Hurricane Forecasting: A Novel Multimodal Machine Learning Framework

Léonard Boussioux*[1], Cynthia Zeng*[1],
Théo Guénais[2], Dimitris Bertsimas[1, 3]

[1]Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA, USA,
[2]School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA,
[3]Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA, USA

leobix@mit.edu

Abstract

This paper describes a machine learning (ML) framework for tropical cyclone intensity and track forecasting, combining multiple distinct ML techniques and utilizing diverse data sources. Our framework, which we refer to as Hurricast (HURR), is built upon the combination of distinct data processing techniques using gradient-boosted trees and novel encoder-decoder architectures, including CNN, GRU and Transformers components. We propose a deep-learning feature extractor methodology to mix spatial-temporal data with statistical data efficiently. Our multimodal framework unleashes the potential of making forecasts based on a wide range of data sources, including historical storm data, and visual data such as reanalysis atmospheric images. We evaluate our models with current operational forecasts in North Atlantic (NA) and Eastern Pacific (EP) basins on 2016-2019 for 24-hour lead time, and show our models consistently outperform statistical-dynamical models and compete with the best dynamical models. Furthermore, the inclusion of Hurricast into an operational forecast consensus model leads to a significant improvement of 5% - 15% over NHC's official forecast, thus highlighting the complementary properties with existing approaches.

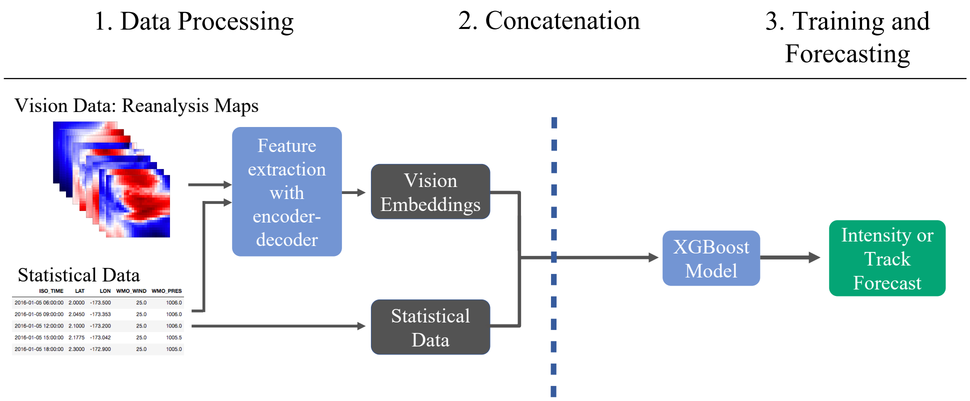
1 Introduction

A tropical cyclone (TC) is a low-pressure system originating from tropical or subtropical waters and developing by drawing energy from the sea. It is characterized by a warm core, organized deep convection and a closed surface wind circulation about a well-defined center. TCs draw energy from the warm ocean waters, and releases energy through heavy rainfall. There is growing evidence suggesting consistent hurricane intensity escalation due to climate change and warmer ocean waters [8]. For example, the recent storm Ida unexpectedly intensified more than the National Hurricane Center (NHC)'s forecast, strengthening rapidly overnight. Therefore, producing an accurate prediction for TC track and intensity with sufficient lead time is critical to undertake life-saving measures, and there is a need for new forecasting methodologies that can learn from the data directly.

Most forecasting models focus on producing track (trajectory) forecasting and intensity (such as the maximum sustained wind speed) forecasting. Current operational TC forecasts can be classified into dynamical models, statistical models and statistical-dynamical models [13]. Dynamical models utilize powerful supercomputers to simulate atmospheric fields' evolution using sophisticated physically-motivated dynamical equations. Statistical models approximate historical relationships between storm behavior and storm-specific features. Statistical-dynamical models use statistical techniques but further includes atmospheric variables provided by dynamical models. Lastly, ensemble models, also known as consensus forecasts, combine the forecasts made by individual models or multiple runs of a single model.

Recent developments in Deep Learning (DL) enable ML models to employ multiple data processing techniques to process and combine information from a wide range of sources, and create sophisticated architectures to model spatial-temporal relationships. Several studies have demonstrated the use of Recurrent Neural Networks to predict TC trajectory based on historical data [10, 1, 5]. In addition, Convolutional Neural Networks (CNNs) have also been applied to process reanalysis data and satellite data for track forecasting [9, 6, 11] and storm intensification forecasting [2, 15].

