I.J. Image, Graphics and Signal Processing, 2025, 2, 119-134
Published Online on April 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2025.02.08

Performance Comparison and Investigation of Tropical Cyclone Intensity Estimation from Satellite Images Using Deep Learning and Machine Learning

Md. Ahsan Rahat
Department of Computer Science and Engineering, Military Institute of Science and Technology Mirpur Cantonment, Dhaka 1216, Bangladesh
ORCID iD: https://orcid.org/0009-0009-2570-434X

Nusrat Sharmin*
Department of Computer Science and Engineering, Military Institute of Science and Technology Mirpur Cantonment, Dhaka 1216, Bangladesh
ORCID iD: https://orcid.org/0000-0003-2856-3561
*Corresponding Author

Fairooz Nawar Nawme
Department of Computer Science and Engineering, Military Institute of Science and Technology Mirpur Cantonment, Dhaka 1216, Bangladesh
ORCID iD: https://orcid.org/0009-0003-2450-3454

Sabbir Rahman
Department of Computer Science and Engineering, Military Institute of Science and Technology Mirpur Cantonment, Dhaka 1216, Bangladesh
ORCID iD: https://orcid.org/0009-0003-2450-3454

Received: 25 November, 2023; Revised: 23 January, 2024; Accepted: 12 March, 2024; Published: 08 April, 2025

Abstract: Tropical cyclones, considered extreme weather events, can cause significant damage to coastal areas, impacting millions of people and animals while also posing the risk of substantial economic losses. Traditionally, the Dvorak technique has been employed to assess the intensity of these cyclones, involving the visual analysis of satellite data to evaluate the storm's cloud patterns and strength. In recent years, various studies have explored the use of deep learning (DL) and machine learning (ML) techniques to estimate tropical cyclone intensity. However, there is a lack of research providing a comparative analysis that integrates both ML and DL approaches for the estimation of tropical cyclone intensity. This study looks into the use of ML and DL techniques to estimate the strength of tropical cyclones. On diverse datasets and satellite imagery, we study the usage of convolutional neural networks (CNN, VGG16, DenseNet), recurrent neural networks (LSTM), and other machine learning methods (XGBoost, CatBoost, SVM, DT). Our findings suggest that both ML and DL methods have substantial promise for improving tropical cyclone intensity estimation accuracy; however, in our case study, DL algorithms outperformed ML algorithms. This study investigates the utilization of ML and DL techniques in assessing the strength of tropical cyclones. Employing various datasets and satellite imagery, we examine the performance of convolutional neural networks (CNNs such as VGG16 and DenseNet), recurrent neural networks (LSTM), and other ML methods (XGBoost, CatBoost, SVM, DT). Our results indicate that both ML and DL approaches show significant promise in enhancing the accuracy of tropical cyclone intensity estimation. Nevertheless, in our specific case study, DL algorithms demonstrated superior performance compared to ML algorithms.

Index Terms: Machine Learning, Deep Learning, Intensity Estimation

1. Introduction

Tropical cyclones, known as typhoons or hurricanes [1], depending on their location, are powerful and destructive weather systems that can cause significant damage and loss of life. Effective disaster preparedness and response need precise predictions of cyclone intensity. By leveraging a variety of data sources, researchers have recently developed a number of methodologies for estimating the strength of tropical cyclones through their intensity. However, a tropical cyclone's intensity [2]is defined as its maximum sustained wind speed, which is frequently accompanied by various factors like central pressure and overall size. It offers a quantitative evaluation of the cyclone's potential for destruction as well as its capacity to produce storm surges, huge amounts of rain, and high waves.

It is crucial to comprehend and correctly anticipate cyclone intensity for a variety of reasons. Firstly, accurate predictions of cyclone intensity allow policymakers and emergency response teams to send out timely and pertinent warnings to potentially affected regions. Communities may therefore take the appropriate evacuation precautions, safeguard their infrastructure, and distribute resources wisely as a result. Early warnings based on precise intensity forecasts help to lessen the death toll and the damage the storm causes[1]. Secondly, proper cyclone intensity estimation aids in organizing emergency responses. Based on the anticipated intensity, emergency response teams may evaluate the possible effect and distribute resources accordingly. For instance, compared to a Category 5 cyclone, a Category 1 cyclone requires a drastically different level of readiness. Authorities can deploy people, pre-position emergency supplies, and prioritize response activities by making accurate predictions. Furthermore, accurate intensity predictions can help in determining the possible economic effects of a cyclone. This information may be used by enterprises and industries to safeguard assets, conduct preventative measures, and make wise operational decisions. Intensity predictions can be used by insurance firms and risk assessment organizations to estimate prospective financial obligations and premiums due to cyclone-related losses.

Estimating the intensity of tropical cyclones has typically been done using classic statistical techniques like the Dvorak approach. These techniques depend on actual relationships between certain cloud patterns and storm intensity when they include human interpretation of satellite images. When determining the severity of a cyclone, the Dvorak approach, for instance, examines the arrangement and appearance of cloud formations surrounding the cyclone's core. But these processes are arbitrary, and the knowledge and expertise of the analyst doing the interpretation can have an impact.

Both machine learning techniques and traditional statistical methods may be used to classify these systems. In contrast, automatic and objective methods for evaluating tropical cyclone strength are provided by machine learning techniques. These techniques make use of algorithms that can recognize patterns and relationships in data and generate predictions based on those patterns and relationships. This topic has been tackled using a variety of machine learning methods, such as artificial neural networks, decision trees, and support vector machines. These algorithms may be taught to spot intricate patterns and generate predictions utilizing a variety of data sources, including satellite images and meteorological data, by training them on big datasets.

