Graduate School - College of Informatics and Computing Sciences
Batangas State University - The National Engineering University
Alangilan Campus, Pablo Borbon II, Batangas City

Data mining approaches for classifying tropical cyclone rapid intensification in the Western Pacific from wind structure

Capstone Project Presentation | MSDS 503: Machine Learning & Neural Networks | May 26, 2024

Presenter:
Henric Rosas Jandoc
Graduate Student, MSDS

Professor:
Francis Jesmar P. Montalbo, DIT
CICS Faculty

AUTHOR BACKGROUND

Henric R. Jandoc
BS Meteorology (grad. 2023)
Professorial Lecturer
Department of Physics & Meteorology
Bicol University
Legazpi City, Albay

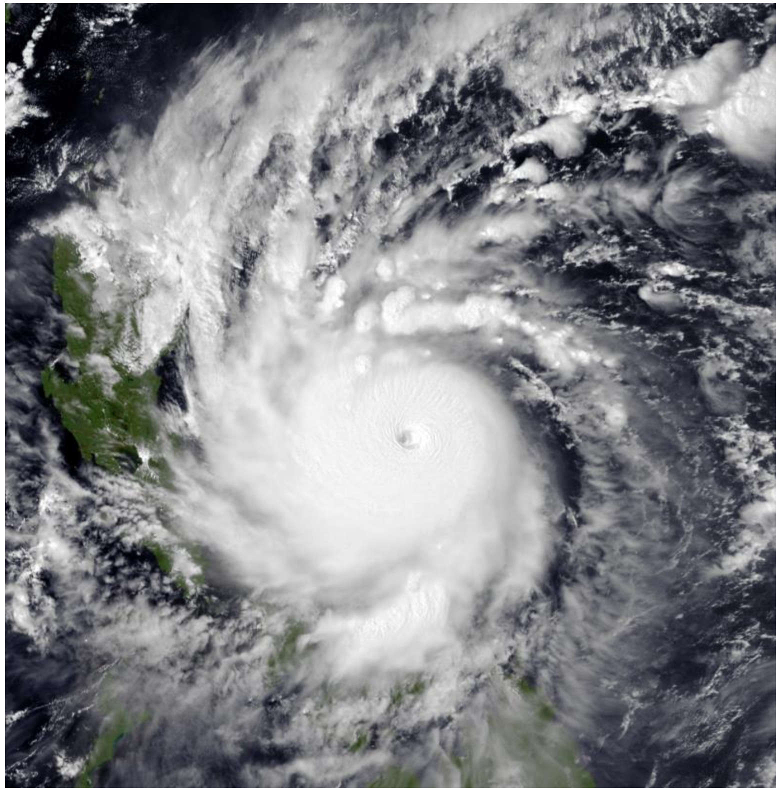
Francis Jesmar P. Montalbo, DIT
Doctor in Information Technology
Associate Professor
College of Informatics & Computing Sciences
Batangas State University
Batangas City, Batangas

TABLE OF CONTENTS

BACKGROUND What are Tropical Cyclones?

Tropical Cyclones (Storms)
definition by the World Meteorological Organization (WMO)

- a **non-frontal** synoptic scale low-pressure system over tropical or subtropical waters with **organized convection** and **definite cyclonic surface wind circulation**



Super Typhoon Goni (Rolly, 2020) making landfall over PH (as seen from visible satellite)