Tackling Climate Change with Machine Learning workshop at NeurIPS 2021


1. Data Processing
2. Concatenation
3. Training and Forecasting
Vision Data: Reanalysis Maps
Feature extraction with encoder-decoder
Vision Embeddings
Statistical Data
Statistical Data
XGBoost Model
Intensity or Track Forecast

ISO_TIME	LAT	LON	WMO_WIND	WMO_PRES
2016-01-05 06:00:00	2.0000	-173.500	25.0	1006.0
2016-01-05 09:00:00	2.0450	-173.353	25.0	1006.0
2016-01-05 12:00:00	2.1000	-173.200	25.0	1006.0
2016-01-05 15:00:00	2.1775	-173.042	25.0	1005.5
2016-01-05 18:00:00	2.3000	-172.900	25.0	1005.0

Figure 1: Representation of our multimodal machine learning framework using statistical data and reanalysis maps. In Step 1, data are processed individually to produce features using corresponding techniques. In particular, vision embeddings are extracted using either encoder-decoder architectures or tensor decomposition. In Step 2, perform feature selection and concatenation. In Step 3, XGBoost models are trained to produce 24h intensity and track forecasts.

This paper introduces a machine learning framework, called Hurricast (HURR), for both intensity and track forecasting by combining multiple machine learning approaches and several data sources. Our contributions are four-fold:

1. We present a new multimodal machine learning methodology for TC intensity and track predictions by combining distinct forecasting methodologies to utilize multiple individual data sources.
2. We demonstrate that our machine learning approach with advanced processing techniques, such as feature extraction from computer vision and deep learning, can outperform operational statistical-dynamical models and compete with dynamical models.
3. To the best of our knowledge, this is the first time a deep learning architecture is explicitly compared to operational forecasts for TC track and intensity forecast concurrently. This is also the first time the success of a Transformer architecture is demonstrated in TC prediction.
4. Based on our testing, ensemble models that include our machine learning model as an input can outperform all current operational models. This demonstrates the potential value of developing machine learning approaches as a new branch methodology for hurricane forecasting.

2 Data

Our study employs three kinds of data: historical storm data, reanalysis maps, and operational forecast data from the NHC. We include TCs dated from 1980, and reach a speed of 34 kn at some time and more than 60 h of data are available after they reached the speed of 34 kn for the first time. First, historical storm data are obtained from the National Oceanic and Atmospheric Administration, through the post-season storm analysis dataset IBTrACS [7]. Second, reanalysis data are obtained from the ERA-5 data set, which contains hourly high spatial resolution reanalysis maps from 1979 to present [4]. We extract nine reanalysis maps for each TC time step, corresponding to three different features, geopotential z, u and v components of the winds, at three atmospheric altitudes. Finally, operational forecast data is obtained from the ATCF data set, maintained by the NHC [12, 14].

3 Methodology

Overall, we adopt a three-step approach to combine distinct forecasting approaches and multiple data sources. First, we process each data source with a relevant methodology. Second, we concatenate all the processed features in a one-dimensional format. Third, we make our predictions using an XGBoost model [3] trained on the selected data. At a given time step, we perform two 24-hour lead time forecasts: an intensity prediction task, i.e., predicting the intensity of the storm at a 24-hour lead time; a displacement regression task, i.e., predicting the latitude and longitude storm displacement in degrees between given time and forward 24-hour time. Figure 1 illustrates the three-step pipeline. In the following section, we elaborate on the specific feature processing techniques for each type of data.

3.1 Vision Data Processing

We experimented with two feature extraction techniques on the vision data: i) deep learning-based encoder-decoder neural network approach; ii) unsupervised tensor decomposition approach. For the deep learning based approach, we report two specific encoder-decoder architectures: a CNN-encoder GRU-decoder and a CNN-encoder Transformer-decoder.

3.1.1 Encoder - Decoder Architectures

We perform two successive tasks based on the encoder-decoder architectures: (i) directly predict TC intensity and track, (ii) once the network is trained, we fix its weights and extract low-dimensional embeddings as features to be input into the final XGBoost model.

The CNN component acts as an encoder, and either a GRU or a Transformer component acts as a decoder. The CNN component aims to process (embed) the reanalysis maps. The GRU aims to model the temporal aspect through a recurrence mechanism, while the Transformer utilizes attention mechanisms and positional encoding to model long-range dependencies. Figure 2 and 3 illustrate the encoder-decoder architectures. Detailed explanations of the encoder-decoder architecture can be found in Appendix A.