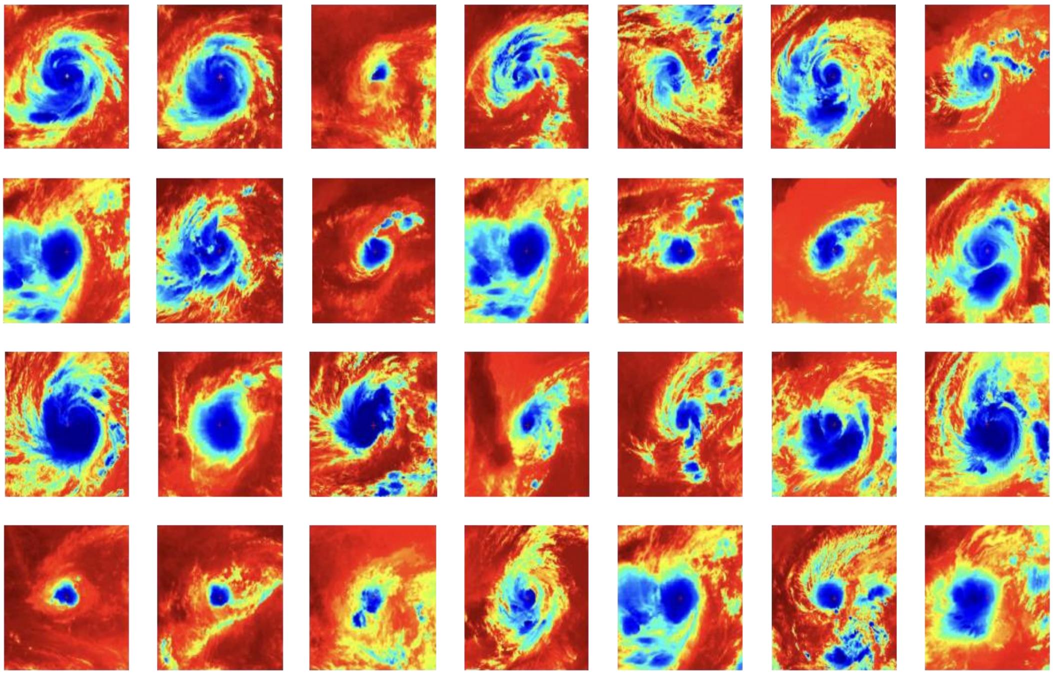

Fig. 1. Satellite images of Tropical Cyclones

Deep learning algorithms have various benefits when used to estimate tropical cyclone intensity. These models eliminate the need for human feature engineering by processing enormous volumes of data and automatically extracting useful features. Deep learning algorithms can also manage complicated and nonlinear interactions between input variables, leading to predictions that are more accurate. The models' accuracy is further improved by the ability to use a variety of data sources, including satellite images, atmospheric factors, and previous cyclone data. One of the most promising approaches for estimating tropical cyclone intensity is deep learning, a branch of machine learning inspired by the composition and operation of the human brain. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs), two types of deep learning techniques, have excelled in a number of fields, including computer vision and natural language processing. These algorithms excel at automatically recognizing complex characteristics and relationships in hierarchical data representations. Recently, deep learning algorithms have been successfully used to predict the intensity of tropical cyclones, with encouraging results. For instance, significant elements that correlate with cyclone intensity have been extracted from satellite images (see Figure 1) using convolutional neural networks. These algorithms may be taught to recognize patterns in images that are indicative of cyclone intensity by training on historical data with known cyclone intensity. On the other hand, recurrent neural networks have been used to evaluate time series data from weather sensors and meteorological observations because they are good at capturing temporal relationships.

While a significant amount of research has been dedicated to estimating the intensity of tropical cyclones, the predominant approach has been to treat it as a regression problem, wherein the focus is on computing the degree of cyclone intensity. Moreover, most of these studies have concentrated on either utilizing machine learning algorithms or employing deep learning techniques. Notably, there has been a lack of integration between these two approaches, and a comprehensive comparison of their performances has been lacking. Moreover, the performance comparison between machine learning and deep learning is of paramount importance as it provides insights into the effectiveness of different approaches in cyclone intensity estimation. Understanding how these techniques perform relative to each other allows researchers and practitioners to make informed decisions regarding model selection for specific tasks. This comparison sheds light on the strengths and weaknesses of each paradigm, guiding the appropriate application based on the nature of the data and the complexity of the problem at hand. Moreover, it contributes to the ongoing evolution of the field by identifying areas where one approach may outperform the other, fostering advancements in both machine learning and deep learning methodologies. Ultimately, a comprehensive performance evaluation is crucial for optimizing model selection and enhancing the overall efficiency of predictive modeling in diverse domains, in terms of cyclone intensity estimation.

In contrast to existing research, our study introduces a novel framework for tropical cyclone intensity estimation. We approach the problem by formulating it as a classification task on the same dataset, leveraging both machine learning and deep learning techniques. The objective is to recommend the most effective approach by thoroughly comparing the performance of these algorithms. Furthermore, we have conducted an empirical analysis of both methods to provide a comprehensive understanding of their capabilities. This study makes a significant contribution to the existing literature in the field by emphasizing the distinctions in the application of machine learning and deep learning algorithms for cyclone intensity estimation. The novelty of this paper is as follows:

1. We employ diverse machine learning and deep learning techniques, such as support vector machines, boosting algorithms, and convolutional neural networks (CNN), among others. However, for effective application of deep learning, we require a well-balanced and enhanced dataset, which is not available in our specific case. Consequently, we utilize augmentation methods to enrich and balance the dataset. To the best of our knowledge, there is currently no literature that offers a comparative analysis specifically based on satellite imagery. It is important to highlight that while considerable research has been devoted to predicting intensity using numerical data, there is a distinct lack of attention given to the exploration and examination of satellite images in this context.
2. This paper thoroughly explores and presents a comprehensive analysis of Machine Learning (ML) and Deep Learning (DL) algorithms for cyclone intensity estimation. Such analysis will help the community to develop any application for cyclone intensity estimation.

Section 2 gives a review of the related works of TC intensity estimation using machine learning and deep learning. Section 3 describes the problem, datasets, proposed algorithms, model training, and overall research methodology. Section 4 shows the evaluation results and analyzes the findings of the experiment. Section 5 summarizes the work and provides limitations of the study and future works.

2. Literature Review

In this section, we look into the use of deep learning and machine learning approaches, analyzing their strengths, drawbacks, and most recent advancements in the context of predicting tropical cyclone intensity. In Figure 2 we have summarized the various algorithms summary based on the existing literature.

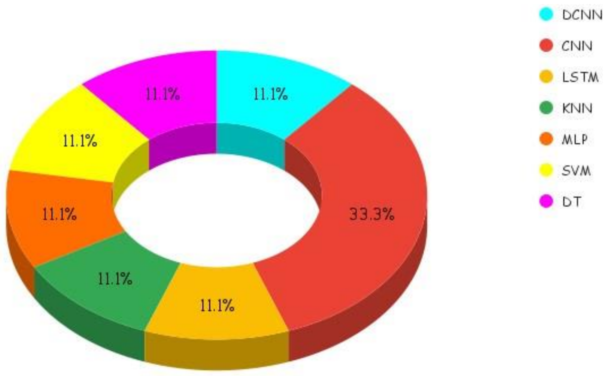

Fig. 2. ML and DL algorithms' ratio in existing model

2.1. Literature based on Numerical Model

Numerical models have been routinely utilized to assess the intensity of tropical cyclones. These models employ mathematical equations to replicate the physical processes that occur within tropical cyclones and give forecasts of their intensity. The Hurricane Weather Research and Forecast (HWRF) model by [2], the Geophysical Fluid Dynamics Laboratory (GFDL) Hurricane Model by [3], and the European Centre for Medium-Range Weather Forecasts (ECMWF) operational model by [4] are three of the most commonly used numerical models for estimating tropical cyclone intensity.

2.2. Literature based on Machine learning

[5] has used satellite data and machine learning techniques for automating the estimation of various meteorological parameters including tropical cyclone intensity estimation. To estimate the TC intensity, they used a K-nearest neighbor algorithm. The model takes as input a set of meteorological parameters derived from satellite data and predicts the intensity of the cyclone in terms of its maximum sustained wind speed. The result shows that overall RMSE is 18.1 knots and mae is 14.3 knots.

[6] developed a multilayer perceptron(MLP) model to compare the performance of the MLP model with other neural network and statistical models in forecasting the intensity of tropical cyclones and the result shows that the error is 4.70% using MLP model which is the minimum. They have used a dataset consisting of the cyclones that occurred in Arabian sea and Bay of Bengal of North Indian Ocean(NIO). The performance of the MLP model is tested for severe cyclonic storms of NIO like Sidr, Nargis, Aila etc.

[7] proposed a methodology for classifying changes in tropical cyclone intensity in the western North Pacific region using decision tree models. They also highlighted that data mining like decision tree can be effective in classifying changes in tropical cyclone intensity. The authors suggested that their model outperforms the existing methods and has the potential of improving tropical cyclone intensity forecasts.

[8] demonstrated a new method for diagnosing the potential for rapid intensification (RI) of tropical cyclones using kernel methods and reanalysis datasets. Rapid intensification is a phenomenon where the maximum sustained winds of a tropical cyclone increase by at least 30 knots (about 35 miles per hour) in 24 hours or less and it is important to predict as it can lead to more severe storms. They have used the kernel methods and SVM algorithm which was used to identify key factors that contribute to rapid intensification and their study allowed for an efficient diagnosis of the potential for rapid intensification of tropical cyclones, which will also help to improve TC forecasting.

2.3. Literature based on Deep learning

Deep learning and other machine learning approaches have recently been applied to estimate the severity of tropical cyclones. As inputs, these strategies use satellite remote sensing data and numerical model outputs to train a model that estimates the intensity of tropical cyclones. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long-Short Term Memory (LSTM) networks are among the most extensively utilized deep learning approaches, and Support Vector Machine(SVM), Decision tree(DT) are common machine learning techniques for assessing tropical storm intensity.

