remote sensing

MDPI

Article

An Adaptive Learning Approach for Tropical Cyclone Intensity Correction

Rui Chen [1,2,3,†], Ralf Toumi [3,†], Xinjie Shi [1,2], Xiang Wang [2], Yao Duan [1] and Weimin Zhang [2,*]

[1] College of Computer Science and Technology, National University of Defense Technology, Changsha 410073, China; r.chen21@imperial.ac.uk (R.C.); shixinjie19@nudt.edu.cn (X.S.); duanyao16@nudt.edu.cn (Y.D.)
[2] College of Meteorology and Oceanography, National University of Defense Technology, Changsha 410073, China; xiangwangcn@nudt.edu.cn
[3] Department of Physics, Imperial College London, London SW7 2AZ, UK; r.toumi@imperial.ac.uk
[*] Correspondence: weiminzhang@nudt.edu.cn; Tel.: +86-139-7492-5867
[†] These authors contributed equally to this work.

Abstract: Tropical cyclones (TCs) are dangerous weather events; accurate monitoring and forecasting can provide significant early warning to reduce loss of life and property. However, the study of tropical cyclone intensity remains challenging, both in terms of theory and forecasting. ERA5 reanalysis is a benchmark data set for tropical cyclone studies, yet the maximum wind speed error is very large (68 kts) and is still 19 kts after simple linear correction, even in the better sampled North Atlantic. Here, we develop an adaptive learning approach to correct the intensity in the ERA5 reanalysis, by optimising the inputs to overcome the problems caused by the poor data quality and updating the features to improve the generalisability of the deep learning-based model. Specifically, we use understanding of TC properties to increase the representativeness of the inputs so that the general features can be learned with deep neural networks in the sample space, and then use domain adaptation to update the general features from the known domain with historical storms to the specific features for the unknown domain of new storms. This approach can reduce the error to only 6 kts which is within the uncertainty of the best track data in the international best track archive for climate stewardship (IBTrACS) in the North Atlantic. The method may have wide applicability, such as when extending it to the correction of intensity estimation from satellite imagery and intensity prediction from dynamical models.

Keywords: tropical cyclones; ERA5 reanalysis; deep learning; generalisability; domain adaptation

check for updates

Citation: Chen, R.; Toumi, R.; Shi, X.; Wang, X.; Duan, Y.; Zhang, W. An Adaptive Learning Approach for Tropical Cyclone Intensity Correction. *Remote Sens.* **2023**, *15*, 5341. https://doi.org/10.3390/rs15225341

Academic Editor: Yuriy Kuleshov

Received: 13 October 2023
Revised: 5 November 2023
Accepted: 9 November 2023
Published: 13 November 2023

1. Introduction

Tropical cyclones (TCs) cause enormous damage around the world every year, especially in coastal areas [1–3]. Many scientists are contributing to finding the regularity of the genesis, development, and disappearance of tropical cyclones from the past to the present [4,5]. This regularity is also the cornerstone of forecasting techniques, for providing more accurate early warnings to protect people and property. Over the last century, observational technology, dynamical theory, and forecasting techniques have made great progress, but there are still many key problems to be solved. Some typical problems are related to tropical cyclone intensity. For example, the theory of intensity change is incomplete, intensity observations are scarce, and intensity prediction is difficult [6,7].

Definitions of tropical cyclone intensity vary from agency to agency. Generally, it is defined as the maximum sustained wind speeds at the standard meteorological height of 10 m near the centre of the storm or the minimum central pressure [8]. Agencies collect as many historical records as possible and then re-analyse them to provide a standard reference for future researchers. Usually, in situ observations of tropical cyclones are very difficult to collect, so most data comes from satellite observations, with very little from aircraft reconnaissance [9–11]. Tropical cyclone reanalysis is integrated into a well-known dataset known as the best track dataset, like the international best track archive for climate stewardship (IBTrACS). This dataset also provides a reference for the development of tropical cyclone monitoring and forecasting techniques. The Dvorak technique quantitatively estimates intensity from satellite imagery [12], so it is still widely used by operational agencies and has been continuously updated [13,14]. The most representative methods for intensity forecasting are the statistical model [15–18] and the dynamical forecasting model [19]. In most cases, they need to be combined to produce a more accurate forecast, such as the statistical-dynamical model [20].

With the advent of artificial intelligence in recent years, many new studies have attempted to explore different intelligent techniques to optimise or replace existing monitoring or forecasting techniques [21]. For example, they use convolutional neural networks (CNN) in image recognition to update the Dvorak technique [22–26], or non-linear deep neural networks to replace the traditional statistical model for intensity prediction [20,27–30]. An approach combining the predictors of the statistical hurricane intensity prediction scheme (SHIPS) and the multi-layer perceptron (MLP) for intensity prediction is presented in [27], which outperforms the statistical-dynamical models and achieves comparable results to the observation-adjusted Hurricane Weather Research and Forecasting (HWFI) in the Atlantic basin. Such machine learning methods have been used for the first time in operational forecasting at the National Hurricane Center (NHC) in 2022 [20], and the model is Neural Network Intensity Consensus (NNIC). In addition, our previous work in 2019 [28] considers intensity prediction as a spatio-temporal sequence and then develops a hybrid model that fuses CNN and Long Short-Term Memory (LSTM) to predict it using atmospheric and oceanic variables from the ERA-Interim reanalysis. This type of method has been updated and optimised over the last two years by [29,30] with a novel feature extraction and aggregation module using Transformer.

There are perhaps two reasons why artificial intelligence methods can be widely used in tropical cyclone research. The first is that these kinds of data-driven methods are out of the existing physical theory, so they are expected to provide new insight to find unknown knowledge. The second is that the strong representative ability of deep neural networks is shown to be a powerful tool to fit a specific pattern hidden in the data, which is to a large extent better than traditional methods such as linear regression. However, these types of methods are highly dependent on the quality of the data. This means that it may be impossible to learn the correct knowledge or representation if the intelligent models are trained on poor-quality data. In reality, it is also very difficult to assess the quality of data, especially for tropical cyclones, which are a type of suddenly changing weather phenomenon without comprehensive observations.

Another problem is the model generalisability; the deep neural network is verified with a strong representative ability, but the understanding ability is still questionable. It is because of the basic assumption of machine learning; the training dataset used to train the model and the testing dataset used to verify the model generalisability should be independent and identically distributed [31,32]. This means that the sample size of the training dataset should be large enough, and the hidden information in the samples should be fully representative of the entire data space. The testing dataset can then be used as a measure to test whether the model is capable of understanding or strong generalisability. So far, models trained on the well-known ImageNet dataset [33,34] may be close to the above assumptions. However, in practical applications, especially for tropical cyclones, the basic assumptions are far from being achieved. This is because the learning task is largely limited by the sample size and the valuable data with less noise.

The emergence of transfer learning [35] may provide a great opportunity to address the above issues [36], and there are several cases of transfer learning being used in tropical cyclone research. In 2017, transfer learning was mentioned to improve the generalisability of the Indian Ocean-trained model for better intensity prediction performance in the South Pacific [37]. Similar works on intensity estimation and real-time TC detection were presented in [23,36,38]. More recently, two works refer to the transfer learning used to generate tropical cyclone data. One is by [39], which fine-tunes the pre-trained model in the North Atlantic with data from the Western North Pacific to reconstruct a historical size dataset in that basin. The other pre-trains the model using climate model simulations and fine-tunes them by reanalysis for data generation to emulate seasonal tropical cyclone activity [40].

In addition, the ERA5 reanalysis (ECMWF Reanalysis v5) provides an optimised estimate of the current atmospheric state, which includes a detailed description of tropical cyclones [41]. It may not be the most accurate representation of tropical cyclone intensity compared to satellite imagery or in situ observations, but it contains comprehensive environmental information that can be an effective supplement to tropical cyclone monitoring and forecasting [42–44]. The data format is similar to the outputs of dynamical models, so it can be a benchmark dataset to develop new methods for later applications. There have been some studies exploring the capability of tropical cyclone representations in the ERA5 reanalysis [45–47].

Here, we aim to correct the intensity in the ERA5 reanalysis using the true intensity in IBTrACS as a reference. There are some related works to correct the intensity in operational forecasting [48–50]. Differently, we develop an adaptive learning approach based on deep neural networks and domain adaptation, which helps to solve the issues of data quality and weak model generalisability. Experiments in the North Atlantic verify the effectiveness of our approach. It can be easily extended to other learning tasks using different data sets, such as satellite images or the outputs of dynamical models. Furthermore, it is not limited to the same computing platform, deep learning framework, and Python version, which can be easily developed by other researchers.

This paper is organised as follows: Section 1 gives a brief description of the background, related works, and our work. Section 2 presents the learning tasks, the concept of our approach, and the basic knowledge and experimental setup for implementing this approach. Section 3 shows the preliminary data evaluation and analysis of the experimental dataset, and how to optimise the results from the baseline using our approach. We also express our attempts, unsolved problems, and future plans in Section 4. Finally, we summarise the whole work, including the progress and potential in Section 5.

2. Methodology

2.1. Our Approach

Our learning tasks can be defined as follows:

$$Y \sim F(X). \tag{1}$$

Our goal is to find the optimal F with strong generalisability. X are the inputs, Y are the outputs, and $F(\cdot)$ is the mapping to be learned between inputs and outputs. Considering that the correspondence between X and Y hidden in the data space is not clear, the difficulty of learning F may increase due to poor data quality.

Here, we develop an adaptive approach to learning it, and the methods can be divided into three steps, either incrementally or individually. The first is to learn F from X to Y directly, and Expression (3) or the brown path in Figure 1 denotes this step. If this works, the goal is reached. The loop is decided by the decision block Q, which represents whether the final goal is achieved or not. If not, we choose the second step, which is to update the inputs to $X^{'}$ and then learn the mapping F from $X^{'}$ to Y. We define $F = f_1 \circ f_2$, where f_1 is the feature extractor from inputs $X^{'}$ to features χ, and f_2 is the mapping from features χ to outputs Y. Thus, it can also be formulated by Expression (4) or represented by the blue path in Figure 1. If it still does not work, we will go to the next step to update f_2 to $f_2^{'}$ or update the general features χ to $\chi^{'}$ as the specific features until the goal is achieved. This step is described in Expression (5) and shown by the red path in Figure 1. So now

$F = f_1 \circ f_2'$ or $F = f_1 \circ f_{11}{}' \circ f_{12}{}'$. In this step, $f_{11}{}'$ is the feature mapping from χ to χ' and $f_{12}{}'$ is the new mapping from the updated features χ' to Y.