In addition to directly forecasting TC intensity and track, the architecture provides predictive and low-size reanalysis maps' embeddings. After the training process described previously is completed, we freeze the encoder-decoder's weights and use the embeddings produced by the penultimate fully connected layer for each given sequence of reanalysis maps and statistical data.

3.1.2 Tensor decomposition

The motivation of using tensor decomposition is to represent high-dimensional data using low dimension features. Throughout this work, we use the Tucker decomposition definition, which is also known as the higher order singular value decomposition (SVD). In contrast to the aforementioned neural network-based feature processing techniques, tensor decomposition is an unsupervised extraction technique. More details can be found in Appendix B.

We treat each sequence of reanalysis maps as a four-dimensional tensor, of size $8 \times 9 \times 25 \times 25$ (corresponding to 8 past time steps of 9 reanalysis maps of size 25 pixels by 25 pixels), and used the core tensor obtained from the Tucker decomposition as extracted features. Finally, we flatten this core tensor that can now be concatenated with the corresponding statistical features to train the final XGBoost model.

4 Results

We have experimented with various architectures and data combinations to make intensity and track forecasting. A comprehensive list, as well as descriptions of all Hurricast models, can be found in Appendix C. We now summarize the key findings.

A multimodal approach leads to more accurate forecasts than using single data sources. Table 1 exhibits all variations of Hurricast for intensity forecasting. Multimodal models using vision and statistical data, i.e., HURR(stat/viz, ...) models, achieve higher accuracy and lower standard deviation than the models using only one data source, i.e., HURR(stat, ...) or HURR(viz, ...) models. Similar results for track forecasting can be found in Table 5 in Appendix D. In particular, we show that extracting embeddings from encoder-decoder architectures is particularly useful and outperforms the standalone predictions made directly by the deep learning model. This suggests the additional value from using tree-based models that can efficiently combine diverse data sources. Combining different machine learning techniques is a promising framework for TC forecasting.

Standalone machine learning models can produce competitive results compared to operational models. Table 1 shows that for intensity forecasting, standalone Hurricast models outperform the most advanced statistical-dynamical model (Decay-SHIPS) and dynamical model (GFSO) in both North Atlantic (NA) and Eastern Pacific (EP) basins. Furthermore, our models produce competitive results with the most advanced dynamical model (HWRF). Similarly, for track forecasting (see Table 5 in Appendix D), Hurricast models outperform the statistical benchmark model (CLP5) but underperform dynamical models (HWRF, GFSO). In addition, encoder-decoder standalone architectures are competitive with the best operational statistical-dynamical models. These results highlight that machine learning approaches can emerge as a new methodology to currently existing forecasting methodologies in the field.

Our machine learning framework brings additional insights to consensus models. Ensemble models often produce better performance by averaging out errors from individual models; for instance, the NHC employs

Table 1: Mean Average Error (MAE) and standard deviation of the error (Error sd) of standalone Hurricast models and operational forecasts on the same test set between 2016 and 2019, for intensity forecasting task.

Model Type	**Model Name**	Eastern Pacific Basin		North Atlantic Basin	
		Comparison on 36 TC		Comparison on 45 TC	
		MAE (km)	**Error sd (km)**	**MAE (km)**	**Error sd (km)**
Hurricast (HURR) Methods	HURR-(viz, cnn/gru)	**10.7**	10.1	**11.4**	9.6
	HURR-(viz, cnn/transfo)	**10.5**	**10.0**	**11.4**	**9.5**
	HURR-(stat, xgb)	10.5	10.4	10.8	9.3
	HURR-(stat/viz, xgb/td)	10.6	10.4	10.5	9.1
	HURR-(stat/viz, xgb/cnn/gru)	**10.3**	10.1	10.8	9.3
	HURR-(stat/viz, xgb/cnn/transfo)	**10.3**	**9.8**	**10.4**	**8.8**
Standalone Operational Forecasts	Decay-SHIPS	11.7	**10.4**	10.2	9.3
	HWRF	**10.6**	11.0	**9.7**	**9.0**
	GFSO	15.7	14.7	14.2	14.1

ensemble models to construct official forecasts. This work includes testings for two types of consensus models: first, ensemble of individual Hurricast variations; second, ensemble of our best Hurricast variation with other standalone operational models.