Deep convolutional neural networks (DCNNs) are proposed for forecasting the intensity of tropical cyclones by [9]. The authors compare the performance of the proposed DCNN model to two widely used models: the operational model of the European Centre for Medium-Range Weather Forecasts (ECMWF) and the Hurricane Weather Research and Forecast (HWRF). Microwave radiances and numerical weather forecast data are utilized as input features for the DCNN model to collect information about the current condition and evolution of cyclones.

Experiment findings reveal that the proposed DCNN model outperforms the ECMWF and HWRF models in terms of tropical cyclone intensity prediction accuracy. The DCNN model has a smaller mean absolute error (MAE) than the ECMWF and HWRF models, indicating greater forecast performance. Furthermore, the authors show the efficacy of employing microwave radiances as input characteristics for the DCNN model. The study's findings suggest that deep learning technologies have the potential to increase tropical cyclone intensity prediction accuracy.

[9] research describes a deep-learning approach for forecasting tropical cyclone intensity. The model was trained and evaluated using real-world tropical cyclone intensity data and was built on Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) networks. In terms of prediction accuracy, the proposed model beat standard statistical models such as the persistence and linear regression models.

[10] research describes the development and evaluation of a deep learning-based system for estimating the intensity of tropical cyclones. The authors demonstrated the superiority of the suggested method over conventional intensity prediction models by using a convolutional neural network (CNN) to learn the link between satellite-observed variables and the related tropical cyclone intensity.

[11] describes a deep learning-based model for estimating the intensity of tropical cyclones using Himawari-8 satellite cloud products. The authors employ a Convolutional Neural Network (CNN) to predict the strength of tropical cyclones and evaluate the model's performance using mean absolute error (MAE) and mean square error (MSE) metrics. The results reveal that the model outperforms existing techniques, with lower MAE and MSE values, highlighting the promise of employing deep learning for tropical cyclone intensity estimation.

[12]'s study offers a deep learning model for estimating tropical cyclone intensity using infrared satellite images. The model is made up of a Convolutional Neural Network (CNN) that was trained on an infrared satellite image dataset. The authors compare the model's performance to that of a classic gradient descent method and discover that the deep learning model surpasses the old method in both accuracy and computing efficiency. Deep learning has the potential to greatly improve tropical storm strength prediction, according to the findings.

2.4. Research Gap

While extensive research has been carried out to estimate the intensity of tropical cyclones, the prevailing approach has been to formulate these estimations as regression problems. Additionally, the majority of research efforts have concentrated on either machine learning algorithms or deep learning techniques. Remarkably, there has been a notable absence of studies that integrate these two approaches and systematically compare their performance.

In response to this gap, we introduce a novel framework for tropical cyclone intensity estimation. This involves framing the problem as a classification task using the same dataset and employing both machine learning and deep learning techniques. Our aim is to recommend the most effective approach by conducting a detailed comparison of the algorithms' performance.

3. Methodology

3.1. Intensity Estimation of Tropical Cyclones as Classification Problem

The purpose of formulating tropical cyclone intensity estimation as a classification issue is to classify the intensity of the cyclone into a certain category based on its features. In our research, we split the intensity level into three categories depending on wind speed: tropical depression (t_d), tropical storm (t_s), and typhoon (t_t).

The problem is stated as follows: Using a classification algorithm, classify the intensity of a tropical cyclone into one of the specified categories (e.g., tropical depression, tropical storm, typhoon) based on wind speed collected from satellite imagery and other data sources.

Assume the tropical cyclone set is $X = \{tc1, tc2, tc3, ...\}$ and the collection of images for each tropical cyclone is $images = \{tc1\ img, tc2\ img, tc3\ img..........\}$ where, tc1 img=satellite image of tropical cyclone 1.

To estimate the intensity based on the wind speed, set $wind\ speed = \{tc1\ wisd, tc2\ wisd, tc3\ wisd...........\}$(where, tc1 wisd=wind speed of tropical cyclone 1 on corresponding satellite images of tropical cyclone 1.

The designation is then divided into three classes: (i)if the tropical cyclone is a tropical depression, it should be in class 0; (ii)if it is a tropical storm, it should be in class 1; and (iii) if it is a Typhoon, it should be in class 2. As a result, the final wind speed is, wind speed = {class 0, class 1, class 2} if the corresponding image set contains the images of a tropical depression then a tropical storm, and then the image of a typhoon, etc. The wind speed set is basically the label of images of tropical cyclones. The model is trained on labeled data, with each satellite image and label identifying the intensity category of the tropical cyclone. The goal is to train the model to predict the intensity category of new, previously unseen cyclones.

3.2. Proposed Methodology

For the purpose of comparison, we have selected eight machine learning and deep learning models, namely SVM, XGBoost, CatBoost, Decision Tree, CNN, LSTM, VGG16, and DenseNet. These specific models were chosen based on their documented impact in the literature regarding cyclone intensity estimation. Each of these models has demonstrated significance within this domain, highlighting their effectiveness and contributions in advancing the understanding and prediction of cyclone intensity. This selection ensures a comprehensive assessment, leveraging the strengths and unique features of each model to provide valuable insights for the improvement of cyclone intensity forecasting and management strategies. In Figure 3, we demonstrate the proposed methodology to detect the cyclone, and followed by in Algorithm 1 we have listed the steps of the methodology.

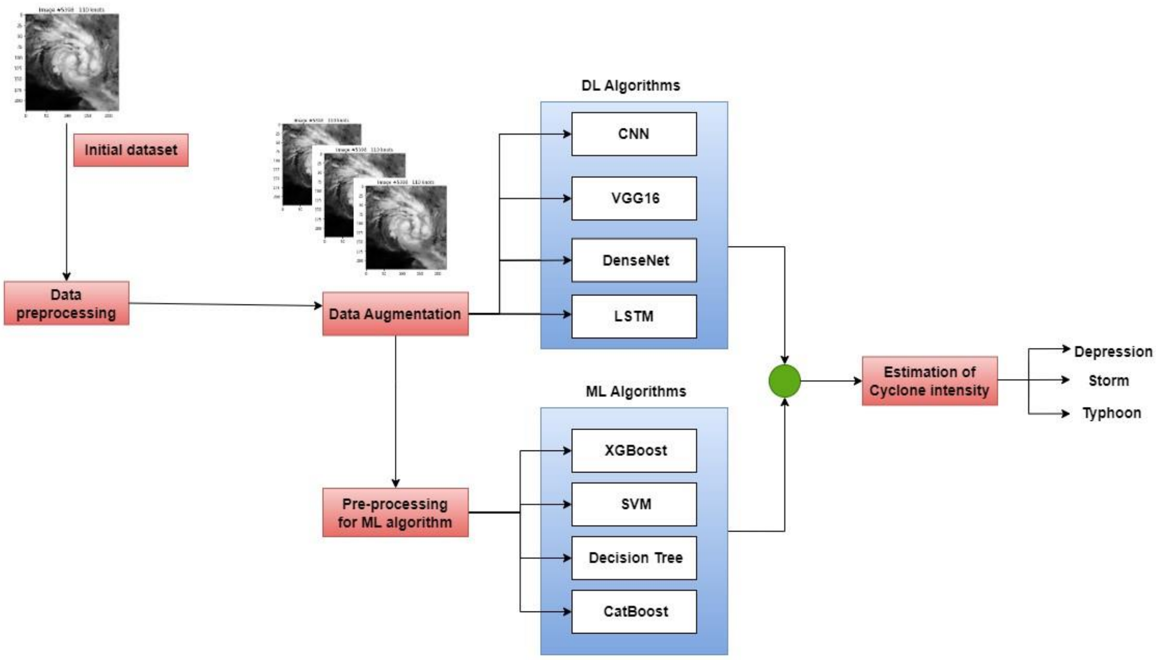

Fig. 3. Proposed Methodology

Algorithm 1 Proposed Methodology

1: Load the dataset of Tropical Cyclone images, including images of Tropical Depression, images of Tropical Storms, and images of Typhoon.
2: Pre-processing the dataset.
3: Augmentation
4: Split the dataset into training, validation, and test sets.
5: Define the Deep learning architecture to be used, including CNN, VGG16, LSTM, and DenseNet. 6: while each algorithm has been used do
7: Select an algorithm.
8: if it's an ML algorithm then
9: does pre-processing for the ML algorithm.
10: end if
11: Train the algorithm using the training set of images, using the corresponding loss function of the chosen algorithm.
12: Validate the algorithm using the validation set of images.
13: Test the algorithm using the test set of images and evaluate its performance.
14: end while
15: Compare the performance of different algorithms and choose the best one for estimating the intensity of Tropical Cyclone task