Tropical Cyclones (Storms); (In PH, *Bagyo*)
audience-friendly definition

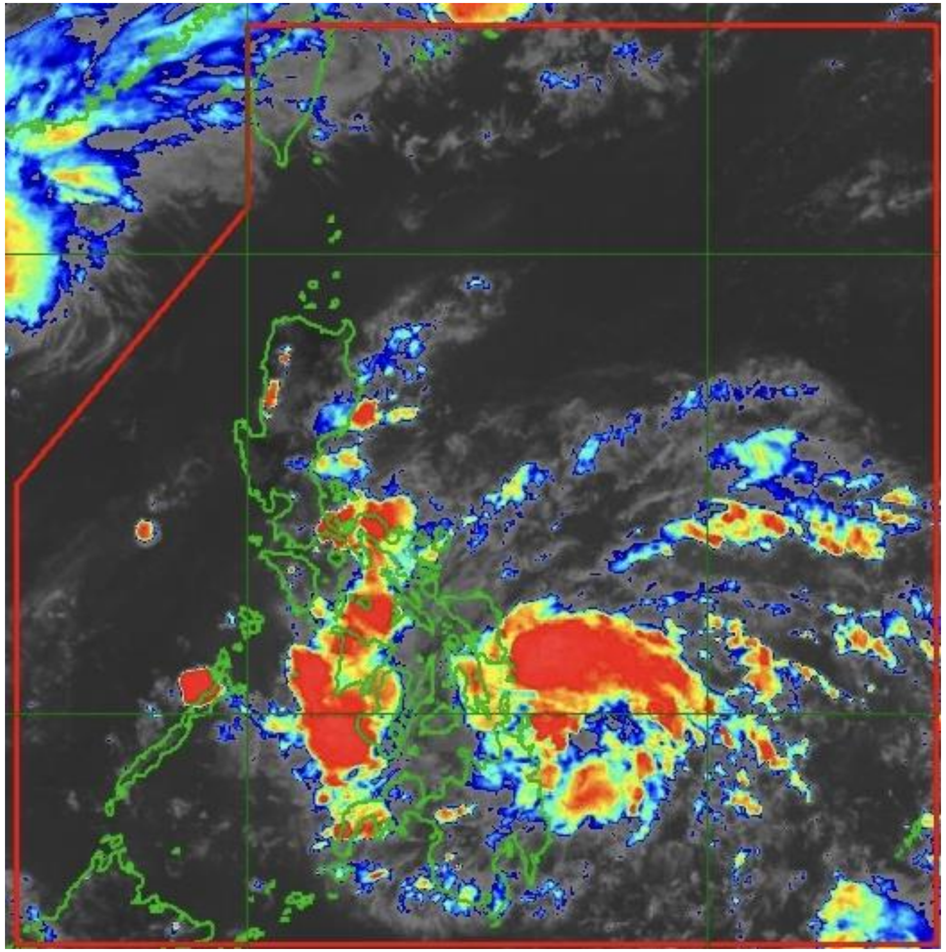
- a meteorological phenomenon characterized by a circular wind pattern with accompanying cloud systems and rainfall

BACKGROUND | What are Tropical Cyclones?

Why do meteorologists / atmospheric scientists study Tropical Cyclones?

It brings ***very violent winds***, ***heavy-torrential rain***, ***high waves*** and potentially very destructive ***storm surges*** and ***coastal flooding***.

By understanding the characteristics of tropical cyclones or storms, we can develop mitigation strategies to minimize the risks. In other words, to save lives and businesses.



TD Aghon as seen from the Himawari Satellite (05 /24 / 24)

BACKGROUND The Challenge in Rapid Intensification

Rapid Intensification (RI)
When a Tropical Cyclone's wind speed increases by equal to or greater than 30 kt (56 kph) in 1 day

The 95th percentile of all intensity changes of Tropical Cyclones globally (Kaplan et al., 2010)

In order to mitigate the disastrous effects of tropical cyclones that underwent rapid intensification, we need to identify the two main factors that influence it.

[1] Environmental Conditions

- ❑ Sea Surface Temperature
- ❑ Vertical Wind Shear
- ❑ Relative Humidity
- ❑ Outflow

[2] Internal Dynamics

- ❑ Deep Convective Clouds
- ❑ Precipitation
- ❑ Cloud Microphysical Properties
- ❑ Wind Structure