Table 2 exhibits the results on intensity forecasting; similar results for track forecasting can be found in the appendix in Table 6. The weighted consensus of all individual Hurricast variations consistently improves upon the best performing Hurricast variation, showcasing the possibility of building practical ensembles from machine learning models.

In addition, to validate whether a machine learning model can bring additional insights alongside current operational forecasts, we compare the accuracy of an average consensus model with and without our best model's inclusion. The version of consensus with the inclusion of Hurricast, i.e., HURR/OP-average consensus, is the best performing model, surpassing the NHC official forecast achieving higher accuracy with lower standard deviation for both track and intensity forecasting. These results highlight the complementary benefits of including a machine learning model.

Table 2: Mean Average Error (MAE) and standard deviation of the error (Error sd) of consensus models compared with NHC's official model OFCL on the same test set between 2016 and 2019 for intensity forecasting task.

Model Type	**Model Name**	Eastern Pacific Basin		North Atlantic Basin	
		Comparison on 36 TC		Comparison on 45 TC	
		MAE (kn)	**Error sd (kn)**	**MAE (kn)**	**Error sd (kn)**
Hurricast (HURR) Methods	HURR-(stat/viz, xgb/cnn/transfo)	10.3	**9.8**	10.4	**8.8**
	HURR-consensus	**10.2**	9.9	**10.2**	8.9
Operational Forecasts	FSSE	**9.7**	**9.5**	**8.5**	**7.8**
	OFCL	10.0	10.1	**8.5**	8.1
Consensus Models	Average consensus op. forecast	9.6	9.7	8.5	7.9
	HURR/OP-average consensus	**9.2**	**9.0**	**8.3**	**7.6**

5 Conclusion

This study demonstrates a novel multimodal machine learning framework for tropical cyclone intensity and track forecasting utilizing three distinct data sources: historical storm data, reanalysis geographical maps, and operational forecasts. We present a three-step pipeline to combine multiple machine learning approaches. We demonstrate that a successful combination of computer vision techniques and gradient-boosted trees can achieve strong predictions for both track and intensity forecasts, producing competitive results compared to current operational forecast models, especially in the intensity task. Moreover, once trained, our models run in seconds, demonstrating the potential for forecasting using real-time data.

In conclusion, our work demonstrates that machine learning can be a valuable approach to address bottlenecks in the field of tropical cyclone forecasting. We hope this work opens the door for further use of machine learning in meteorological forecasting.

References

[1] S. Alemany, J. Beltran, A. Perez, and S. Ganzfried. Predicting hurricane trajectories using a recurrent neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 468–475, 2019.

[2] R. Chen, X. Wang, W. Zhang, X. Zhu, A. Li, and C. Yang. A hybrid cnn-lstm model for typhoon formation forecasting. *GeoInformatica*, 2019.

[3] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016.

[4] ERA5. Era5 reanalysis, 2017. Accessed 2020-09-20.

[5] S. Gao, P. Zhao, B. Pan, Y. Li, M. Zhou, J. Xu, S. Zhong, and Z. Shi. A nowcasting model for the prediction of typhoon tracks based on a long short term memory neural network. *Acta Oceanologica Sinica*, 37:8–12, 2018.

[6] S. Giffard-Roisin, M. Yang, G. Charpiat, C. Kumler Bonfanti, B. Kégl, and C. Monteleoni. Tropical cyclone track forecasting using fused deep learning from aligned reanalysis data. *Frontiers in Big Data*, 3:1, 2020.

[7] K. R. Knapp, M. C. Kruk, D. H. Levinson, H. J. Diamond, and C. J. Neumann. The international best track archive for climate stewardship (ibtracs): Unifying tropical cyclone best track data, 2010.

[8] T. Knutson, S. J. Camargo, J. C. L. Chan, K. Emanuel, C.-H. Ho, J. Kossin, M. Mohapatra, M. Satoh, M. Sugi, K. Walsh, and L. Wu. Tropical Cyclones and Climate Change Assessment: Part I: Detection and Attribution. *Bulletin of the American Meteorological Society*, 100(10):1987–2007, 10 2019.

[9] J. Lian, P. Dong, Y. Zhang, J. Pan, and K. Liu. A novel data-driven tropical cyclone track prediction model based on cnn and gru with multi-dimensional feature selection. *IEEE Access*, 2020.