In fact, any path from X or X' to Y shown in Figure 1 below is available and can be chosen in different learning tasks if it works. We present and describe only three of the above paths that were used in the next experiments. Q indicates if the intensity error is less than or equal to the uncertainty of the best back data in this paper.

$$X \xrightarrow{F} Y, \tag{2}$$

$$X' \xrightarrow{f_1} \chi \xrightarrow{f_2} Y, \tag{3}$$

$$X' \xrightarrow{f_1} \chi \xrightarrow{f_2'} Y \quad or \quad X' \xrightarrow{f_1} \chi \xrightarrow{f_{11}'} \chi' \xrightarrow{f_{12}'} Y. \tag{4}$$

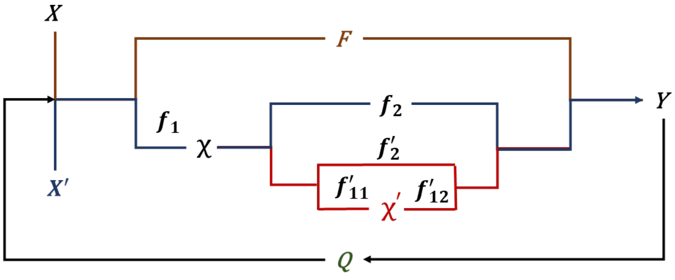


Figure 1. This is the flowchart of our adaptive approach. The loop is decided by the decision block Q, which indicates whether the final goal has been reached or not. The brown path from X to Y represents the first step, which is to learn F directly. If Q returns negative feedback, it turns to the second step, which is to optimise the inputs from X to X', and the learning process can be described by the blue path from X' to χ to Y. If the final goal is still not achieved, Q will guide the learning approach to the third step, which is updating the features and the feature-output mapping, represented by the red line. The learning process that updates the features and the feature-output mapping is from X' to χ to χ' to Y.

2.2. Basic Descriptions

2.2.1. Data

In this paper, Y is the intensity label from the best track dataset—IBTrACS. IBTrACS collects the global best track dataset and provides the uncertainty estimate for intensity in maximum wind speed [16]. From 2004 to now, the uncertainty of intensity in the North Atlantic is 7 kts (knots: 1 knot = 0.518 m/s), while in other basins it is 10 kts [51]. It provides the storm centre, time, and other attributes we need, and is widely used as a reference for developing tropical cyclone techniques. The data from the US agencies tends to be of the highest quality. Table 1 below shows the basic information collected from the IBTrACS technical report (https://www.ncei.noaa.gov/sites/default/files/2021-07/IBTrACS_version4_Technical_Details.pdf, accessed on 5 October 2023).

Table 1. This is the data description of IBTrACS.

Variable name (units)	Maximum Sustained Wind Speed (kts) Storm Center (degrees lat/lon) Other variables
Temporal resolution	Interpolated to 3 hourly (Most data reported at 6 hourly)
Coverage	70°N to 70°S and 180°W to 180°E 1841—present (Not all storms captured)

X denotes inputs from the ERA5 reanalysis. ERA5 provides the global atmospheric state with a latency of about 5 days and is available from 1940 to the present. The spatial resolution is 0.25° and the temporal resolution is one hour. In addition, it is a homogeneous and consistent gridded data set with a large number of atmospheric, ocean wave, and land surface variables. Table 2 presents the basic information of the ERA5 hourly data on pressure levels collected from the official website (https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-pressure-levels?tab=overview, accessed on 5 October 2023).

Table 2. This is the data description of ERA5 reanalysis.

Data Type	Gridded
Horizontal coverage Horizontal resolution	Global 0.25° × 0.25°
Vertical coverage Vertical resolution	1000 hPa to 1 hPa 37 pressure levels
Temporal coverage Temporal resolution	1940 to present Hourly

2.2.2. Deep Neural Networks

Given the strong representative ability of deep neural networks, we use deep learning methods to learn F. The methods start from MLP with nonlinear activation functions, which can be used to effectively fit the nonlinear mapping or function. The reason for using them is inspired by the assumption of deep neural networks that, given enough data, they can be used to estimate any function. Convolutional neural networks reduce network parameters by sharing parameters in the receptive field, which is widely used in image tasks and is now achieving great success. It can effectively extract the spatial features of the image and can therefore be used as a general feature extractor for practical applications. However, the deeper the network, the greater the vanishing gradient problems. ResNet [52] is designed to use a short-cut way to solve the vanishing gradient problem, but also keep the strong representative ability of deep neural networks. It has been widely used in computer vision and is also the foundation stone of many industrial applications. The core module of ResNet is the Residual Block. Here, we only use the simplest version, ResNet-18. Given the inputs, after feature extraction of residual blocks with convolutional layers, there is an average pooling layer to reduce the dimensionality of the features to 512, and then the features are converted to the output layer to complete our task.

2.2.3. Transfer Learning

Given the problems mentioned in the Section 1, transfer learning can be a good way to improve the generalisability of the model. The simplest method is fine-tuning, which can be used to help the trained model adapt to new samples. Although it is able to quickly retrain the model to improve accuracy, it still does not solve the problems of different data distribution between the training and testing datasets. Therefore, the idea of domain adaptation (DA) is also referred to in this paper. It is designed to help a model trained in a known source domain to adapt to an unknown target domain, which is similar to our problems. The core idea of domain adaptation is to find the similarity of two domains and try to reduce the distance between two data sets by the general distance function. One of the key problems is to define a new loss function that merges the distance between the source and target domains. Maximum mean discrepancy (MMD) is the most widely used metric in transfer learning, especially in domain adaptation.

2.3. Experimental Setting

2.3.1. Dataset

In order to carry out experiments, the first step is to prepare the data set. We select the samples from the following rules:

- Data are post-reanalysed by agencies, and it means that 'TRACK_TYPE' is flagged as the 'main';

- Only tropical cyclones ('NATURE' is marked as 'TS') are analysed, and 'USA_SSHS' (Saffir-Simpson Hurricane Scale) is larger than 0. This means that the wind speed provided by the US agencies is ≥64 kts;
- Records from 2004 to 2022, only in the North Atlantic, and they are provided by US agencies.

The description of TC samples used for experiments can be found in Table 3. We download the one-to-one matching ERA5 data with each TC sample (central location and time) using the Python API by adjusting specific variables, levels, region size, etc. In detail, we follow our previous work in 2019 [28] but adapt it for this learning task and then set new download rules. They are described below:

- Atmospheric variables: u (u-component of wind), v (v-component of wind), t (temperature), and r (relative humidity);
- Pressure levels: 850 hPa/500 hPa/200 hPa;
- Surface variables: u_{10} (10 m u-component of wind), v_{10} (10 m v-component of wind);
- Region size: 20° × 20°, and the spatial resolution is 0.25° × 0.25°.

Table 3. These are the North Atlantic samples collected in IBTrACS between 2004 and 2020.

	TC Numbers	Samples
Category 1 ($64 \leq W^* < 83$)	32	2061
Category 2 ($83 \leq W < 96$)	13	774
Category 3 ($96 \leq W < 113$)	20	626
Category 4 ($113 \leq W < 137$)	29	562
Category 5 ($W \geq 137$)	14	122
Total	108	4145

* W denotes the 1-min wind speeds (kts) provided by the US agencies.

The experimental dataset therefore consists of two parts. One is the inputs' dataset from the ERA5 reanalysis, and the other is the labels dataset from IBTrACS. The attributes and shape of the inputs vary from experiment to experiment, and we will clarify them in each experiment below. But, the label is always a single value of intensity from the US agencies, labeled 'USA_WIND' in IBTrACS, and we also refer to it as 'IBTrACS_Vmax' in this paper.

We also need to split the dataset to train the model, select the model, and evaluate the model. They are the training dataset, validation dataset, and testing dataset, separately. There are several splitting methods in machine learning, such as hold-out with stratified sampling, cross-validation, and bootstrapping [32]. These splitting methods are based on a basic assumption, which is that the training and testing datasets are drawn from the same data distribution. The testing dataset is used to test whether the model is capable of learning the knowledge from the sample space. For our research problem of intensity correction, we need to apply the trained model in the real scene. Therefore, we split the dataset according to the consecutive years. We try to use the new storms to test the generalisability of the model trained on the historical storms. Therefore, we compare three splitting methods presented here. The first one uses the hold-out method in machine learning; we use 80% for training, 10% for validation, and 10% for testing. The second one uses the 2021–2022 samples as the testing dataset, and the rest is divided into a training (90% of the samples in 2004–2020) and validation dataset (10% of the samples in 2004–2020). The third divides the data strictly by years. Thus, we use the samples from 2004–2018 for training, 2019–2020 for validation, and 2021–2022 for testing.

2.3.2. Objective Function

The objective function of the training process we use here is the mean square error (MSE), and the formula is defined as follows:

$$Loss = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2. \tag{5}$$

N is the size of the samples, y_i is the label of the ith sample, and $\hat{y}_i$ is the ith output of the network.

2.3.3. Evaluation Metrics

The metrics we choose here are bias and root mean square error (RMSE). The former is used to evaluate the accuracy of the model, and the latter evaluates the stability of the model. They are also the key metrics to evaluate the performance of tropical cyclone forecasting and are formulated as follows:

$$Bias = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i), \tag{6}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}. \tag{7}$$

Also, N is the size of the samples, y_i is the label of the ith sample, and $\hat{y}_i$ is the ith output of the network.

We perform all comparative experiments using the same computational environment. The module we use in this paper is Python 3.8, Keras 2.8.0, Tensorflow 2.8.0, Scikit-learn 1.3.0, Numpy 1.24.4, Pandas 2.0.3, MetPy 1.5.1, and so on. We also use the TESLA-V100 GPU to improve computational efficiency. We set all random seeds in the experiments to 42 to reduce the influence of randomness.

3. Results

3.1. Data Analysis

3.1.1. Original Information

To evaluate the hidden correlation in the original dataset, we analyse it using statistical methods. Unlike experimental datasets in image classification or object recognition, the relationship between inputs and outputs is certain and obvious. Our dataset contains the samples recorded in IBTrACS; only some samples are filtered according to our specific task in tropical cyclones, and are not specifically selected for machine learning tasks. Therefore, the correspondence between inputs and outputs needs to be explored.