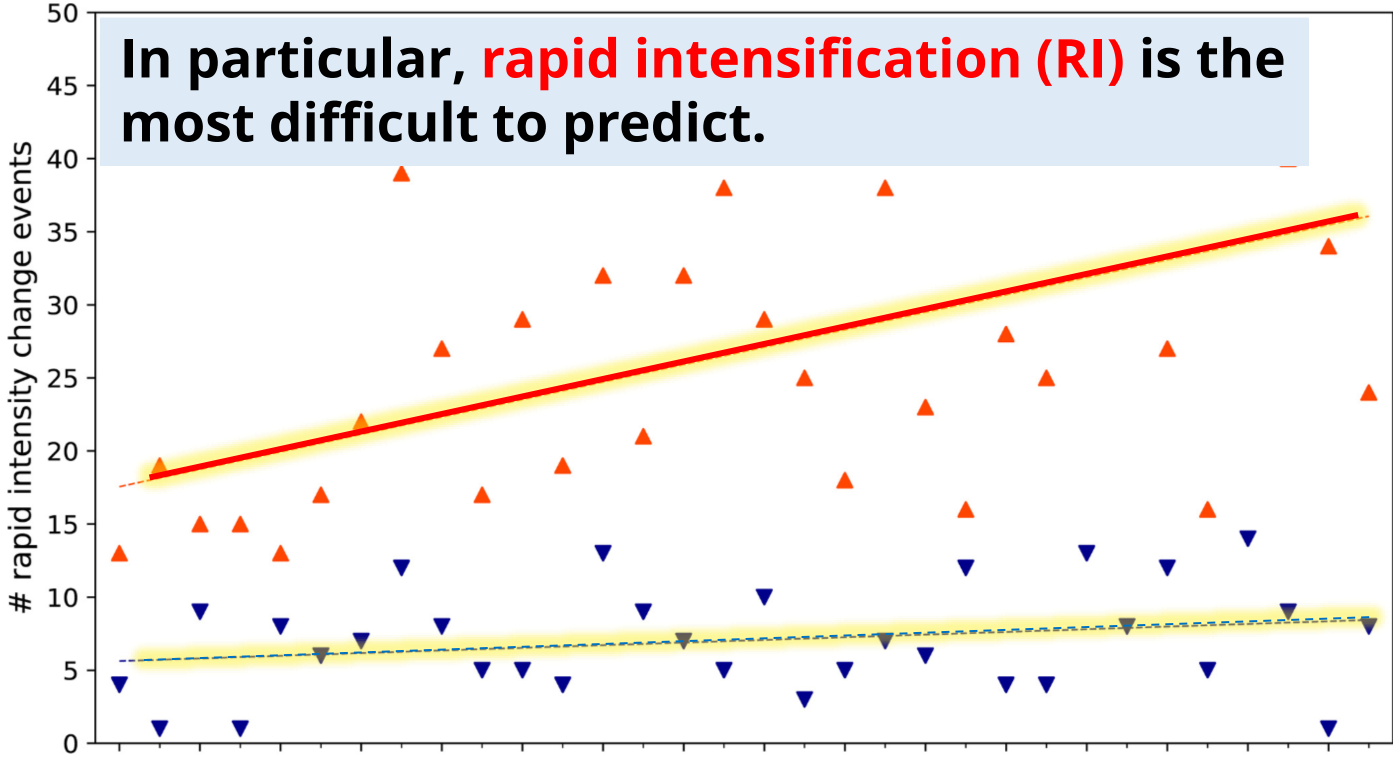
Hendricks et al. (2012), Ruan and Wu (2016), Guo and Tan (2022)

BACKGROUND

The problem in TC rapid intensification forecasts

Problem!!! Forecasting or predicting intensification has barely improved over the past few decades.

In particular, rapid intensification (RI) is the most difficult to predict.



More TCs are attaining RI over the years!

Bhatia et al., 2022

BACKGROUND

The problem in TC rapid intensification forecasts

- ❑ Most **Super Typhoons (>185 kph)** underwent RI
- ❑ RI can have different durations (**Super Typhoons usually have long RI durations**)