[10] M. Moradi Kordmahalleh, M. Gorji Sefidmazgi, and A. Homaifar. A sparse recurrent neural network for trajectory prediction of atlantic hurricanes. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, pages 957–964, 2016.

[11] M. Mudigonda, S. Kim, A. Mahesh, S. Kahou, K. Kashinath, D. Williams, V. Michalski, T. O'Brien, and M. Prabhat. Segmenting and tracking extreme climate events using neural networks. 2017.

[12] National Hurricane Center. Automated tropical cyclone forecasting system (atcf), 2021. Accessed: 2021-04-06.

[13] J. R. Rhome. Technical summary of the national hurricane center track and intensity models. *Updated September*, 12:2007, 2007.

[14] C. Sampson and A. J. Schrader. The automated tropical cyclone forecasting system (version 3.2). *Bulletin of the American Meteorological Society*, 81:1231–1240, 2000.

[15] H. Su, L. Wu, J. H. Jiang, R. Pai, A. Liu, A. J. Zhai, P. Tavallali, and M. DeMaria. Applying satellite observations of tropical cyclone internal structures to rapid intensification forecast with machine learning. *Geophysical Research Letters*, 47(17), 2020.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. 2017.

A Encoder-Decoder

Figures 2 and 3 provide a schematic representation of our neural network architectures.

The CNN-encoder At each time step, the corresponding nine reanalysis maps are fed into the CNN-encoder, which produces one-dimensional embeddings. The CNN-encoder consists of three convolutional layers, with ReLU activation and MaxPool layers in between, then followed by two fully connected layers.

Next, we concatenate the reanalysis maps embeddings with processed statistical data corresponding to the same time step. Note that at this point data is still sequentially structured as 8 time steps to be passed on to the GRU-decoder or the Transformer-decoder.

The GRU-decoder Our GRU-decoder consists of two unidirectional layers. The data sequence embedded by the encoder is fed sequentially in chronological order into the GRU-decoder. For each time step, the GRU-decoder outputs a hidden state representing a "memory" of the previous time steps. Finally, a track or intensity prediction is made based upon these hidden states concatenated all together and given as input to fully-connected layers (see Figure 2).

The Transformer-decoder Conversely to the GRU-decoder, the sequence is fed as a whole into the Transformer-decoder. Since attention mechanisms allow each hidden representation to attend holistically to the other hidden representations, the time-sequential aspect is lost. Therefore, we add a *positional encoding* token at each timestep-input, following standard practices [16]. This token represents the relative position of a time-step within the sequence and re-introduces some information about the inherent sequential aspect of the data and experimentally improves performance.

Then, we use two Transformer layers that transform the 8 time steps (of size 142) into an 8-timestep sequence with similar dimensions. To obtain a unique representation of the sequence, we average the output sequence feature-wise into a one-dimensional vector, following standard practices. Finally, a track or intensity prediction is made based upon this averaged vector input into one fully-connected layer (see Figure 3).