3.3. Dataset

For both machine learning and deep learning, we have used Hurricane Satellite (HURSAT) Data ([13]matic Data Center (NCDC), n.d.) for hurricane images and HURDAT2 database ([14] Data Archive, n.d.) for best track data. Mainly the best track data is used for labeling of the images. HURSAT dataset contains satellite images of hurricanes from around the world in NetCDF format. In these images, the center of each hurricane is in the middle of the image. The HURDAT2 database provides 'besttrack.csv' which contains information of the hurricanes, more precisely their wind speeds at 6-hour intervals. We have used these wind speeds for labeling the satellite images, but the HURSAT database has images of all hurricanes whereas best-track data contains wind speeds only for the hurricanes from the Atlantic and Pacific basin. So we have kept only the images whose wind speeds are available in besttrack data. The image file provides the name of the hurricane, time and date. So for each image, the corresponding wind speed is retrieved by searching the csv file by name, date and time. Thus the images are properly labeled by their wind speed. Finally, we have appended the images and labels to numpy arrays which are actually used for training and testing the models. In our experiments, we used data of cyclones from 2012 to 2016; the total number of data samples is 5573.

The dataset includes images of Tropical Depression, images of Tropical Storm and images of Typhoon.

1. *Tropical Depression:* A tropical depression is a weather phenomenon that has low pressure and winds that are less than 34 knots. It is the first stage of a tropical cyclone's development.

2. *Tropical Storm:* A tropical storm is a meteorological system with low pressure and wind speeds ranging from 34 to 64 knots. It is the second stage of a tropical cyclone's development, and when its winds reach this speed, it is usually designated as a tropical storm.
3. *Typhoon:* A tropical cyclone in the western Pacific Ocean is referred to as a typhoon. It is the same as a hurricane in the Atlantic Ocean, with wind speeds of 64 knots or greater.

3.4. Data Pre-processing

Data pre-processing has been explained in detail in the following two sections. *First* we will elaborate on the common steps used in data pre-processing for machine learning and deep learning and next we will discuss the additional steps used in machine learning.

Data Pre-processing Algorithm (for both ML and DL)

Algorithm 2 Data Pre-processing Algorithm (for both ML and DL)

1: Load the dataset of Tropical Cyclone images, including images of Tropical Depressions, images of Tropical Storms, and images of Typhoon.
2: Classify the labels into three classes(Tropical Depression, Tropical Storm, and Typhoon) depending on wind speed. 3: while each image of the dataset do
4: Resize the images.
5: Convert the grayscale image to RGB if it is not already in that format.
6: Convert the image to an 8-bit unsigned integer numpy array. 7: end while

3.4.1 Data Preprocessing Algorithm for Machine Learning Algorithm

Algorithm 3 Data Preprocessing Algorithm for Machine Learning Algorithm

1: Load the dataset of Tropical Cyclone images, including images of Tropical Depression, images of Tropical Storms, and images of Typhoons.
2: Convert labels to NumPy arrays.
3: while each image of the dataset do
4: Calculate hog (histogram of oriented gradients) features of the image.
5: Calculate the intensity value for a larger region called block and thus all cell-level histograms within each block are normalized.
6: Combine the gray images and their hog features(thus a three-dimensional array is flattened into a one-dimensional array).
7: end while

3.5. Data Augmentation

Before training the model, an examination of the data was performed. The findings are in table 1.

Table 1. Data sample description (Before Augmentation)

Sr	*Tropical Cyclone Type*	*Number Of data samples*
1	Tropical Depression	2062
2	Tropical Storm	2368
3	Typhoon	1143
Total number of data samples		5573

The table illustrates that in the dataset, weak tropical cyclones (tropical depressions and tropical storms) outweigh powerful tropical cyclones (typhoons). This is hardly surprising given that tropical depressions and tropical storms occur far more frequently than typhoons. This disparity, however, indicates that the dataset is imbalanced, which causes the neural network to perform badly during typhoons. *Model.py* uses Keras's *ImageDataGenerator* to enhance data from existing typhoons in the dataset to balance the dataset. Including this supplemented data in model training increases model performance dramatically, notably when calculating wind speeds for typhoons. Data augmentation is crucial in deep learning as it enhances model generalization by artificially expanding the training dataset. By applying diverse transformations to existing data, such as rotations or flips, the model learns to be more robust and invariant to variations in the input. This helps prevent overfitting and improves the model's ability to perform well on unseen data, ultimately contributing to better overall performance and reliability in deep learning tasks. Note that Data augmentation requires 2-3 minutes to augment the data.

If the data is already in a balanced state, it suggests that each class or category within the dataset has a relatively equal representation. This balance is advantageous as it helps prevent the model from being biased toward one specific class during training, promoting fair and accurate learning across all categories. While data augmentation is commonly used to address class imbalances, in your case, the balanced state of the data provides a solid foundation for training deep learning models without the need for additional balancing techniques.

The dataset after augmentation looks like this table 2.

Table 2. Data sample description (After Augmentation)

Sr	*Tropical Cyclone Type*	*Number Of data samples*
1	Tropical Depression	2062
2	Tropical Storm	3206
3	Typhoon	5878
Total number of data samples		11146

Algorithm 4 Augmentation Algorithm

1: Load the initial dataset of Tropical Cyclone images, including images of Tropical Depressions, images of Tropical Storm, and images of Typhoon.
2: while each image of initial dataset do
3: if the image is a tropical depression(wind speed less or equal 33 knots) then 4: generates 0 images.
5: else if the image is a tropical storm(wind speed greater than 33 knots and less or equal 64 knots) then 6: generate 1 image.
7: else
8: generate 6 images(Typhoons' image).
9: end if 10: end while

Steps of data augmentation is as follows:

Algorithm 5 Augmentation Algorithm

1: Import Libraries: Import necessary libraries, including the image module from the Keras preprocessing library.
2: Initialize Flip Generator: Create a flip generator using the ImageDataGenerator class. Specify parameters for horizontal and vertical flips (e.g., horizontal flip=True, vertical flip=True).
3: Initialize Rotate Generator: Create a rotate generator using the ImageDataGenerator class. Specify parameters for rotation, such as rotation range=360 and fill mode='nearest'.

A details of data augmentation in order to balance the dataset has been included here.