One of the definitions of intensity is the 10 m maximum wind speed near the storm centre; thus, we can calculate intensity in the ERA5 reanalysis using surface variables (u_{10} and v_{10}). Specifically, the wind speed is the square root of the sum of the squares of the u and v components and is also represented as 'Wind' in the following sections and tables. We also need to convert the units (m/s) of wind speed in ERA5 to kts in IBTrACS. We call the reanalysis intensity ERA5_Vmax and the best track intensity IBTrACS_Vmax.

The total bias of the full dataset of 4145 samples between 2004 and 2022 is −43.08 kts and the RMSE is 47.41 kts. The scatter plot in Figure 2 describes the correlation between ERA5_Vmax and IBTrACS_Vmax. We can observe a moderate correlation between these two variables, which is also confirmed by the Pearson correlation coefficient (r), r = 0.4. It is noticeable that there are obvious one-to-many and many-to-one relationships between ERA_Vmax and IBTrACS_Vmax. For example, the Vmax in IBTrACS is 100 kts, but the possible values in ERA range from about 22 kts to 70 kts. We also plot the cumulative curve of the RMSE, as shown in Figure 3 and we observe that the minimum error is close to 20 kts and the maximum error is about 90 kts. The error distribution is comparatively balanced in different ranges and the RMSE of nearly 80% samples is less than 55 kts.

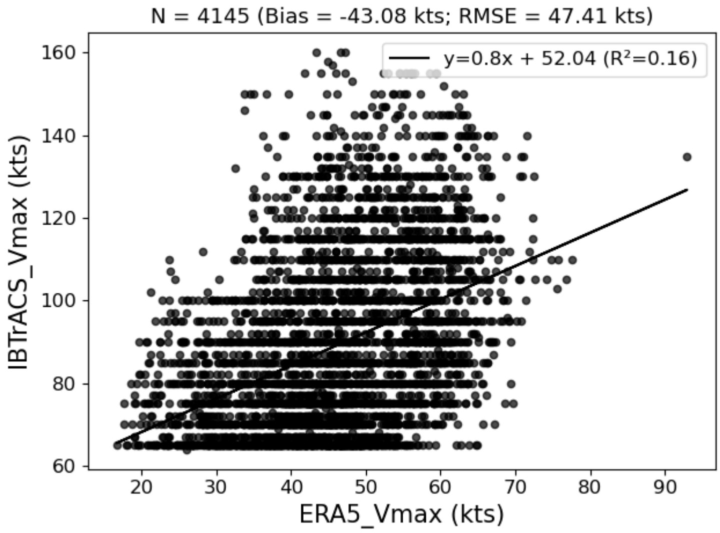


Figure 2. This is a scatter plot showing the correlation between ERA5_Vmax and IBTrACS_Vmax in the full dataset (2004–2022) with a sample size (N) of 4145. The bias and RMSE of the dataset are −43.08 kts and 47.41 kts, respectively. The Pearson correlation coefficient (r) of these two intensity values is 0.4 (0.3 < r < 0.5), indicating a moderate positive correlation. The solid line shows the linear fit with R^2 = 0.16.

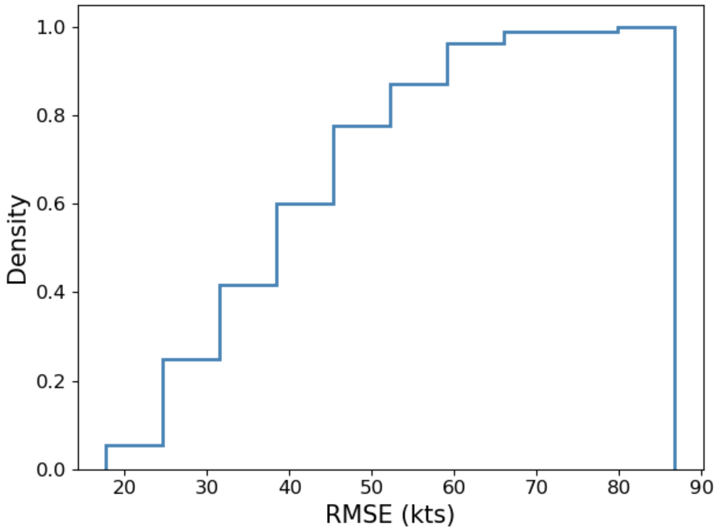
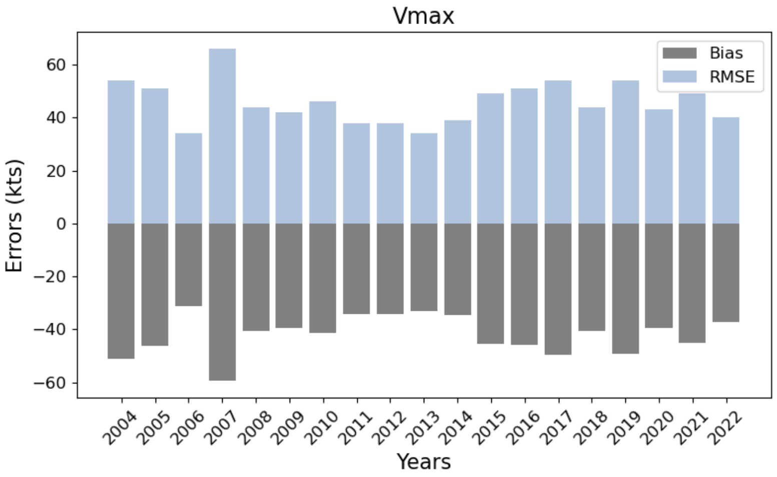


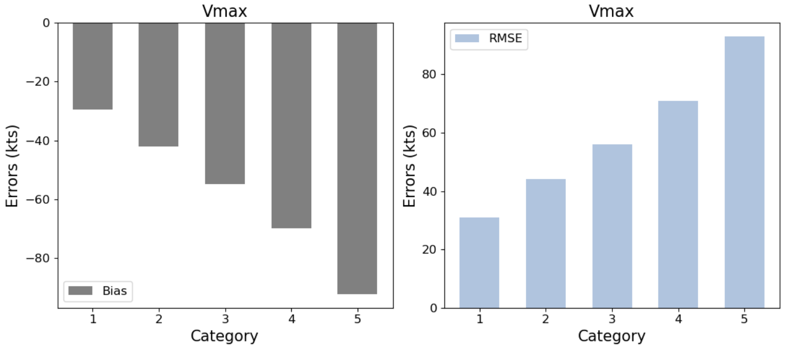
Figure 3. This is a figure showing the cumulative curve of RMSE of the full dataset (2004–2022).

3.1.2. Error Analysis

To analyse the factors related to the error distribution, we divide the RMSE of all samples into different groups. Here, we only consider the different years and categories. From Figure 4a we can observe the variation from 2004 to 2022, but there is no obvious trend with the years, whether for the bias or the RMSE. During these years, the average RMSE has a minimum in 2006 of about 30 kts, while it increases to a maximum in 2007 of about 70 kts. The RMSE of the remaining years fluctuates with an average value of about 40 kts. From Figure 4b, we can observe an obvious increasing trend in the errors as the category of storms grows. The average RMSE of the Category 1 samples is about 30 kts, but 90 kts for the Category 5 samples. This means that the average Vmax of different categories in the ERA reanalysis makes a small difference.



(**a**) The Bias and RMSE distributed in different years.



(**b**) The Bias and RMSE distributed in different categories.

Figure 4. This is a figure showing the error distribution for different factors. (**a**) The bias and RMSE distributed in different years. (**b**) The Bias and RMSE distributed in different categories.

3.1.3. Storms' Correspondence

Apart from the errors in the full dataset, we all try to check the correspondence between the storms contained in ERA5 and IBTrACS. We use the life maximum intensity (LMI) to demonstrate the storm characteristics. We show the scatter plot in Figure 5 to describe the relationship between the LMI of storms in IBTrACS (IBTrACS_LMI) and storms in ERA5 (ERA5_LMI). This shows an increasing trend as the number of categories increases. Here, the category indicates the type of storm, not the samples. For example, the Category 3 storms show that the LMI of this type of storm is in the range of [96, 113]. There are 20 Category 3 storms in this dataset. We also rank the storms by LMI in ERA5 and IBTrACS and calculate the overlap rate in different categories shown in Table 4. We can observe that the overlap rate of the top 10% is 0.5 only for Category 3 storms, and the rest is 0. As for other categories, such as Category 1 and Category 5, they all show a weak correspondence. The overlap rate of the top 50% is only 0.57 for Category 5 storms.

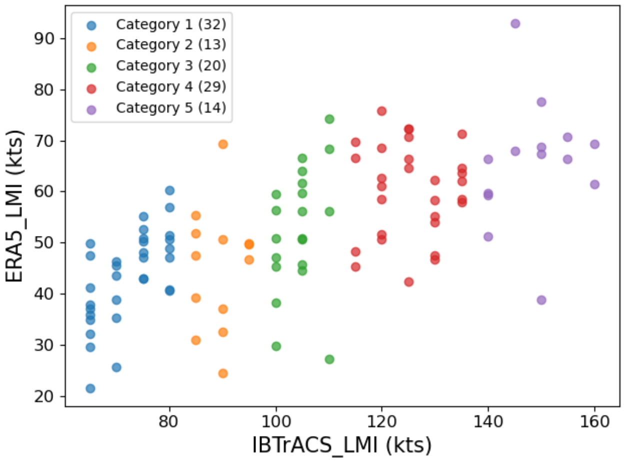


Figure 5. This is a figure showing the storm correspondence between ERA5 and IBTrACS, by comparing the life maximum intensity (LMI) of different storms in five categories.

Table 4. This is a table describing the overlap rate of different category storms between ERA5 and IBTrACS.