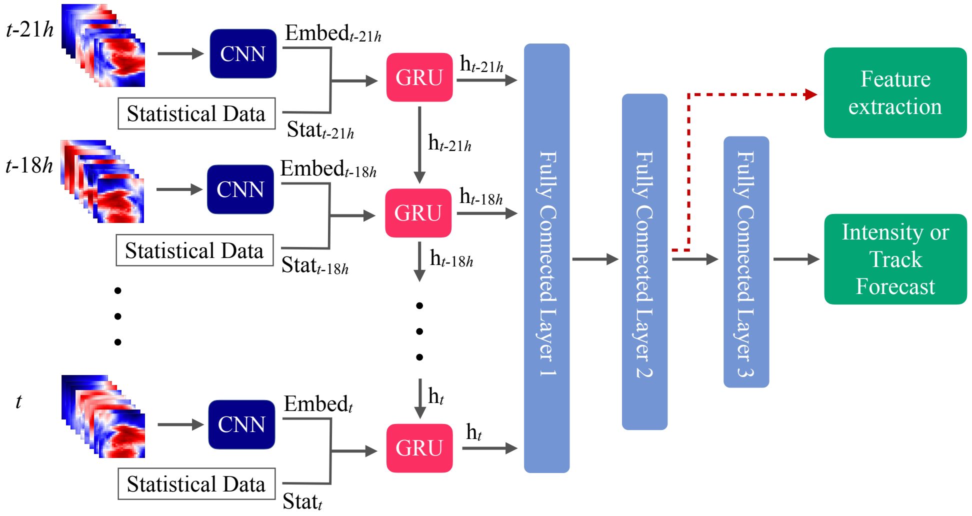


Figure 2: Representation of our CNN-encoder GRU-decoder network for an 8-time step TC sequence. At each time step, we utilize the CNN to produce one-dimensional embeddings of the reanalysis maps. Then, we concatenate these embeddings with the corresponding statistical features to create a sequence of inputs fed sequentially to the GRU. At each time step, the GRU outputs a hidden state passed to the next time step. Finally, we concatenate all the successive hidden states and pass them through three fully connected layers to predict intensity or track with a 24-hour lead time. Spatial-temporal embeddings can be extracted as the output of the second fully connected layer.

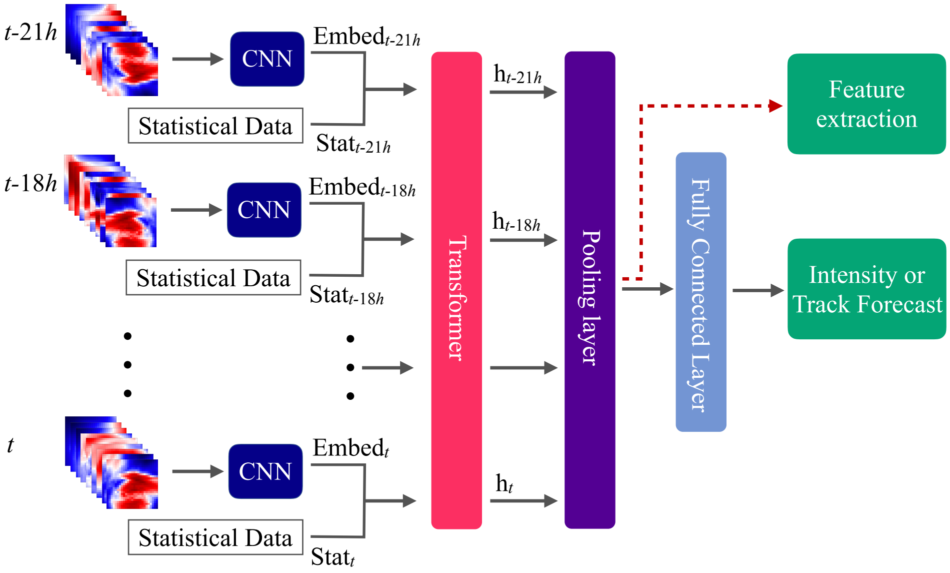


Figure 3: Representation of our CNN-encoder Transformer-decoder network for an 8-time step TC sequence. At each time step, we utilize the CNN to produce one-dimensional embeddings of the reanalysis maps. Then, we concatenate these embeddings with the corresponding statistical features to create a sequence of inputs fed as a whole to the Transformer. The Transformer outputs a new 8-timestep sequence that we average (pool) feature-wise and then feed into one fully connected layer to predict intensity or track with a 24-hour lead time. Spatial-temporal embeddings can be extracted as the output of the pooling layer.

B Tucker decomposition for tensors

The multilinear singular value decomposition (SVD) expresses a tensor $\mathcal{A}$ as a small core tensor $\mathcal{S}$ multiplied by a set of unitary matrices. The size of the core tensor, denoted by $[k_1, \dots k_N]$, defines the rank of the tensor. Formally, the multilinear decomposition can be expressed as:

$$\begin{aligned} \mathcal{A} &= \mathcal{S} \times_1 U^{(1)} \times_2 \cdots \times_N U^{(N)} \\ \text{where } \mathcal{A} &\in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \\ \mathcal{S} &\in \mathbb{R}^{k_1 \times k_2 \times \cdots \times k_N} \\ U^{(i)} &\in \mathbb{R}^{I_i \times k_i} \end{aligned}$$

where each $U^{(i)}$ is a unitary matrix, i.e., its conjugate transpose is its inverse $U^{(i)*}U^{(i)} = U^{(i)}U^{(i)*} = I$, and the mode-n product, denoted by $\mathcal{A} \times_n U$, denotes the multiplication operation of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ by a matrix $U \in \mathbb{R}^{I_n \times J_n}$.