Algorithm 6 Augmentation Algorithm

1: Loop each images in the set to augment
2: Reshape image for generator
3: Reset the number of augmented images have been created to zero
4: Generate 2 new images if the image is of a tropical cyclone between 50 and 75 knots
5: Generate 6 new images if the image is of a tropical cyclone between 75 and 100 knots
6: Generate 12 new images if the image is of a tropical cyclone greater than or equal to 100 knots 7: Convert lists of images/labels into numpy arrays

When outside information is used to train a model, it undermines the independence of the training data, which is known as data leaking. The most important but frequently disregarded requirement is total independence between the training and testing datasets. To guarantee an objective assessment of the model, the values in the testing set must not be influenced by any data in the training set. Note that In this work, test data and train data are totally independent which leads no data leakage. It's essential to highlight that in this study, the test data and training data are entirely independent, ensuring the absence of any data leakage.

3.6. Machine Learning Approaches

1. *Support Vector Machine(SVM)*: It is a well-known machine learning method that can do classification and regression tasks. SVMs function by locating a hyperplane in a high-dimensional space that can classify the data. SVMs are supervised learning algorithms, which means they require labeled data to train on. SVMs seek the hyperplane that optimizes the margin between the distinct classes in classification problems. The margin measures the distance between the hyperplane and the nearest data points in each class.
2. *eXtreme Gradient Boosting(XGBoost)*: It is a highly effective machine-learning method that is frequently used for classification and regression problems. XGBoost is a gradient-boosting method that uses decision trees as base learners. XGBoost is a supervised learning method that must be trained on labeled data. The approach works by adding decision trees to a model iteratively, correcting the faults of earlier trees. This is referred to as gradient boosting. Many hyperparameters in XGBoost can be modified to optimize the model, such as the learning rate, maximum depth of the trees, number of trees, and regularization parameters.

3. *CatBoost*: CatBoost is a machine learning method that works especially well with categorical data. It is a gradientboosting approach for working with high-dimensional categorical features. The approach is similar to XGBoost in that it adds decision trees to a model iteratively, correcting the faults of earlier trees. CatBoost employs ”Ordered Boosting,” a unique technique for analyzing category information. By combining categorical features, this technique can efficiently handle high-dimensional category data. CatBoost resolves missing values automatically and can cope with text data without the need for feature engineering.
4. *Decision Tree*: A decision tree is a machine-learning technique that can be used for classification as well as regression. It is an easy-to-understand and comprehend paradigm, making it popular in both academic and business settings. Decision trees are a sort of supervised learning algorithm that must be trained on labeled data. The method operates by recursively separating the data into subsets based on the data’s properties. Each split is chosen to reduce the amount of impurity in the ensuing subgroups.

3.7. Deep Learning Approaches

Convolutional Neural Network (CNN): A Convolutional Neural Network (CNN) is a type of deep learning neural network that excels at picture classification and processing. A CNN’s architecture is intended to automatically and adaptively learn hierarchical characteristics from input images, beginning with lower-level features like edges and curves and progressing to higher-level features like object pieces and finally the object as a whole.

The main components of a CNN are:

1. *Convolutional Layers:* Multiple filters are applied to the input image at this layer to detect features such as edges, corners, and textures. The filters generate a feature map by sliding across the image and applying a dot product between the filter and the area of the image that it is currently covering.
2. *Pooling Layers:* These layers are used to reduce the size of the feature maps while keeping the most critical information.
3. *Activation Functions:* To inject nonlinearity into the network, nonlinear activation functions such as ReLU (rectified linear unit) are applied to the output of each neuron.
4. *Fully Connected Layers:* In this layer, the preceding layers’ features are flattened and then connected to a number of artificial neurons, which produce the prediction.
5. *Dropout:* Dropout is a regularization technique used to reduce overfitting in deep neural networks, which is a prevalent problem. During training, certain neurons are deleted at random, thus shrinking the network and forcing it to learn more robust characteristics.

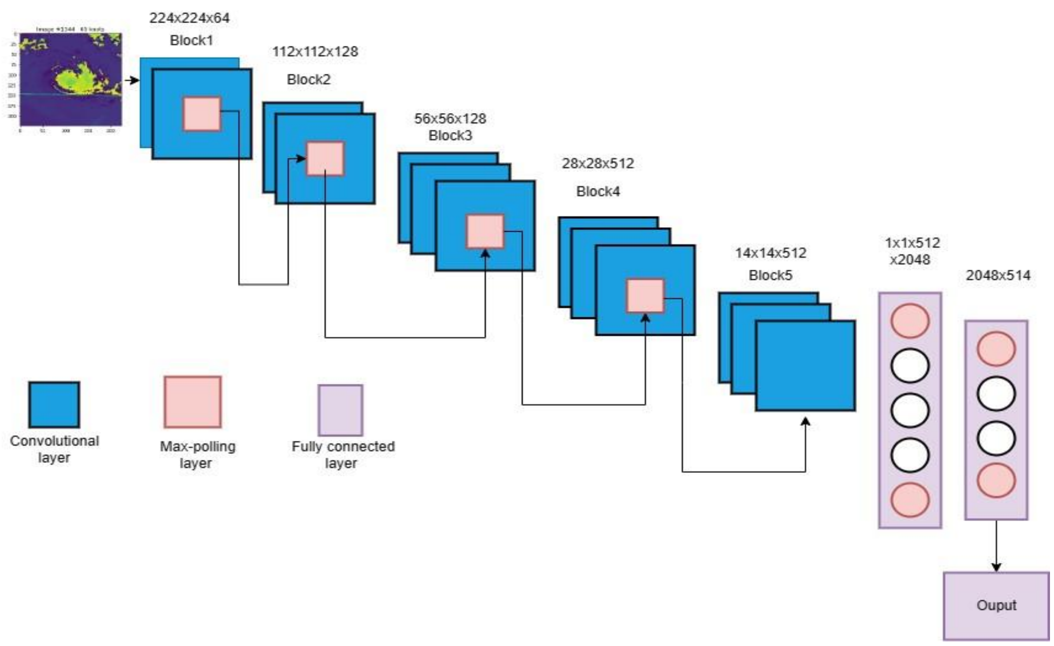

Fig. 4. CNN Architecture

Visual Geometry Group 16(VGG16): VGG16 is a deep learning convolutional neural network architecture. The network architecture is made up of 16 trainable layers, 13 convolutional layers, and 3 fully connected layers. The use of many tiny 3x3 convolutional filters in the VGG16 architecture allows the network to learn detailed feature representations at multiple sizes. The VGG16 network was trained on the massive ImageNet dataset, which contains over one million photos and 1000 image classifications. This training enables the network to learn high-level semantic notions required for picture classification tasks, such as edges, textures, and forms. VGG16 is a well-known deep learning architecture that is commonly utilized in image classification and object recognition applications in computer vision. The design has been refined and used in a variety of other disciplines, including medical imaging and video analysis.

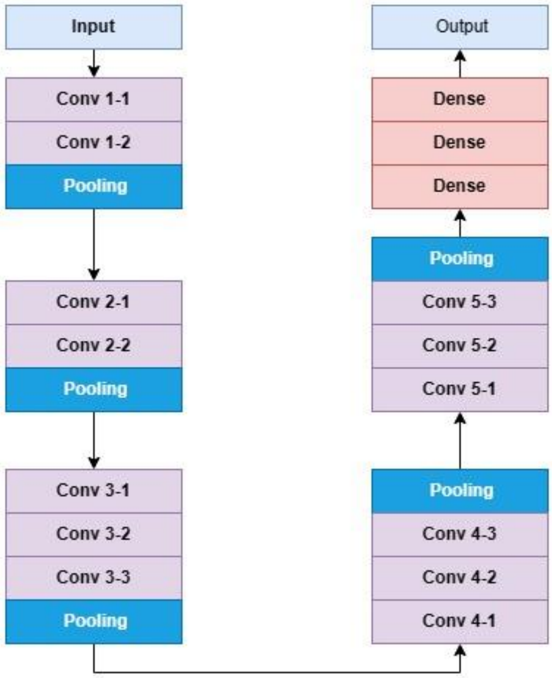

Fig. 5. VGG16 Architecture

Long Short-Term Memory(LSTM): Long Short-Term Memory (LSTM) is a Recurrent Neural Network (RNN) architecture that is well suited for modeling sequential data such as time series or natural language. The incorporation of memory cells, gates, and external memory is the key difference between LSTM and regular RNNs. The memory cells hold the sequence's key information, while the gates control the flow of information into and out of the memory cells. This enables LSTMs to deal with long-term dependencies in sequential data, which is a major problem in standard RNNs. Speech recognition, natural language processing, and time series forecasting have all made extensive use of LSTMs. *DenseNet:* DenseNet is an architecture for convolutional neural networks. It is built on the concept of dense connections, in which each layer is feed-forward coupled to every other layer. The primary aim for dense connections is to alleviate the vanishing gradient problem that arises in deep neural networks when activations become diluted as they travel through the network. The following are the essential features of the DenseNet architecture:

1. *Dense blocks:* These are convolutional layer groups that are closely coupled to one another.
2. *Transition layers:* These are used to reduce the number of feature maps that must be generated between dense blocks, allowing for more efficient use of computer resources.
3. *Global average pooling:* This is used to decrease the number of feature mappings to a 1x1xC tensor. As a result, the features can be represented in a more compact manner.

DenseNets have been proven to reach state-of-the-art performance in a variety of computer vision applications, including image classification, semantic segmentation, and object recognition.

3.8. Model Training

3.8.1 ML Model training

The dataset is split into train and test sets. The training dataset is used for training models using some machine learning algorithms (SVM, DT, CatBoost, XGBoost) and the testing set is used for evaluating the performance of the model. In the XGBoost algorithm, we have done hyperparameter tuning for increasing the performance. We have used a learning rate of 0.1 and n-estimators of 1000 which means the algorithm will try to run 1000 times. Max depth is set to 5 as higher depth will allow over-fitting. Subsample and colsample by tree both values are set to 0.8. All these parameters control the overfitting of the model. The objective is used to minimize the loss function. We have used 'multi:softprob', which is used for multiclass classification and returns the predicted probability of each class for each observation. The seed is set to 27. For CatBoost, we have kept all the default parameters. For the decision tree classifier, we have used 'entropy' as the criterion, and max depth is set to 5. Entropy is for information gain, the entropy is used as the impurity measure to decide how to split a node such that it gives the most amount of information gain. For SVM, we have tried kernel values linear, poly, rbf, and sigmoid. Among them, the linear kernel gives higher accuracy.

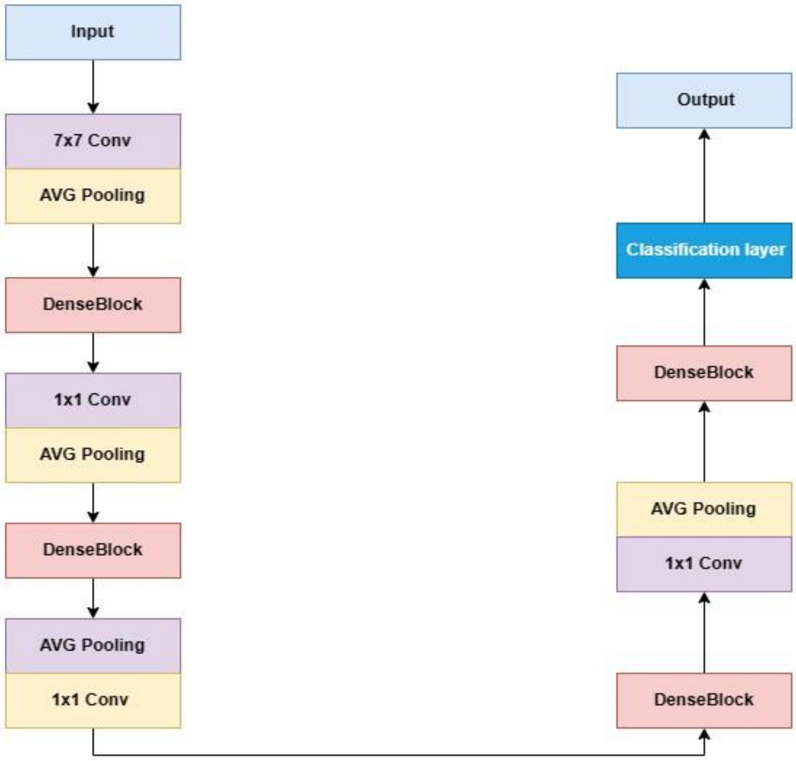

Fig. 6. DenseNet Architecture

3.8.2 DL Model training

Hyper parameter Tuning: Establishing appropriate hyperparameter values are crucial as it directly influences the performance of the resulting model during the training process. The process of finding the appropriate collection of hyperparameters for a deep learning model in order to improve its performance is referred to as hyperparameter tuning.

We incorporated various functions, such as Sequential, Dropout, Dense, Adam optimizer, etc., for hyperparameter tuning. . For model compilation, we utilized the Adam optimizer and employed the RMASE as the performance metric to calculate the loss. To train the model, an epoch was implemented in the learning algorithm, allowing each sample in the training dataset an opportunity to update the internal model once during an epoch. In our configuration, we employed 50 epochs and a batch size of 64 with learning 0.001.

Train test Split the dataset undergoes division into training and test sets, with the test set primarily serving as a validation subset. Utilizing various deep learning algorithms such as CNN, VGG16, LSTM, and DT, the models are trained on the training dataset. The experimental setup involves an 80% allocation for training, 10% for validation, and another 10% for testing, determined through empirical analysis to yield optimal results.

Hyperparameters are parameters that are established prior to training rather than learned during training, such as the learning rate, the number of hidden layers, the number of neurons in each layer, the regularization strength, and others. The method of hyperparameter tuning entails searching for the best collection of hyperparameters by experimenting with different combinations and evaluating the model's performance on a validation set. Hyperparameters of this implememtation as followings:

Table 3. Hyperparameters

Learning Rate	*Epochs*	*Optimizer*	*Activation Function*
0.001	50	Adam	Sigmoid

Adam optimizer: The Adam optimizer is a prominent deep-learning optimization technique that is designed to update network weights effectively depending on the gradients of the loss function. The Adam optimizer, which stands for Adaptive Moment Estimation, is a combination of two previous optimization techniques, AdaGrad and RMSProp, that seeks to overcome their shortcomings. The Adam optimizer maintains a running average of the gradient's first and second moments and adjusts the learning rate for each weight based on these moments. Its adjustable learning rate enables Adam Optimizer to converge rapidly and efficiently, and it has been demonstrated to operate well in a variety of deep-learning applications.

The rationale for selecting the Adam optimizer lies in its efficiency and effectiveness for optimizing deep learning models. Adam combines the advantages of both the RMSprop and momentum optimization algorithms. It adapts the learning rates individually for each parameter, providing adaptive updates that result in faster convergence and improved performance on a wide range of tasks. The algorithm's ability to handle sparse gradients and its general robustness make it a popular choice for training deep neural networks, contributing to its widespread adoption in various machine learning applications. *Sigmoid function:* A mathematical function that transfers every input value to a value between 0 and 1 is known as the sigmoid activation function. The formula is as follows:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{1}$$

where x is the input value and exp is the exponential function. The sigmoid function is often employed as an activation function in neural networks, particularly in binary classification issues with output probabilities ranging from 0 to 1. Because the sigmoid function is differentiable, efficient backpropagation during the training process is possible. Unfortunately, it is susceptible to the vanishing gradient problem, which makes deep network training challenging.