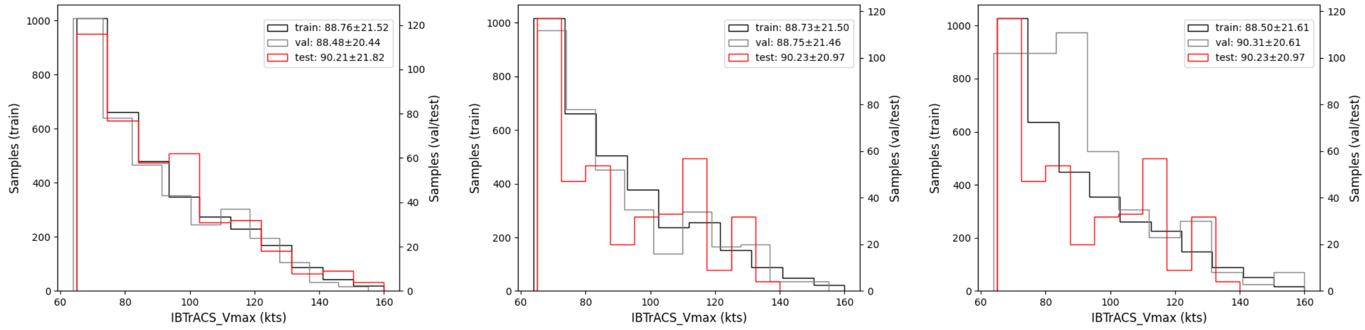
	10%	**20%**	**30%**	**40%**	**50%**
Category 1	0.00	0.33	0.60	0.77	0.75
Category 2	0.00	0.00	0.25	0.40	0.50
Category 3	0.50	0.50	0.33	0.50	0.70
Category 4	0.00	0.17	0.11	0.25	0.43
Category 5	0.00	0.33	0.50	0.33	0.57

3.2. Our Adaptive Approach

3.2.1. Baseline

After a preliminary evaluation of the hidden correlation between the intensity value in ERA5 and IBTrACS, here, we use our approach to correct the intensity in ERA5 to be close to the intensity in IBTrACS. To verify the effectiveness of the methods used, we need to split the testing dataset to evaluate them. We present three methods in the Section 2 and compare the distributions of the labels in Figure 6. If we apply the first method, we can observe that the distribution in the training, validation, and testing dataset is very similar to Figure 6a. For the second method, the distribution of training and validation is similar but different from the testing dataset shown in Figure 6b. As for the third method in Figure 6c, the validation and testing are all different from the training dataset.

In fact, there are only two sets of testing data. One is randomly split and the other is split by consecutive years. The bias and RMSE of point to point in Table 5 represent the errors of the original intensity in the ERA5 reanalysis. If we calculate the Vmax using the 10 m wind speed, we can observe that the RMSE is 69.82 kts in the testing dataset (10%) and 67.98 kts in the testing dataset (2021–2022) before correction. After linear correction, the RMSE is reduced to 20.86 kts and 19.01 kts, respectively. The biases of these two datasets are all close to 1 kt, confirming the accuracy of the linear model. There are a few differences between the results of these two testing datasets. We also calculate the maximum wind speed from 850 hPa and obtain similar results with the surface wind speed. The linear method corrects the bias and RMSE to some extent. But, we consider that the wind speed at 850 hPa is collected from pressure levels , which may contain less noise. From the comparison of the wind structure on these two levels, shown in the Figure 7, we can observe that the pattern at 850 hPa is more obvious than at the surface. Therefore, 850 hPa was chosen as the base level for constructing the inputs.



(**a**) (**b**) (**c**)

Figure 6. This is a figure showing the data distribution of the labels in the training dataset (marked as train), validation dataset (val), and testing dataset (test) from three splitting methods. (**a**) shows the first method, which uses 80% for training, 10% for validation, and 10% for testing. The mean value of the train and val are around 88 kts with a standard deviation of 21 kts, but a different value with the test is about 2 kts. (**b**) shows the second method, that is using the 2021–2022 samples for testing, and the rest (2004–2020) is split into training (90%) and validation dataset (10%). The mean and standard deviation are similar to the first. (**c**) demonstrates the third method, using the 2004–2018 samples for training, 2019–2020 for validation, and 2021–2022 for testing . Here, the means of val and test are close, and slightly different from the training dataset.

Table 5. This is a table showing the intensity errors of the original dataset (Point to Point) and the linear correction (Linear model) on two testing datasets, one taken from 10% of all samples between 2004–2022 and the other from samples collected between 2021–2022. And, the intensity in the original ERA5 reanalysis is defined as the maximum wind speed calculated from the surface wind speed or 850 hPa. The intensity errors before and after linear correction in the testing dataset (2021–2022) are shown in bold type.

Data	Method	Testing Dataset (10%)		Testing Dataset (2021–2022)	
		Bias (kts)	RMSE (kts)	Bias (kts)	RMSE (kts)
Surface	Point to Point	−66.65	69.82	**−65.16**	**67.98**
	Linear model	−1.48	20.86	**1.08**	**19.01**
850 hPa	Point to Point	−52.48	56.7	−49.72	53.6
	Linear model	−1.51	21.04	0.53	19.74

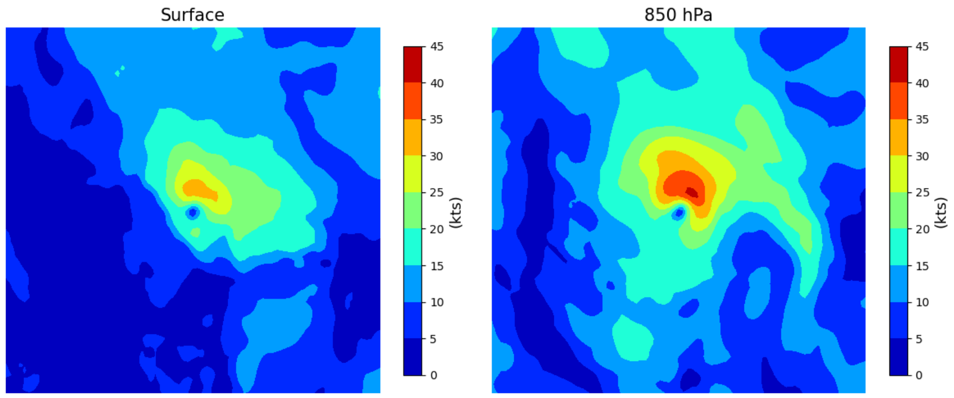


Figure 7. This is a figure showing the differences in the wind speed pattern between the surface (**left**) and 850 hPa (**right**). The wind speed data are from Hurricane ALEX at 3:00 on 3 August 2004. Since the 850 hPa pattern is more obvious than the surface pattern, and the reanalysis data are from ERA5 pressure levels with less noise, it was chosen as the base level for constructing the inputs for the next experiments.

The above methods show the potential of linear correction. However, it remains a large RMSE when used for applications. In addition, it ignores the spatial environmental information surrounding the storms when correcting the intensity. We are therefore considering the use of non-linear methods for further correction. Deep neural networks are our first choice, which we introduce in the previous parts. ResNet-18, without pre-trained parameters, is used as our basic network architecture. We split the dataset into the three ways mentioned above and then use the gridded wind speed around the storm centre with a spatial region of $20^\circ \times 20^\circ$ to train, validate, and test the network. We use bilinear interpolation to convert the original shape of the ERA5 wind speed (81, 81, 1) to (224, 224, 1) to match the original input shape of ResNet-18. We also change the unit of the output layer to 1 for our regression task. We set the loss function to mean square error (MSE) and select the adaptive moment estimation (Adam) as the optimal algorithm. For the hype parameters, we set the batch size to 32, the epochs to 50, and the learning rate to 0.0001 through a series of experiments.

The results are very similar between the surface and 850 hPa, which also validates the feasibility of our operation of using 850 hPa as a proxy for the base level. We focus on the bias and RMSE of 850 hPa as an input. They show significantly different results when the testing dataset is partitioned in Table 6. The test RMSE is 9.8 kts when the testing data are taken from the same data distribution as the training dataset using randomly partitioned methods. The scatter plot in Figure 8a shows a linear correlation between IBTrACS_Vmax and ResNet_Vmax (the predicted intensity of ResNet-18) in the testing dataset, so it is possible to use linear correction to remove the residuals. However, there is no obvious correlation with IBTrACS_Vmax and ResNet_Vmax in the testing dataset (2021–2022). It can be observed from Figure 8b,c that the RMSE is all above 16 kts even using different validation datasets if the testing dataset is from consecutive years. We can conclude that the non-linear model based on ResNet-18 performs better in intensity correction than the linear model; however, the error is still far from the acceptable average intensity error in practical applications. Therefore, we next focus on the testing dataset split from consecutive years and optimise the inputs and features to improve the performance. Since the validation method does not have much impact on the testing results shown in the previous experiments, we only use the third splitting method (2004–2018 for training, 2019–2020 for validation, and 2021–2022 for testing) in the next experiments.

Table 6. This is a table showing the errors of the trained model based on ResNet-18 without pre-trained parameters on two testing datasets, one taken from 10% of all samples between 2004 and 2022 and the other from samples collected between 2021 and 2022. In the first case, the validation dataset is taken from 10% of all samples between 2004 and 2022. In the second case, the validation dataset is taken from 10% of the remaining years (2004–2020) or from the samples between 2019 and 2020. These three partitioned datasets correspond to the methods described in Figure 6. The inputs are gridded wind speed with the inputs' shape of (None, 224, 224, 1) at the surface or 850 hPa. Errors are shown in bold for the testing dataset (2021-2022) and will be used as the baseline for the next optimisations.

Data	Testing Dataset (10%)		Testing Dataset (2021–2022)			
	Validation (10%)		Validation (10% in 2004–2020)		Validation (2019–2020)	
	Bias (kts)	RMSE (kts)	Bias (kts)	RMSE (kts)	Bias (kts)	RMSE (kts)
Surface	0.70	11.03	−0.64	16.06	−1.99	16.67
850 hPa	−0.04	9.8	1.60	16.99	**−2.28**	**16.41**

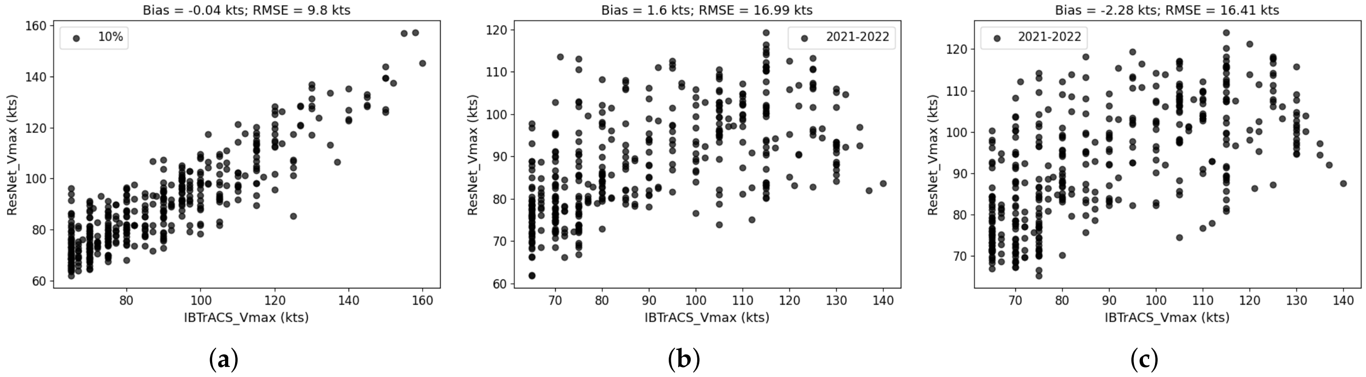


Figure 8. This is a figure showing the correlation between the predictions of the ResNet-based model in three partitioned datasets presented in Figure 6. (**a**) shows linear correlation between IBTrACS_Vmax (labels) and ResNet_Vmax (predictions) on the testing dataset (10% in 2004–2022). (**b**,**c**) show that there is no obvious correlation between IBTrACS_Vmax and ResNet_Vmax on the testing dataset (2021–2022), even using different validation datasets. The results of the third partitioned dataset in (**c**) will be the baseline for the next optimisation using our approach.