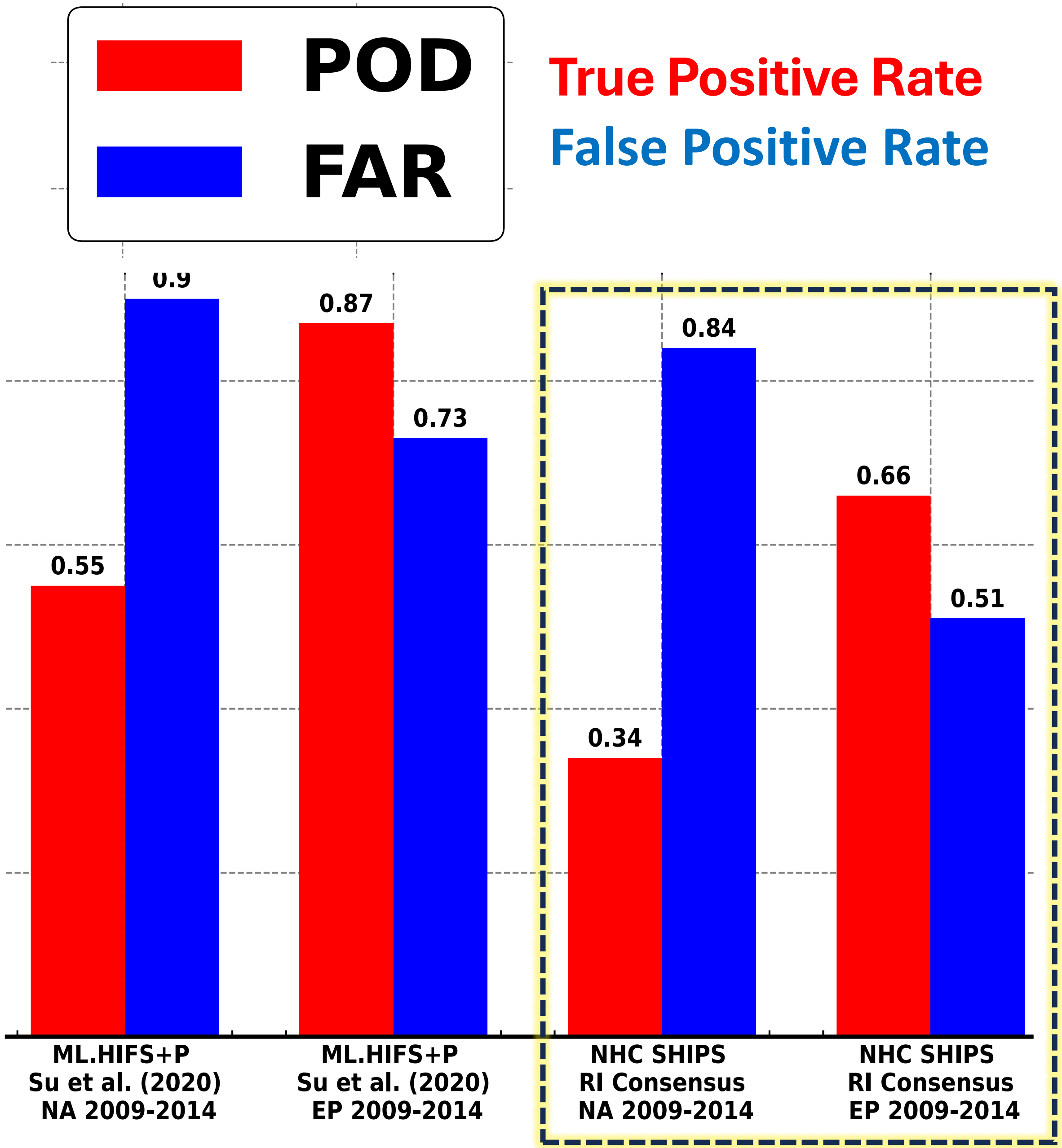
IBTrACS

Poor forecast of RI can lead to:

- ❑ False alarm
- ❑ Distrust of community to authorities
- ❑ Complacency / Panic of LGUs

BACKGROUND Rapid Intensification Prediction with ML

One solution to improve RI forecasts is to utilize numerical models and statistical methods (Wang et al., 2023).