Analogous to truncated SVD, we can reduce the dimensionality of tensor $\mathcal{A}$ by artificially truncating the core tensor $\mathcal{S}$ and corresponding $U^{(i)}$. For instance, given a 4-dimensional tensor of TC maps, we can decide reduce the tensor to any desired rank by keeping only the desired size of core tensor $\mathcal{S}$. For instance, to reduce TC tensor data into rank $3 \times 5 \times 3 \times 3$, we first perform multilinear SVD, such that S reflects descending order of the singular values, and then truncate $\mathcal{S}$ by keeping only the first $3 \times 5 \times 3 \times 3$ entries, denoted by $\mathcal{S}'$, and the first 3 columns of each of $U^{(i)}$, denoted by $U'^{(i)}$.

Finally, we flatten the truncated core tensor $\mathcal{S}'$ into a vector, which is treated as the extracted vision features in order to train tree-based model.

C Models

We provide a summary of all variations of our Hurricast models (Table 3) and of all operational forecast models included in our benchmark (Table 4).

Table 3: Summary of the various versions of the Hurricast framework for which we report results. Models differ in architecture and in data used. The data used can be a subset of statistical, vision (reanalysis maps embedded or not), and operational forecasts. For this reason, models are named based on two characteristics: the data used and the methods used.

N°	Name	Data Used	ML Methods
1	HURR-(viz, cnn/gru)	Vision, Statistical	CNN, GRU
2	HURR-(viz, cnn/transfo)	Vision, Statistical	CNN, Transformers
3	HURR-(stat, xgb)	Statistical	XGBoost
4	HURR-(stat/viz, xgb/td)	Statistical, Vision embeddings	XGBoost, Tensor decomposition
5	HURR-(stat/viz, xgb/cnn/gru)	Statistical, Vision embeddings	XGBoost, CNN, GRU
6	HURR-(stat/viz, xgb/cnn/transfo)	Statistical, Vision embeddings	XGBoost, CNN, Transformers
7	HURR-consensus	Models 1-6 forecasts	ElasticNet
8	HURR/OP-average	Operational forecasts, HURR-(stat/viz, xgb/cnn/transfo)	Simple average

Model 1 and 2 utilize vision data (reanalysis maps) coupled with statistical data, employing neural networks only with the encoder-decoder architecture detailed in Section 3.1.1. Both models employ a CNN-encoder architecture, but Model 1 employs GRUs as decoder, while Model 2 employs Transformers.

Models 3-6 are variations of the three-step framework described in Figure 1, with the variation of input data source or processing technique. Model 3, HURR-(stat,xgb), has the simplest form, utilizing only statistical data. Model 4-6 utilize statistical data and vision data. They differ on the extraction technique used on the reanalysis maps. Model 4, HURR-(stat/viz,xgb/td), uses vision features extracted with tensor decomposition technique (see 3.1.2), whereas Model 5 and 6 utilize vision features extracted with the encoder-decoder (see 3.1.1), with GRU and Transformer decoders respectively.

Finally, Model 7 — HURR-consensus — and 8 — HURR/OP-average — are consensus models. Model 7 is a weighted consensus model of Models 1 to 6 forecasts. The weights given to each model are optimized using ElasticNet. As we trained multiple models using different data processing techniques and data sources, we built a diversified pool of predictors. This is typically the scenario where ensemble models can improve performance.

Model 8 is a simple average consensus of a few operational forecasts models used by the NHC and our Model 6, HURR-(stat/viz,xgb/cnn/transfo). We use Model 8 to explore whether the Hurricast framework can bring additional benefits to current operational forecasts by comparing its inclusion as a member model.

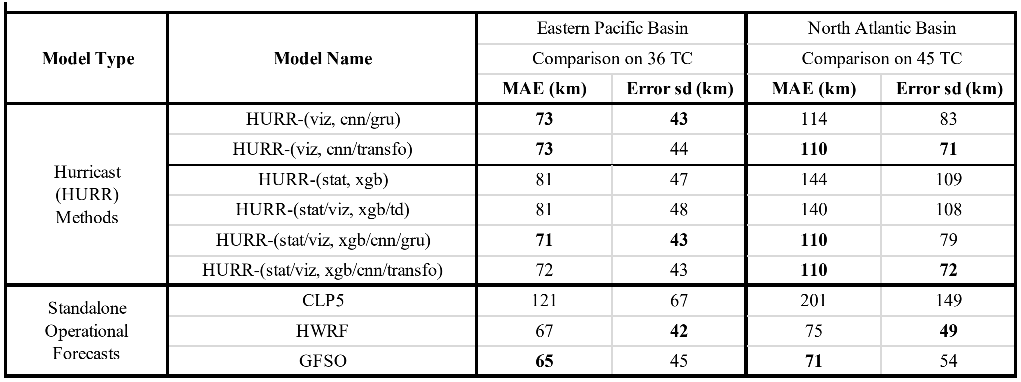
We provide further details on the performance of these different models compared to operational forecasts in the following Section D for the track task. The results on the intensity task are in the main paper.