3.2.2. TC Knowledge for Optimising the Inputs

As machine learning approaches rely highly on data quality, the construction of inputs is extremely important. We find that there is no obvious correspondence between the inputs and the outputs in the data analysis section using statistical methods. To further improve the restrictiveness of the single-level inputs, we update them in three ways, as shown in Figure 9. The first is to use the original data without bilinear interpolation to preserve the true information hidden in the data. The second is to crop the region to 10° × 10° and then use bilinear interpolation to resize it. The reason for using the crop operation is to make the structure of the wind near the centre of the storm clearer. And, the third operation is to rotate the inputs according to the direction of the storm speed to unify and standardise the wind pattern, and then crop and resize it. From Table 7, we can check the effectiveness of resizing the inputs compared to the result of the original inputs. Another finding is that the crop operation makes a small improvement, but the rotate operation is not useful here. Therefore, in the next experiments, we will only use the crop operation.

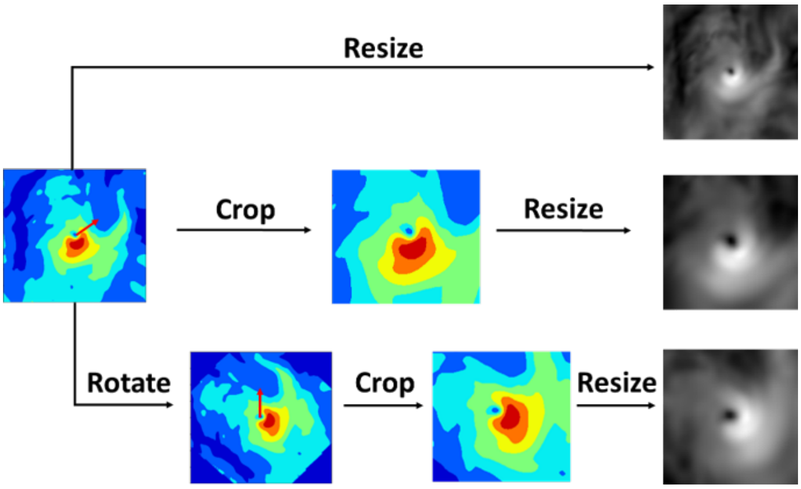


Figure 9. This is a figure showing three data processing methods. The red arrow in the figure is the direction of the storm. The gridded wind speed field in the reanalysis dataset can be plotted as a contour with different colours and also could be shown as a greyscale image.

Table 7. This is a table showing the testing errors of comparative experiments using different data processing methods. Samples from 2004 to 2018 are used for training, 2019–2020 for validation, and 2021–2022 for testing. The network and the training process are the same as in the baseline section; only the shape of the inputs is changed. Best performance is in bold.

Data Augmentation	Inputs' Shape	Bias (kts)	RMSE (kts)
Original	(None, 81, 81, 1)	0.88	17.46
Crop + Resize	(None, 224, 224, 1)	**−1.54**	**15.21**
Rotate + Crop + Resize	(None, 224, 224, 1)	−1.56	16.08

The above experiments may provide new evidence that the correspondence between single-level wind and label is ambiguous. The one-to-many and many-to-one problems remain to be solved. Therefore, we use additional information to update the inputs, trying to ensure that the inputs contain enough information that can be learned by the neural networks. We implement this in two ways, by increasing the spatial levels of the wind and by increasing the atmospheric variables. Here, we use the base level of 850 hPa and add the middle level (500 hPa) and the top level (200 hPa). We add the equivalent potential temperature θ_E that contributes to the TC evolution [53] calculated by MetPy using pressure (p), temperature (t), and relative humidity (r). We find the effectiveness of this operation in Table 8, and the RMSE is reduced to 14.90 kts when we use the variables including wind and θ_E on three pressure levels of 850 hPa, 500 hPa, and 200 hPa.

Table 8. This is a table showing the testing errors of comparative experiments using different additional information. Samples from 2004 to 2018 are used for training, 2019–2020 for validation, and 2021–2022 for testing. The network and the training process are the same as in the baseline section; only the input shape of each sample is changed. Best performance is in bold.

Variables	Levels	Input Shape	Bias (kts)	RMSE (kts)
Wind	850 hPa, 200 hPa	(224, 224, 2)	−0.66	16.19
	850 hPa, 500 hPa	(224, 224, 2)	−1.11	16.60
	850 hPa, 500 hPa, 200 hPa	(224, 224, 3)	−1.36	15.47
	850 hPa	(224, 224, 2)	0.54	18.00
Wind, θ_E	850 hPa, 200 hPa	(224, 224, 4)	−0.52	16.45
	850 hPa, 500 hPa	(224, 224, 4)	2.32	17.03
	850 hPa, 500 hPa, 200 hPa	(224, 224, 6)	**0.82**	**14.90**

3.2.3. Feature Learning for Improving the Generalisability

Although the results are now better than the baseline (16.41 kts in Figure 8c), after updating the inputs, the generalisability of the model does not seem to improve much. This is because the testing error of the trained model using (Wind, θ_E) at (850 hPa, 500 hPa, 200 hPa) remains 14.90 kts larger than the uncertainty of the best track data (7 kts) in the North Atlantic. So, in this section, we start to change the way to find solutions. We split the model from inputs to outputs into two parts and update them separately. Because ResNet-18 was used in this paper, the network architecture consists of an input layer, a convolutional layer, a max pooling layer, four types of residual blocks (each type of number is two), an average pooling layer, and an output layer. Specifically, features are extracted from the residual blocks constructed by convolutional layers and then reduced to the dimension of 512 by the average pooling layer. Thus, we can obtain the features from the average pooling layer after the inputs have been fed into the network. Here, we define the input-feature part (feature extractor) of the model as the process from the inputs to the returns of the average pooling layer, and the feature-output mapping of the model as the process from the returns of the average pooling layer to the output layer. We will gradually consider whether the input-feature and feature-output mapping are effective or not and then decide whether or not to update them to improve performance.

We introduce three ways of splitting the dataset before, and we find that the testing error in Table 6 from the randomly split is satisfactory, but the results from the subsequent newly coming years are vice versa. We can conclude from the former finding that the ResNet-18 network is able to effectively represent the inputs from 2004 to 2022 with general features in the entire data space. This can also be used to validate the effectiveness of the representative ability of deep neural networks. However, in the sub-data space of the validation dataset (2019–2020) and testing dataset (2021–2022) of the unseen new years, the ResNet-18 model trained by the data of the historical years (2004–2018) does not work well enough. As for the reasons, we try to deduce them from the two aspects. One of them may be that the feature-output mapping is not effective enough, and another reason may be that the general features learned from the historical years are not appropriate for the storms in the new coming years, or both.

We perform the following experiments to validate the above assumptions. The first operation is to enlarge the training dataset using data augmentation to reduce overfitting and then improve the generalisability of the features. This also helps to reduce the impact of sample size and validate that the small size can also be used to train a network model. Specifically, we use random rotation to increase the size, as we have a finding in Table 7 to verify that there is no obvious effect on the results when the inputs are rotated. We use the same experimental setup as in the previous experiments, including the computational environment, network architecture, hype parameters, and so on. We just increase the epochs to 100 and then save the best model based on the validation error in the training process. And, we find that there is no significant difference between the different sample sizes in Table 9. In order to balance the computational cost and the generalisability, we adopt the model trained on the third group setting shown in Table 9. Specifically, we use the new training dataset consisting of the original training dataset from 2004 to 2018 and two copied training datasets (fold 2) with random rotation to train the model, and save the trained model without the output layer as the general feature extractor.

Table 9. This is a table showing the testing errors of comparative experiments using different folds to enlarge the training dataset (2004–2018) by random rotation. Samples from 2019 to 2020 are still used for validation and 2021–2022 for testing. For example, the third experimental setting means that the new training dataset consists of the original training dataset from 2004 to 2018 and two copied training datasets (fold 2) with random rotation. The basic network is the same as in Table 8, and the shape of the inputs is (None, 224, 224, 6) with the variables of (Wind, θ_E) at (850 hPa, 500 hPa, 200 hPa). Best performance is in bold.

Fold	Sample Size (Training Dataset)	Bias (kts)	RMSE (kts)
0	3258	0.82	14.90
1	6516	−2.43	15.47
2	9774	**−1.07**	**14.76**
3	13,032	−0.77	15.16
4	16,290	−0.26	15.14

We freeze the general feature extractor described in the last paragraph and then reorganise the dataset. For the next set of experiments, the inputs to the learning tasks are changed from the ERA5 variables X to the corresponding features χ after the feature extraction of the ERA5 variables, but the labels remain the same. In the first set of experiments, the features or feature-output mapping are updated using the samples from the previous training years (2004–2018), and also using the samples from 2019 to 2020 for validation, and the testing years (2021–2022) for testing. The dataset used to retrain a model is D1, shown in Table 10, and the methods we use here are classical machine learning (ML) algorithms and MLP. The top three ML algorithms in our validation are the logistic regression (LR), support vector machine (SVR), and gradient boosting regression (GBR). The choice of MLP is a single-layer network with 1 unit, or 3 layers with units of 1024, 512, and 1, or 5 layers with units of 1024, 4096, 1024, 512, and 1. In fact, LR, SVR, GBR, and MLP with one layer only update the feature-output mapping, but MLP with three layers and MLP with five layers update the features and the feature-output mapping simultaneously, referring to our definition for them in this paper.

The second set of experiments uses the same methods and settings as the first set, only changing the samples to reorganise the dataset as D2. Here, we use 90% of the samples from the previous validation years (2019–2020) to retrain the model to update the features or feature-output mapping, and the remaining 10% for validation. The 2021–2022 samples are still used for testing. The third set of experiments also uses the same methods and settings as the first set but changes the samples to reorganise the dataset as D3 and add DA as a new method. We use 90% of the samples from the previous training and validation years (2004–2020) to retrain the model to update the features or feature-output mapping, and the remaining 10% for validation. The 2021–2022 samples are still used for testing. We set the loss weight of MMD to 1, 100, and 1000 separately when using the DA method.