Learning rate and epoch Selecting the learning rate is a delicate task, as it determines the extent of model adjustments during weight updates. Too small a value may lead to slow convergence or getting stuck, while a value that is too large may result in learning sub-optimal weights rapidly or an unstable training process. The learning rate, a crucial hyperparameter in neural network training, typically has a small positive value, commonly within the range of 0.0 to 1.0. Through statistical testing, we determined 0.01 as the optimal learning rate. Opting for 10 epochs was based on the observation that, after this point, the results remained consistent, indicating that further epochs did not significantly improve the model performance.

3.9. Evaluation Metrics

Root Mean Squared Error (RMSE): The Root Mean Squared Error (RMSE) is a popular evaluation statistic for calculating the difference between actual and anticipated values. The square root of the average of the squared discrepancies between the actual and anticipated values is used to calculate it. The equation is:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\text{actual\_value}_i - \text{predicted\_value}_i)^2} \tag{2}$$

Where: n is the number of samples, actual value is the actual value of the sample, predicted value is the predicted value of the sample.

Mean Squared Error (MSE): Mean Squared Error (MSE) is a machine learning regression statistic that quantifies the average of the squared differences between the actual and predicted values. It is defined as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\text{y}_\text{i} - \hat{y_i})^2 \tag{3}$$

Table 4. Training Details for CNN

Epoch No	Phase	RMSE	loss
1	train	0.231	0.356
1	val	0.234	0.30
2	train	0.22	0.487
2	val	0.27	0.361
3	train	0.30	0.367
3	val	0.23	0.35
4	train	0.21	0.40
4	val	0.20	0.35
5	train	0.24	0.34
5	val	0.26	0.34772
6	train	0.28	0.35
6	val	0.29	0.348641
7	train	0.29	0.31
7	val	0.29	0.30
8	train	0.29	0.31
8	val	0.29	0.34
9	train	0.0.29	0.39
9	val	0.0.27	0.30
10	train	0.295	0.30
10	val	0.297	0.31

where y i is the actual value and yˆ i is the predicted value and n is the total number of observations. The MSE is a scalar metric that indicates the quality of the model's predictions by measuring the average deviation of the forecasts from the actual values. Higher MSE values imply a poorer model fit, while lower MSE values suggest a better model fit, which is $RMSE = \sqrt{MSE}$

Normalized Root Mean Squared Error(NRMSE) The NRMSE is a percentage that can be used to compare the performance of several models across different datasets or variables. It has a value between 0 and 100%, with lower values signifying better performance. NRMSE is calculated as follows:

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}}{max(y) - min(y)} \quad (4)$$

Where RMSE represents the root mean squared error, max(y) represents the greatest value of the dependent variable, and min(y) represents the minimum value of the dependent variable.

4. Experimental Results

4.1. MSE, RMSE, NRMSE of Deep Learning approaches

The performance of various machine learning approaches for tropical cyclone intensity prediction was evaluated using the Mean Square Error (MSE), Root Mean Square Error (RMSE), and Normalized Root Mean Square Error (NRMSE). The models assessed in this study include VGG16, CNN, DenseNet, and LSTM.

Table 4 presents the MSE, RMSE, and NRMSE values obtained from training and validation datasets for each model. These metrics provide insights into the accuracy and reliability of the intensity predictions. Lower values indicate better performance, as they reflect smaller differences between the predicted and actual cyclone intensities.

The results show that all evaluated models achieved relatively low values for MSE, RMSE, and NRMSE, demonstrating their effectiveness in predicting tropical cyclone intensities. Specifically, VGG16, CNN, DenseNet, and LSTM consistently performed well across all evaluation metrics. However, slight variations were observed among the models, highlighting their unique characteristics and performance nuances.

Table 5. Training Details for Densenet

Epoch No	Phase	RMSE	loss
1	train	0.231	0.356
1	val	0.234	0.30
2	train	0.22	0.487
2	val	0.29	0.361
3	train	0.27	0.367
3	val	0.29	0.35
4	train	0.21	0.40
4	val	0.26	0.35
5	train	0.25	0.34
5	val	0.26	0.34772
6	train	0.29	0.35
6	val	0.29	0.34
7	train	0.27	0.27
7	val	0.28	0.27
8	train	0.28	0.31
8	val	0.281	0.35
9	train	0.0.281	0.35
9	val	0.29	0.30
10	train	0.291	0.32
10	val	0.291	0.31

4.2. MSE, RMSE, NRMSE of Machine Learning approaches:

The MSE, RMSE, and NRMSE values are given for each machine learning method based on the evaluation of further machine learning algorithms, including Support Vector Machine (SVM), Decision Tree, CatBoost, and XGBoost. The average squared difference between the expected and actual cyclone intensity is represented by the MSE.

The average prediction error is shown by the RMSE, which is the square root of the MSE. By dividing the RMSE by the range of the actual cyclone intensities, the RMSE is normalized to get the NRMSE.

The findings in Table 5 show that all four machine learning methods did an adequate task of predicting the intensity of tropical cyclones. The MSE values of SVM and Decision Tree were comparable, while Decision Tree's RMSE and NRMSE were somewhat higher than SVM's. Compared to SVM and Decision Tree, CatBoost has reduced MSE, RMSE, and NRMSE values. Among the examined models, XGBoost beat all other algorithms, having the lowest MSE, RMSE, and NRMSE values.

These findings suggest that CatBoost and XGBoost are particularly effective for tropical cyclone intensity prediction, exhibiting superior accuracy and lower prediction errors. The lower MSE, RMSE, and NRMSE values obtained from these algorithms indicate their ability to capture complex relationships within the dataset, resulting in improved intensity predictions.

Table 6. MSE, RMSE, NRMSE of DL approaches

sl		Algorithms		Training		Validation			Testing	
1	VGG16	0.060	0.245	0.14	0.0604	0.245	0.144	0.060	0.245	0.144
2	CNN	0.086	0.293	0.17	0.0867	0.294	0.172	0.086	0.294	0.172
3	DenseNet	0.084	0.291	0.17	0.0857	0.292	0.171	0.085	0.292	0.171
4	LSTM	0.083	0.288	0.16	0.0854	0.292	0.169	0.085	0.292	0.169

Table 7. MSE, RMSE, NRMSE of ML approaches

Sl.	*Algorithms Name*	*MSE*	*RMSE*	*NRMSE(%)*
1	SVM	0.304	0.551	0.32
2	Decision Tree	0.304	0.596	0.35
3	CatBoost	0.203	0.451	0.26
4	XGBoost	0.176	0.419	0.25

4.3. MSE, RMSE of Machine Learning approaches:

Table 6 provides the MSE (Mean Squared Error) and RMSE (Root Mean Square Error) values for each algorithm. These metrics, which are expressed in knots, represent the average squared difference between projected and actual cyclone intensities as well as the average prediction error.

The findings show that the four machine learning algorithms perform differently when forecasting the intensity of tropical cyclones. With an MSE of 178.584 knots and an RMSE of 13.364 knots, SVM obtains the lowest MSE. Higher MSE and RMSE values of 489.126 knots and 22.116 knots, respectively, are displayed by Decision Tree. As opposed to SVM and Decision Tree, CatBoost and XGBoost exhibit greater performance with lower MSE and RMSE values.

CatBoost achieves the lowest MSE of 141.572 knots and the lowest RMSE of 11.898 knots among the evaluated algorithms. XGBoost also performs well, with an MSE of 153.313 knots and an RMSE of 12.382 knots.

These findings suggest that CatBoost and XGBoost have better accuracy in predicting tropical cyclone intensities, as they exhibit lower MSE and RMSE values. The lower MSE and RMSE values indicate that the predicted intensities are closer to the actual intensities for these algorithms.

