAI C Y, CHEN B F, LIN H M. Benchmarking Tropical Cyclone Rapid Intensification with Satellite Images and Attention-based Deep Models[J]. Cornell University - arXiv,Cornell University - arXiv, 2019.

[11] Chen B F, Davis C A, Kuo Y H. Examination of the combined effect of deep-layer vertical shear direction and lower-tropospheric mean flow on tropical cyclone intensity and size based on the ERA5 reanalysis[J]. Monthly Weather Review, 2021, 149(12): 4057-4076.

[12] IOFFE S, SZEGEDY C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift[J]. arXiv: Learning,arXiv: Learning, 2015.

[13] OLANDER T, WIMMERS A, VELDEN C, et al. Investigation of Machine Learning using Satellite-Based Advanced Dvorak Technique Analysis Parameters to Estimate Tropical Cyclone Intensity[J/OL]. Weather and Forecasting, 2021, 36(6): 2161-2186. http://dx.doi.org/10.1175/waf-d-20-

0234.1. DOI:10.1175/waf-d-20-0234.1.

[14] OLANDER T L, VELDEN C S. The Advanced Dvorak Technique (ADT) for Estimating Tropical Cyclone Intensity: Update and New Capabilities[J/OL]. Weather and Forecasting, 2019, 34(4): 905-922. http://dx.doi.org/10.1175/waf-d-19-0007.1. DOI:10.1175/waf-d-19-0007.1.

[15] Xu G, Xian D, Fournier-Viger P, et al. AM-ConvGRU: a spatio-temporal model for typhoon path prediction[J]. Neural Computing and Applications, 2022, 34(8): 5905-5921.

[16] Qin W, Tang J, Lu C, et al. A typhoon trajectory prediction model based on multimodal and multitask learning[J]. Applied Soft Computing, 2022, 122: 108804.

[17] Wei C C. Collapse warning system using LSTM neural networks for construction disaster prevention in extreme wind weather[J]. Journal of Civil Engineering and Management, 2021, 27(4): 230-245.

[18] Wang C, Li X, Zheng G. Tropical cyclone intensity forecasting using model knowledge guided deep learning model[J]. Environmental Research Letters, 2024, 19(2): 024006.

[19] Li S, Lu L, Hu W, et al. Prediction Algorithm of Wind Waterlogging Disaster in Distribution Network Based on Multi-Source Data Fusion[J]. Mathematical Problems in Engineering, 2022, 2022.

[20] Gao S, Zhao P, Pan B, et al. A nowcasting model for the prediction of typhoon tracks based on a long short term memory neural network[J]. Acta Oceanologica Sinica, 2018, 37: 8-12.

[21] 周冠博，钱奇峰，吕心艳，等. 人工智能新技术在国家气象中心台风业务中的 应用探索[J]. Journal of Tropical Meteorology (1004-4965), 2022, 38(4).

[22] 周冠博，钱奇峰，吕心艳. 人工智能在台风监测和预报中的探索与展望[J]. 气象研究与应用，2022，43(2)：1-8.

[23] 郑小平,李崇银.台风动力模式的发展与应用[J]. 气象科技进展，2014，4(2)：1-9.

[24] 刘敏.基于改进 LSTM 模型的时空序列台风图像预测方法研究[D]. 上海海洋大学，2021.

[25] 徐光宁.基于深度学习的台风路径与强度预测方法研究[D]. 南京理工大学，2020.

[26] 张晓东,李崇银.基于神经网络的台风路径预测模型[J]. 气象科技，2006，34(5)：581-585.

装 订 线

谢 辞

在我即将完成学业、顺利毕业之际，我怀着无比感激的心情向在我整个学习和研究过程中给予我帮助和支持的每一位朋友及家人表示诚挚的谢意。

首先，我要感谢我的家人。父母的无私支持和默默付出是我不断前进的动力。无论是在学习上的鼓励还是生活上的照顾，他们始终是我最坚强的后盾。特别是在我遇到困难和挫折时，家人的关心和支持使我重新振作，继续前行。没有他们的理解和支持，我不可能走到今天。

同时，我要感谢我的朋友们。在这几年的学习生活中，大家互相帮助、共同进步。特别是在论文写作过程中，与朋友们的讨论和交流使我获得了许多宝贵的意见和建议，为我的论文增添了不少亮点。感谢你们在我困惑时给予的指点，在我疲惫时给予的鼓励，以及在我取得成绩时共同分享的喜悦。

此外，我要感谢那些在我求学之路上默默帮助过我的人。无论是图书馆的工作人员、实验室的技术人员，还是其他在我学习过程中提供帮助和支持的人士，你们的无私付出让我倍感温暖。正是因为有了你们的帮助，我才能够顺利完成学业。

感恩之情难以言表，这篇毕业论文的完成不仅是我学术生涯的重要里程碑，更是大家关心、支持和帮助的结果。在未来的日子里，我将带着这份感恩之心，继续努力，不辜负大家的期望。

装

订

线