Here, we design these three sets of experiments to validate the effectiveness of previous training information, validation information not used for training, and all available known information for feature learning. In all experiments, the inputs in D_{train} are x_{train}, although it denotes the features χ, and the labels are y_{train}. The inputs and labels in D_{val} (x_{val}, y_{val}) and D_{test} (x_{test}, y_{test}) are similar.

Table 10. This is a table showing three datasets (D1, D2, D3) described in the text for feature learning.

	Training Dataset (D_{train})	**Validation Dataset (D_{val})**	**Testing Dataset (D_{test})**
D1	$(\chi^*_{train}, y^*_{train})$ *	$(\chi^*_{val}, y^*_{val})$	$(\chi^*_{test}, y^*_{test})$
D2	90% of $(\chi^*_{val}, y^*_{val})$	10% of $(\chi^*_{val}, y^*_{val})$	$(\chi^*_{test}, y^*_{test})$
D3	90% of $((\chi^*_{train}, y^*_{train}) + (\chi^*_{val}, y^*_{val}))$	10% of $((\chi^*_{train}, y^*_{train}) + (\chi^*_{val}, y^*_{val}))$	$(\chi^*_{test}, y^*_{test})$

* χ^*_{train} and y^*_{train} in the training dataset D_{train} denote the features and labels from previous training samples between 2004 and 2018. It is the same with the validation dataset D_{val} and the testing dataset D_{test}. Thus, χ^*_{test} denotes the features and y^*_{test} denotes the labels from previous testing samples between 2020 and 2021.

Here, we try to describe the details of the DA method we use in this part. As we mentioned in Section 2, there may be some differences in the data distribution between the training and testing datasets in practical applications, leading to a weak generalisability of the trained model. Thus, we can consider D_{train} as the source domain and D_{test} as the target domain, referring to the concept of DA. The architecture design is inspired by domain adaptive neural networks (DaNN) [54] and deep domain confusion (DDC) [55]. Our loss, as defined here, consists of two parts. One is the MSE between the predictions and labels in the testing dataset, and the other is the MMD distance between the features of the training and testing data. The method is shown in Figure 10, and the total loss can be expressed as

$$Loss_{DA} = Loss_1(x_{train}, y_{train}) + \alpha Loss_2(x_{train}, x_{test}). \tag{8}$$

In this formula, $Loss_1$ demonstrates MSE and $Loss_2$ demonstrates the square of MMD. In detail, the square of the MMD can be expressed as follows:

$$MMD^2(\chi'_{train}, \chi'_{test}) = \left\| \sum_{i=1}^{n_1} \phi(\chi'_{train_i}) - \sum_{j=1}^{n_2} \phi(\chi'_{test_j}) \right\|_H^2. \tag{9}$$

$\phi(\cdot)$ is the mapping that converts the original data to RKHS (Reproducing Kernel Hilbert Space), and n_1, n_2 are the sample sizes of the training and testing dataset separately. The transition allows the features from these two types of datasets to be compared in a high dimension. Here, we use the multi-kernel MMD (MK-MMD) as the distance function for the features and try to find the appropriate weight α to balance the two parts of the loss. The aim of this method is to update the general features from the training dataset into the specific features of the testing dataset so that it can be used to improve the generalisability of the model and then reduce the testing error.

The results in Table 11 show that there is no significant improvement in error reduction using D1 and D2, whether based on traditional ML algorithms (LR, SVR, and GBR) or MLP with different layers. However, there is an obvious improvement in D3 based on MLP, so we focus on the results analysis of the D3 dataset. In fact, when we use MLP with a single layer of a unit, only the feature-output mapping is updated. But if we use MLP with three layers of 1024, 512, and 1 unit, the features have been updated to a new one, even if the shape is still (None, 512). And, the feature-output mapping was updated at the same time. The results show that three layers perform better than one layer and five layers, reducing the RMSE to 11.55 kts. Although it is much improved compared to the test results (16.41 kts) in 2021–2022 shown in Figure 8c, it is still larger than the uncertainty of the intensity (7 kts) in the North Atlantic. Therefore, we use the DA method shown in Figure 10.

We only adjust the loss weight of MMD (α) as the different setting in this method. They are 1, 100, and 100, respectively. We find that the results are not very different when choosing 1 or 100, and the RMSE are all 5.99 kts, although the bias shows little difference. However, the RMSE increases when the weight is set to 1000. We choose the best result using the DA method with a loss weight of 100 as the final approach in this paper and focus on analysing the results for this. This method reduces the RMSE to 5.99 kts, which is less than the uncertainty of the intensity in the North Atlantic. The scatter plot in Figure 11 shows an obvious linear correlation between the labels (IBTrACS_Vmax) and the predictions (DA_Vmax), and the Pearson correlation coefficient (r) of these two intensity values also confirms that it is 0.7 (>0.5). However, the scatter of the predictions is not very centred and has a dispersed shape. For this reason, we also show the error distribution of the predictions. The error distribution in Figure 12 (left) shows that the errors are centred in the range [−20, 20], although some samples are outside this range. As for the prediction distribution in Figure 12 (right), it shows a Gaussian distribution that does not fit the label distribution. We suspect that the reason for this is that we use the MSE as the main loss function; the distribution does not change when we add the MMD loss. Therefore, the DA method could be optimised and further explored in the future.

Table 11. This is a table showing the testing errors of comparative experiments using different methods (ML, MLP, DA) in three datasets (D1, D2, D3) for feature learning. Best performance is in bold.

Data	Methods	Setting	Bias (kts)	RMSE (kts)
D1	ML	LR	−1.15	14.83
		SVR	−1.53	14.76
		GBR	−1.30	14.92
	MLP	1 (1)	−1.09	14.89
		3 (1024/512/1)	1.24	15.12
		5 (1024/4096/1024/512/1)	−0.39	15.18
D2	ML	LR	9.51	52.31
		SVR	−0.06	15.71
		GBR	2.24	15.88
	MLP	1 (1)	1.63	15.26
		3 (1024/512/1)	−1.09	14.89
		5 (1024/4096/1024/512/1)	0.69	14.76
D3	ML	LR	−0.52	14.82
		SVR	−1.46	14.73
		GBR	−0.63	14.72
	MLP	1 (1)	1.23	11.74
		3 (1024/512/1)	0.27	11.55
		5 (1024/4096/1024/512/1)	0.61	11.51
	DA	1	−2.45	5.99
		100	**−2.43**	**5.99**
		1000	−2.35	10.39

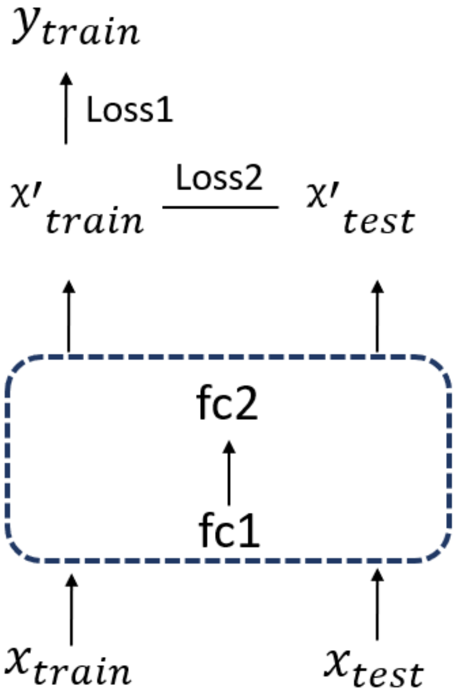


Figure 10. This is a figure showing the domain adaptation (DA) method we use in this section. x_{train} and y_{train} in the training dataset D_{train} denote the inputs and labels. In the blue box, fc1 and fc2 are the fully connected layers of the neural networks, here with 1024 and 512 units, respectively. They share all parameters with two types of inputs (x_{train}, x_{test}). χ'_{train} and χ'_{test} are new features updated by the network. Loss1 and Loss2 are two parts of $Loss_{DA}$, which we define in Formula (8).

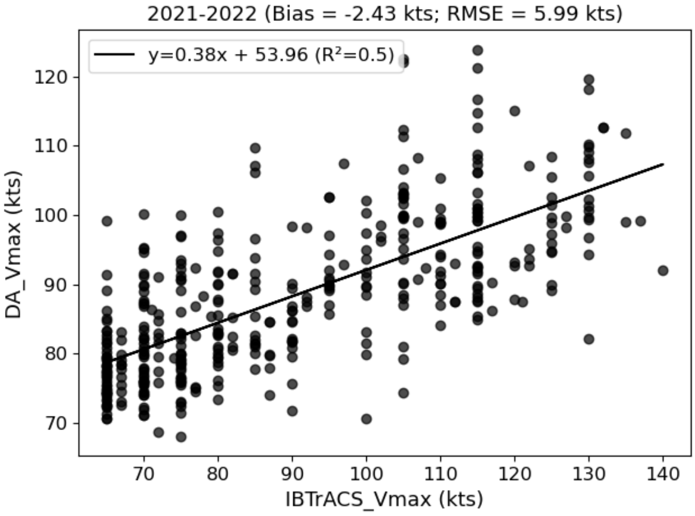


Figure 11. This is a scatter plot showing the correlation between IBTrACS_Vmax and DA_Vmax in the testing dataset (2021–2022). The bias and RMSE of our DA method (MMD weight = 100) are −2.43 kts and 5.99 kts, respectively. The Pearson correlation coefficient (r) of these two intensity values is 0.7 (>0.5), indicating a strong positive correlation. The solid line shows the linear fit with $R^2 = 0.5$.