Table 8. MSE, RMSE of ML approaches

Sl.	*Algorithms Name*	*MSE(knots)*	*RMSE(knots)*
1	SVM	178.584	13.364
2	Decision Tree	489.126	22.116
3	CatBoost	141.572	11.898
4	XGBoost	153.313	12.382

The results show that deep learning algorithms outperform machine learning methods. Because augmenting enriches the dataset while also converting an imbalanced dataset to a balanced one, and we know that deep learning algorithms learn high-level features progressively and hence do feature extraction on their own, they will certainly outperform machine learning methods. When comparing deep learning algorithms, VGG16 outperforms the other three. Table 3 suggests VGG16 has the lowest NRMSE in both training and validation (0.144%). Other three algorithms (CNN, DenseNet, and LSTM) produce very identical results. On the other hand from Table 7. It is clear that, among machine learning algorithms, XGBoost and CatBoost outperform SVM and decision trees.

The results demonstrate that boosting algorithms are more effective at estimating TC intensity. Using the same datasets as ours, [15] achieved an RMSE of 14.3 knots, whereas our model achieved an RMSE of 11.9 knots. It is due to the data augmentation and preprocessing of machine learning algorithms. The images are initially in grayscale format. We generated the HOG (Histogram of Oriented Gradients) characteristics of the images and mixed them with the original color features in the preprocessing step of the machine-learning approach. As a result, the model’s accuracy has improved, and we have a lower RMSE.

5. Conclusion

In this study, we have designed a model to estimate the intensity of tropical cyclones using machine learning and deep learning techniques. We initially defined our challenge as a classification problem, which means that we divided our dataset into three groups (Tropical depression, tropical storm, typhoon). The dataset is then preprocessed before being augmented to enhance it. For machine learning algorithms we have to preprocess the data in particular. Following that, we defined the deep learning model’s architecture and trained all of the models on the augmented dataset. Comparing both approaches, deep learning gives better results than machine learning algorithms as deep learning techniques can capture more complex relationships and patterns in the data. But in terms of lower computational cost and simplicity, ml also proved to be effective for estimating TC intensity. Our result shows that VGG16 among the deep learning techniques and among the machine learning algorithms, the ensemble method specially XGBoost is the most significant in predicting the intensity of Tropical cyclones.

6. Future Work

Combining deep learning and machine learning models with ensembling techniques offers a viable way to improve prediction performance. The combination of these two paradigms can maximize the advantages of each strategy while reducing the drawbacks of each one alone. By combining several projections from various models, ensemble integration produces predictions that are more reliable and accurate. Deep learning models are complicated and expressive; machine learning models, on the other hand, are simple and easy to read. The mixed ensemble preserves the interpretability and generalization powers of conventional machine learning techniques while also successfully capturing complex patterns in the data. This combination could lead to better results in many different applications, which would be a significant breakthrough for predictive modeling.

References

[1] Daloz, A., Peters-Lidard, C., Zhang, Y., Black, P., Velden, C., Toumi, R., 2006. “The hurricane weather research and forecast (hwrf) model”. Monthly Weather Review 134, 1663–1682.

[2] Chaudhuri, S., Dutta, D., Goswami, S., Middey, A., “Intensity forecast of tropical cyclones over north indian ocean using multilayer perceptron model: Skill and performance verification.” Natural Hazards 65, 97–113.Y.

[3] Ramos, A., Del Genio, A., Lindzen, R., Previdi, M., Frikken, B., 2009. “The geophysical fluid dynamics laboratory hurricane prediction system. Bull.” Am. Meteorol. Soc. 90, 825–836.

[4] Demirtas, M., Dee, D., Roberts, M., Uppala, S., Leutbecher, M., Isaksen, L., Berrisford, P., Bechtold, P., Simmonds, I., Kristjansson, J., ” The european centre for medium-range weather forecasts operational model for tropical cyclones.” Quarterly Journal of the Royal Meteorological Society 139, 1863–1876.

[5] Bankert, R.L., Hadjimichael, M., Kuciauskas, A.P., Richardson, K., Turk, J., Hawkins, J.D., 2002. Automating the estimation of various meteorological parameters using satellite data and machine learning techniques, in: IEEE International Geoscience and Remote Sensing Symposium, IEEE. pp. 708– 710.

[6] Chaudhuri, S., Dutta, D., Goswami, S., Middey, A., 2013. Intensity forecast of tropical cyclones over north indian ocean using multilayer perceptron model: Skill and performance verification. Natural Hazards 65, 97–113.Ali, Atif. ”Artificial Intelligence potential trends in military.” Foundation University Journal of Engineering and Applied Sciences (HEC Recognized Y Category, ISSN 2706-7351) 2, no. 1 (2021): 20-30.

[7] Zhang, C.J., Wang, X.J., Ma, L.M., Lu, X.Q., 2021. Tropical cyclone intensity classification and estimation using infrared satellite images with deep learning. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 14, 2070–2086. doi:10.1109/JSTARS.2021.3050767.

[8] Mercer, A., Grimes, A., 2015. Diagnosing tropical cyclone rapid intensification using kernel methods and reanalysis datasets. Procedia Computer Science 61, 422–427.

[9] He, Y., Chen, X., Ren, J., 2019. Intensity prediction of tropical cyclones based on deep learning with microwave radiances and numerical weather prediction data. Journal of Applied Meteorology and Climatology 58, 359–372.

[10] Liu, X., Li, H., Xiong, W., 2017. Deep learning based tropical cyclone intensity prediction, in: 2017 International Conference on Computational Intelligence and Security (CIS), IEEE. pp. 452–456.

[11] Maskey, M., Ramachandran, R., Ramasubramanian, M., Gurung, I., Freitag, B., Kaulfus, A., Bollinger, D., Cecil, D.J., Miller, J., 2020. Deepti: Deeplearning-based tropical cyclone intensity estimation system. IEEE journal of selected topics in applied Earth observations and remote sensing 13, 4271–4281.

[12] Tan, J., Yang, Q., Hu, J., Huang, Q., Chen, S., 2022. Tropical cyclone intensity estimation using himawari-8 satellite cloud products and deep learning. Remote Sensing 14,

[13] Hursat- hurricane satellite data. https://www.ncdc.noaa.gov/hursat/.(Accessed on 02/17/2023).

[14] Nhc data archive. https://www.nhc.noaa.gov/data/hurdat. (Accessedon 02/17/2023

[15] Wimmers, Anthony, Christopher Velden, and Joshua H. Cossuth. "Using deep learning to estimate tropical cyclone intensity from satellite passive microwave imagery." *Monthly Weather Review* 147.6 (2019): 2261-2282.

Authors' Profiles

Md. Ahsan Rahat completed his Bachelor in Computer Science and Engineering from the Military Institute of Science and Technology, Dhaka, Bangladesh. He is interested in Machine Learning, Data Science and Web Programming.

Nusrat Sharmin, assistant professor at the Military Institute of Science and Technology. She did her Ph.D. in machine learning in neuroimaging and her master's thesis was based on computer vision and image processing. Her research interests include digital image processing, machine learning, and combinatorial optimization problems. Her teaching interests include digital image processing, pattern recognition, and information and system design.

Fairooz Nawar Nawme completed his Bachelor in Computer Science and Engineering from Eastern University, Dhaka, Bangladesh. She is interested in Machine Learning, Blockchain Development and Data Science.

Sabbir Ahmed Tanzil Ahmed completed his Bachelor in Computer Science and Engineering from Eastern University, Dhaka, Bangladesh. He is interested in Machine Learning, Blockchain Development and Data Science.

How to cite this paper: Md. Ahsan Rahat, Nusrat Sharmin, Fairooz Nawar Nawme, Sabbir Rahman, "Performance Comparison and Investigation of Tropical Cyclone Intensity Estimation from Satellite Images Using Deep Learning and Machine Learning", International Journal of Image, Graphics and Signal Processing(IJIGSP), Vol.17, No.2, pp. 119-134, 2025. DOI:10.5815/ijigsp.2025.02.08