Table 4: Summary of all operational forecast models included in our benchmark.

Model ID	Model name or type	Model type	Forecast
CLP5	CLIPER5 Climatology and Persistence	Statistical (baseline)	Track
Decay-SHIPS	Decay Statistical Hurricane Intensity Prediction Scheme	Statistical-dynamical	Intensity
GFSO	Global Forecast System model	Multi-layer global dynamical	Track, Intensity
HWRF	Hurricane Weather Research and Forecasting model	Multi-layer regional dynamical	Track, Intensity
AEMN	GFS Ensemble Mean Forecast	Consensus	Track
FSSE	Florida State Super Ensemble	Corrected consensus	Track, Intensity
OFCL	Official NHC Forecast	Consensus	Track, Intensity

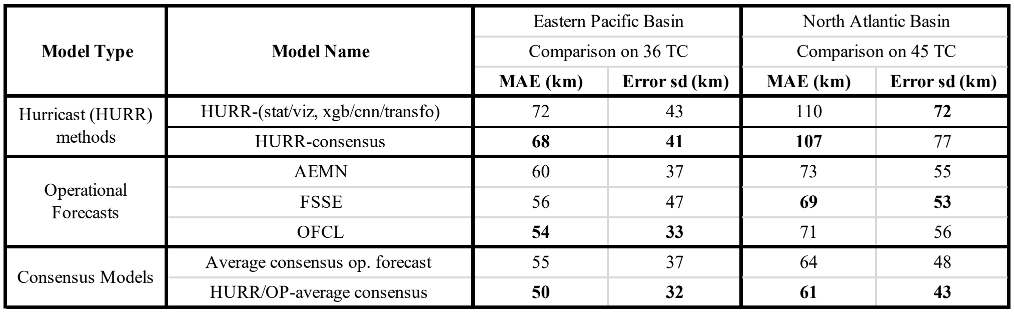
D Track Results

Table 5: Mean Average Error (MAE) and standard deviation of the error (Error sd) of standalone Hurricast models and operational forecasts on the same test set between 2016 and 2019, for track forecasting task.



Model Type	Model Name	Eastern Pacific Basin		North Atlantic Basin	
		Comparison on 36 TC		Comparison on 45 TC	
		MAE (km)	**Error sd (km)**	**MAE (km)**	**Error sd (km)**
Hurricast (HURR) Methods	HURR-(viz, cnn/gru)	**73**	**43**	114	83
	HURR-(viz, cnn/transfo)	**73**	44	**110**	**71**
	HURR-(stat, xgb)	81	47	144	109
	HURR-(stat/viz, xgb/td)	81	48	140	108
	HURR-(stat/viz, xgb/cnn/gru)	**71**	**43**	**110**	79
	HURR-(stat/viz, xgb/cnn/transfo)	72	43	**110**	**72**
Standalone Operational Forecasts	CLP5	121	67	201	149
	HWRF	67	**42**	75	**49**
	GFSO	**65**	45	**71**	54

Table 6: Mean Average Error (MAE) and standard deviation of the error (Error sd) of consensus models compared with NHC's official model OFCL on the same test set between 2016 and 2019 for track forecasting task.



Model Type	Model Name	Eastern Pacific Basin		North Atlantic Basin	
		Comparison on 36 TC		Comparison on 45 TC	
		MAE (km)	**Error sd (km)**	**MAE (km)**	**Error sd (km)**
Hurricast (HURR) methods	HURR-(stat/viz, xgb/cnn/transfo)	72	43	110	**72**
	HURR-consensus	**68**	**41**	**107**	77
Operational Forecasts	AEMN	60	37	73	55
	FSSE	56	47	**69**	**53**
	OFCL	**54**	**33**	71	56
Consensus Models	Average consensus op. forecast	55	37	64	48
	HURR/OP-average consensus	**50**	**32**	**61**	**43**