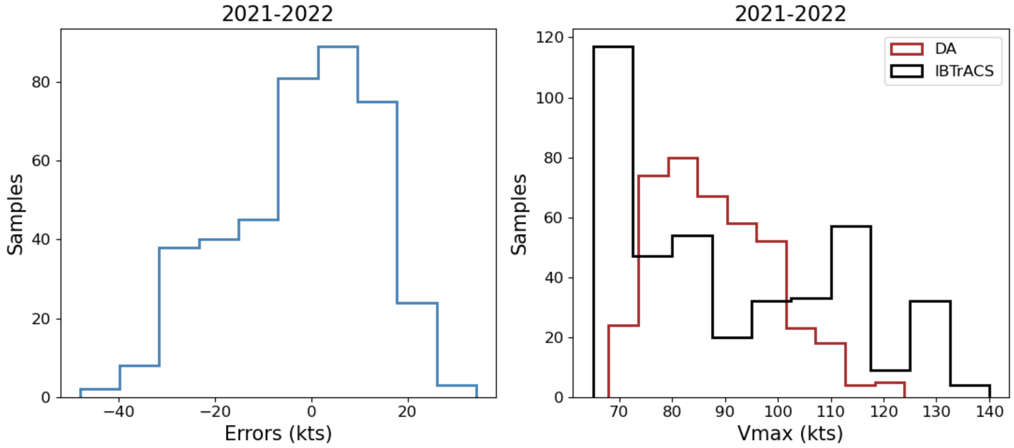


Figure 12. This is a figure showing the error distribution (**left**) and the prediction distribution (**right**) of our DA method (MMD weight = 100). The errors (RMSE) are centred in the range [−20, 20].

4. Discussion

As shown in the previous introduction and experiments, the intensity errors in the reanalysis dataset are large and need to be corrected. Traditional linear correction can reduce the errors to some extent, but some problems remain in practical applications. This may be because there is no obvious linear correlation between the intensity value calculated from reanalysis and the true intensity in the best track dataset. Thus, here we use non-linear methods to try to improve it. Since ResNet is a widely used network with strong representative ability and solves the problem of vanishing gradient in deep neural networks with residual blocks, we use it as the base model and the results of the trained model as the baseline. However, when we split the dataset in different ways, we find that there are many differences, especially in the selection of the testing dataset. If we split the testing dataset randomly with a ratio like 10% in machine learning, we can obtain a satisfactory result. However, if we split the testing dataset based on the requirement of practical tasks such as tropical cyclone correction with consecutive new coming years, the result is not satisfactory. We consider that the former may follow the basic assumption that the training and testing datasets are from the same data distribution, but the latter obeys it. It also means that the correspondence between inputs and outputs may change over time.

To solve this problem, our first option is to optimise the inputs. Since the labels are fixed, the inputs contain more value information, making it easier to learn the correspondence for networks. The methods we use to update the inputs are based on existing knowledge about tropical cyclones and then combined with general data processing methods in machine learning. For example, we crop the inputs to half their original size to make the central pattern clearer. We also rotate the inputs with the direction of the storm to unify the wind pattern going forward in the same direction to reduce noise. But, it seems that the crop operation is a bit useful here and the rotate operation is not. So, for the next experiments, we crop the inputs to optimise the results. We also randomly rotate the inputs to increase the training data set to reduce overfitting and then improve the generalisability of the features. In addition, we increase the spatial levels of the wind into three pressure levels, including the bottom, middle, and top levels of tropical cyclones, and add the physical variable θ_E related to tropical cyclone evolution into the inputs. The results also show positive feedback, so we use the inputs with multiple variables and levels as the optimised inputs for the next parts. The overall performance has been significantly improved compared to the linear correction and the basic version of ResNet-18 with the simplest inputs of single-level wind speed at 850 hPa. It seems that the problem of different data distribution in the training and testing datasets is still not solved, for the reason that the trained model may be of weak generalisability.

Fortunately, transfer learning is designed to solve the problem of different data distribution, so it provides a new insight to solve our problem. So, we start to update the features extracted from the trained model and the feature-output mapping using the idea of fine-tuning and domain adaptation. First, we try to update these two parts using the original training dataset, but it does not work for the testing dataset. So, then we update these two parts using the local information in the original validation dataset that was not used in the training process before, but it still does not work. We also update these two parts using all available information in the original training and validation datasets; we find an improvement in the testing result. Finally, we add the features of the original testing dataset to participate in the new training process for updating the features and feature-output mapping using domain adaptation. The results show a significant decrease in the testing errors, and the RMSE of the intensity is within the uncertainty of the best track data in the North Atlantic. The effectiveness of our approach has now been validated.

However, there are still many issues to be resolved and assumptions to be validated for our work. Firstly, we only use tropical cyclone data in the North Atlantic from 2004 to 2022; the reason why we chose it is that it is of the highest quality to train the network model. However, if our approach is applied to other basins or historical years where the best track data are inhomogeneous and noisy, new problems may appear. Therefore, the approach still needs to be validated for other datasets to make it general and practical. Secondly, we only use ResNet-18 as the feature extractor in this paper; maybe there are better choices like ResNet-34, 101, other general network architectures like Inception or custom networks, and so on. As for the loss function, we use MSE as the basic objective function; maybe there are more appropriate objective functions for the specific tasks. All the designs of deep learning in this paper are created from basic settings, and they can be optimised in many ways. Here, we only aim to provide the basic version for researchers who are interested in related topics. Thirdly, the setting of the final domain adaptation methods still needs to be validated and explored. Although the method helps us to achieve the goal, it still lacks some explanations. For example, does it really bring the features of the testing and training data closer together? Does the choice of MMD or MK-MMD have a major impact on the model? Are there more appropriate distance metrics and loss weights to improve generalisability? This may be a new topic that can be explored in depth in the future.

5. Conclusions

In this paper, we develop an adaptive learning approach to correct tropical cyclone intensity in reanalysis due to the complexity of our research problem. Unlike learning tasks in computer vision or other machine learning applications, the data correspondence between inputs and outputs in our task may not be constant. This means that it may be difficult to learn the mapping from inputs to outputs using deep neural networks alone. Therefore, we first consider fusing additional environmental information about tropical cyclones in space and attributes into the inputs to improve the correspondence in data space. In addition, the data distribution of inputs and outputs may change over time, so we also refer to the basic idea of domain adaptation in transfer learning to optimise the training process to improve the generalisability of the model. The experiments confirm the effectiveness of our approach. In particular, we reduce the RMSE to 5.99 kts within the intensity uncertainty of IBTrACS in the North Atlantic, while the error in the original dataset is 67.98 kts. We also compare our approach with the linear correction and ResNet-18, which have an RMSE of 19.01 kts and 16.41 kts, respectively. More importantly, our approach could be extended to other similar learning tasks, such as the correction of intensity estimation from satellite imagery, intensity prediction from dynamical models, and so on. It is also not limited to the same computational environment and version, making it friendly and convenient for researchers who will use it for practical applications in the future.

Author Contributions: Conceptualization, R.C., R.T. and Y.D.; Methodology, R.C., R.T., X.S. and Y.D.; Software, R.C. and X.S.; Validation, R.C., R.T., X.S. and W.Z.; Formal analysis, R.C., R.T. and X.S.; Investigation, R.C., R.T., X.W., Y.D. and W.Z.; Resources, R.C., R.T., X.W. and W.Z.; Data curation, R.C. and R.T.; Writing—original draft, R.C.; Writing—review and editing, R.C., R.T., X.S. and W.Z.; Visualization, R.C.; Supervision, R.T., X.W. and W.Z.; Project administration, R.T. and W.Z.; Funding acquisition, X.W. and W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by National Key R&D Program of China (2021YFC3101500) and is partially supported by National Natural Science Foundation of China (Grant No. 62372460). R.C. acknowledges the China Scholarship Council (CSC) for the fellowship support (No. 202006110012).

Data Availability Statement: The IBTrACS data used in this study are publicly available from the National Centres for Environmental Information (NCEI) at https://www.ncei.noaa.gov/data/international-best-track-archive-for-climate-stewardship-ibtracs/v04r00/access/csv/ (accessed on 22 May 2023). The ERA5 reanalysis are publicly available from the Climate Data Store at https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-single-levels?tab=form (accessed on 22 May 2023).

Acknowledgments: We are grateful to the IBTrACS and ERA5 data providers for providing the data used in our experiments. We acknowledge Yijun Gu at Imperial College London and Miao Feng at the University of Montreal for their suggestions on this study. We also appreciate the help of colleagues in the tropical cyclone group at the Department of Physics, Imperial College London.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Emanuel, K. Increasing destructiveness of tropical cyclones over the past 30 years. *Nature* **2005**, *436*, 686–688. [CrossRef] [PubMed]
2. Peduzzi, P.; Chatenoux, B.; Dao, H.; De Bono, A.; Herold, C.; Kossin, J.; Mouton, F.; Nordbeck, O. Global trends in tropical cyclone risk. *Nat. Clim. Chang.* **2012**, *2*, 289–294. [CrossRef]
3. Wang, S.; Toumi, R. Recent migration of tropical cyclones toward coasts. *Science* **2021**, *371*, 514–517. [CrossRef]
4. Emanuel, K.; DesAutels, C.; Holloway, C.; Korty, R. Environmental control of tropical cyclone intensity. *J. Atmos. Sci.* **2004**, *61*, 843–858. [CrossRef]
5. Wang, Y.q.; Wu, C.C. Current understanding of tropical cyclone structure and intensity changes—A review. *Meteorol. Atmos. Phys.* **2004**, *87*, 257–278. [CrossRef]
6. DeMaria, M.; Sampson, C.R.; Knaff, J.A.; Musgrave, K.D. Is tropical cyclone intensity guidance improving? *Bull. Am. Meteorol. Soc.* **2014**, *95*, 387–398. [CrossRef]
7. Emanuel, K. 100 years of progress in tropical cyclone research. *Meteorol. Monogr.* **2018**, *59*, 15.1–15.68. [CrossRef]
8. Knapp, K.R.; Kruk, M.C. Quantifying interagency differences in tropical cyclone best-track wind speed estimates. *Mon. Weather Rev.* **2010**, *138*, 1459–1473. [CrossRef]
9. Levinson, D.H.; Diamond, H.J.; Knapp, K.R.; Kruk, M.C.; Gibney, E.J. Toward a homogenous global tropical cyclone best-track dataset. *Bull. Am. Meteorol. Soc.* **2010**, *91*, 377–380.
10. Kossin, J.P.; Olander, T.L.; Knapp, K.R. Trend analysis with a new global record of tropical cyclone intensity. *J. Clim.* **2013**, *26*, 9960–9976. [CrossRef]
11. Emanuel, K.; Caroff, P.; Delgado, S.; Guishard, M.; Hennon, C.; Knaff, J.; Knapp, K.R.; Kossin, J.; Schreck, C.; Velden, C.; et al. On the desirability and feasibility of a global reanalysis of tropical cyclones. *Bull. Am. Meteorol. Soc.* **2018**, *99*, 427–429. [CrossRef]
12. Dvorak, V.F. Tropical cyclone intensity analysis and forecasting from satellite imagery. *Mon. Weather Rev.* **1975**, *103*, 420–430. [CrossRef]
13. Velden, C.; Harper, B.; Wells, F.; Beven, J.L.; Zehr, R.; Olander, T.; Mayfield, M.; Guard, C.C.; Lander, M.; Edson, R.; et al. The Dvorak tropical cyclone intensity estimation technique: A satellite-based method that has endured for over 30 years. *Bull. Am. Meteorol. Soc.* **2006**, *87*, 1195–1210. [CrossRef]
14. Knaff, J.A.; Brown, D.P.; Courtney, J.; Gallina, G.M.; Beven, J.L. An evaluation of Dvorak technique–based tropical cyclone intensity estimates. *Weather Forecast.* **2010**, *25*, 1362–1379. [CrossRef]
15. DeMaria, M.; Kaplan, J. An updated statistical hurricane intensity prediction scheme (SHIPS) for the Atlantic and eastern North Pacific basins. *Weather Forecast.* **1999**, *14*, 326–337. [CrossRef]
16. Knaff, J.A.; DeMaria, M.; Sampson, C.R.; Gross, J.M. Statistical, 5-day tropical cyclone intensity forecasts derived from climatology and persistence. *Weather Forecast.* **2003**, *18*, 80–92. [CrossRef]
17. DeMaria, M.; Mainelli, M.; Shay, L.K.; Knaff, J.A.; Kaplan, J. Further improvements to the statistical hurricane intensity prediction scheme (SHIPS). *Weather Forecast.* **2005**, *20*, 531–543. [CrossRef]

18. Lee, C.Y.; Tippett, M.K.; Camargo, S.J.; Sobel, A.H. Probabilistic multiple linear regression modeling for tropical cyclone intensity. *Mon. Weather Rev.* **2015**, *143*, 933–954. [CrossRef]
19. Cangialosi, J.P.; Blake, E.; DeMaria, M.; Penny, A.; Latto, A.; Rappaport, E.; Tallapragada, V. Recent progress in tropical cyclone intensity forecasting at the National Hurricane Center. *Weather Forecast.* **2020**, *35*, 1913–1922. [CrossRef]
20. DeMaria, M.; Franklin, J.L.; Zelinsky, R.; Zelinsky, D.A.; Onderlinde, M.J.; Knaff, J.A.; Stevenson, S.N.; Kaplan, J.; Musgrave, K.D.; Chirokova, G.; et al. The national hurricane center tropical cyclone model guidance suite. *Weather Forecast.* **2022**, *37*, 2141–2159. [CrossRef]
21. Chen, R.; Zhang, W.; Wang, X. Machine learning in tropical cyclone forecast modeling: A review. *Atmosphere* **2020**, *11*, 676. [CrossRef]
22. Pradhan, R.; Aygun, R.S.; Maskey, M.; Ramachandran, R.; Cecil, D.J. Tropical cyclone intensity estimation using a deep convolutional neural network. *IEEE Trans. Image Process.* **2017**, *27*, 692–702. [CrossRef] [PubMed]
23. Combinido, J.S.; Mendoza, J.R.; Aborot, J. A convolutional neural network approach for estimating tropical cyclone intensity using satellite-based infrared images. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018, pp. 1474–1480.
24. Chen, B.F.; Chen, B.; Lin, H.T.; Elsberry, R.L. Estimating tropical cyclone intensity by satellite imagery utilizing convolutional neural networks. *Weather Forecast.* **2019**, *34*, 447–465. [CrossRef]
25. Zhuo, J.Y.; Tan, Z.M. Physics-augmented deep learning to improve tropical cyclone intensity and size estimation from satellite imagery. *Mon. Weather Rev.* **2021**, *149*, 2097–2113. [CrossRef]
26. Lee, Y.J.; Hall, D.; Liu, Q.; Liao, W.W.; Huang, M.C. Interpretable tropical cyclone intensity estimation using Dvorak-inspired machine learning techniques. *Eng. Appl. Artif. Intell.* **2021**, *101*, 104233. [CrossRef]
27. Xu, W.; Balaguru, K.; August, A.; Lalo, N.; Hodas, N.; DeMaria, M.; Judi, D. Deep learning experiments for tropical cyclone intensity forecasts. *Weather Forecast.* **2021**, *36*, 1453–1470.
28. Chen, R.; Wang, X.; Zhang, W.; Zhu, X.; Li, A.; Yang, C. A hybrid CNN-LSTM model for typhoon formation forecasting. *GeoInformatica* **2019**, *23*, 375–396. [CrossRef]
29. Zhang, Z.; Yang, X.; Shi, L.; Wang, B.; Du, Z.; Zhang, F.; Liu, R. A neural network framework for fine-grained tropical cyclone intensity prediction. *Knowl.-Based Syst.* **2022**, *241*, 108195. [CrossRef]
30. Boussioux, L.; Zeng, C.; Guénais, T.; Bertsimas, D. Hurricane forecasting: A novel multimodal machine learning framework. *Weather Forecast.* **2022**, *37*, 817–831. [CrossRef]
31. Jordan, M.I.; Mitchell, T.M. Machine learning: Trends, perspectives, and prospects. *Science* **2015**, *349*, 255–260. [CrossRef]
32. Zhou, Z.H. *Machine Learning*; Springer Nature: Singapore, 2021.
33. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009, pp. 248–255.
34. Huh, M.; Agrawal, P.; Efros, A.A. What makes ImageNet good for transfer learning? *arXiv* **2016**, arXiv:1608.08614.
35. Bengio, Y. Deep learning of representations for unsupervised and transfer learning. In Proceedings of the ICML Workshop on Unsupervised and Transfer Learning, Bellevue, DC, USA, 2 July 2011; pp. 17–36.
36. Pang, S.; Xie, P.; Xu, D.; Meng, F.; Tao, X.; Li, B.; Li, Y.; Song, T. NDFTC: A new detection framework of tropical cyclones from meteorological satellite images with deep transfer learning. *Remote Sens.* **2021**, *13*, 1860. [CrossRef]
37. Deo, R.V.; Chandra, R.; Sharma, A. Stacked transfer learning for tropical cyclone intensity prediction. *arXiv* **2017**, arXiv:1708.06539.
38. Smith, M.; Toumi, R. Using video recognition to identify tropical cyclone positions. *Geophys. Res. Lett.* **2021**, *48*, e2020GL091912.
39. Zhuo, J.Y.; Tan, Z.M. A Deep-learning Reconstruction of Tropical Cyclone Size Metrics 1981–2017: Examining Trends. *J. Clim.* **2023**, *36*, 5103–5123. [CrossRef]
40. Fu, D.; Chang, P.; Liu, X. Using convolutional neural network to emulate seasonal tropical cyclone activity. *J. Adv. Model. Earth Syst.* **2023**, *15*, e2022MS003596. [CrossRef]
41. Hersbach, H.; Bell, B.; Berrisford, P.; Hirahara, S.; Horányi, A.; Muñoz-Sabater, J.; Nicolas, J.; Peubey, C.; Radu, R.; Schepers, D.; et al. The ERA5 global reanalysis. *Q. J. R. Meteorol. Soc.* **2020**, *146*, 1999–2049. [CrossRef]
42. Bian, G.F.; Nie, G.Z.; Qiu, X. How well is outer tropical cyclone size represented in the ERA5 reanalysis dataset? *Atmos. Res.* **2021**, *249*, 105339. [CrossRef]
43. Slocum, C.J.; Razin, M.N.; Knaff, J.A.; Stow, J.P. Does ERA5 mark a new era for resolving the tropical cyclone environment? *J. Clim.* **2022**, *35*, 7147–7164. [CrossRef]
44. Han, Z.; Yue, C.; Liu, C.; Gu, W.; Tang, Y.; Li, Y. Evaluation on the applicability of ERA5 reanalysis dataset to tropical cyclones affecting Shanghai. *Front. Earth Sci.* **2022**, *16*, 1025–1039. [CrossRef]
45. Gardoll, S.; Boucher, O. Classification of tropical cyclone containing images using a convolutional neural network: Performance and sensitivity to the learning dataset. *Geosci. Model Dev.* **2022**, *15*, 7051–7073. [CrossRef]
46. Bourdin, S.; Fromang, S.; Dulac, W.; Cattiaux, J.; Chauvin, F. Intercomparison of four algorithms for detecting tropical cyclones using ERA5. *Geosci. Model Dev.* **2022**, *15*, 6759–6786. [CrossRef]
47. Accarino, G.; Donno, D.; Immorlano, F.; Elia, D.; Aloisio, G. An Ensemble Machine Learning Approach for Tropical Cyclone Detection Using ERA5 Reanalysis Data. *arXiv* **2023**, arXiv:2306.07291.
48. Ito, K. Errors in tropical cyclone intensity forecast by RSMC Tokyo and statistical correction using environmental parameters. *SOLA* **2016**, *12*, 247–252. [CrossRef]

49. Chan, M.H.K.; Wong, W.K.; Au-Yeung, K.C. Machine learning in calibrating tropical cyclone intensity forecast of ECMWF EPS. *Meteorol. Appl.* **2021**, *28*, e2041. [CrossRef]
50. Faranda, D.; Messori, G.; Bourdin, S.; Vrac, M.; Thao, S.; Riboldi, J.; Fromang, S.; Yiou, P. Correcting biases in tropical cyclone intensities in low-resolution datasets using dynamical systems metrics. *Clim. Dyn.* **2023**, *61*, 4393–4409. [CrossRef]
51. Knapp, K.R.; Diamond, H.J.; Kossin, J.P.; Kruk, M.C.; Schreck, C.J., III. *International Best Track Archive for Climate Stewardship (IBTrACS) Project*, *Version 4*; NOAA National Centers for Environmental Information: Asheville, NC, USA, 2018; Volume 10. [CrossRef]
52. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
53. Cione, J.J.; Kalina, E.A.; Zhang, J.A.; Uhlhorn, E.W. Observations of air–sea interaction and intensity change in hurricanes. *Mon. Weather Rev.* **2013**, *141*, 2368–2382. [CrossRef]
54. Ghifary, M.; Kleijn, W.B.; Zhang, M. Domain adaptive neural networks for object recognition. In Proceedings of the PRICAI 2014: Trends in Artificial Intelligence: 13th Pacific Rim International Conference on Artificial Intelligence, Gold Coast, Australia, 1–5 December 2014; Proceedings 13; Springer: Berlin/Heidelberg, Germany, 2014; pp. 898–904.
55. Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; Darrell, T. Deep domain confusion: Maximizing for domain invariance. *arXiv* **2014**, arXiv:1412.3474.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.