True Positive Rate

False Positive Rate

National Hurricane Center Statistical Hurricane Intensity Prediction Scheme Rapid Intensification Consensys (NHC SHIPS RI Con)

SHIPS RI Consensus utilizes environmental predictors and some satellite-derived infrared features

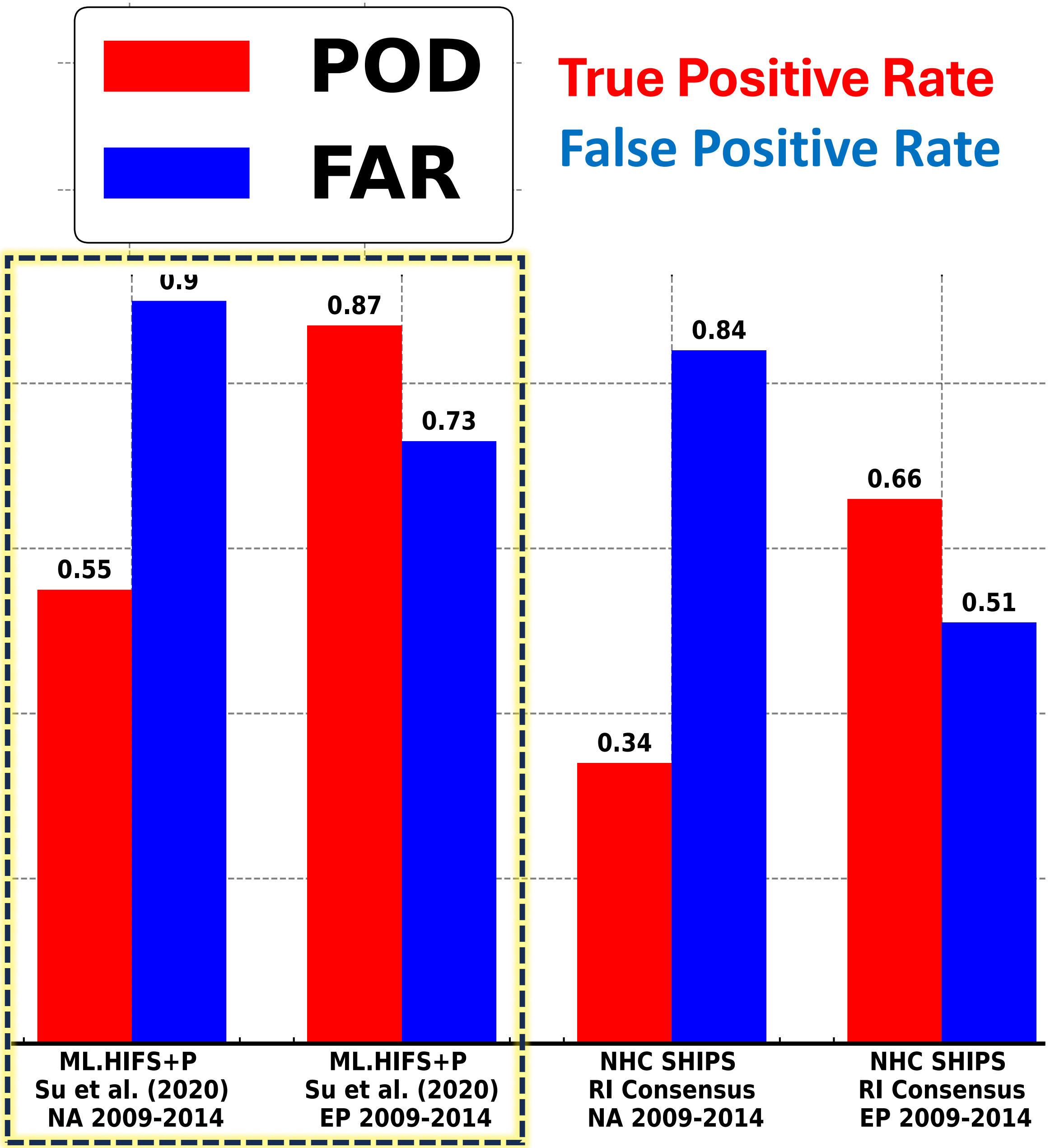
Uses an ensemble of Logistic Regression Bayesian models, and discriminant analysis

BACKGROUND Rapid Intensification Prediction with ML

Machine Learning Hurricane Intensity Forecasting Scheme plus Precipitation (ML.HIFS+P)

ML.HIFS+P **integrates satellite observed TC internal structures** such as inner-core cloud ice water path and content, outflow temperature, and surplus precipitation, along with environmental predictors from the SHIPS dataset

Uses a weighted ensemble of Logistic Regression, Random Forest, Extra Trees, etc.



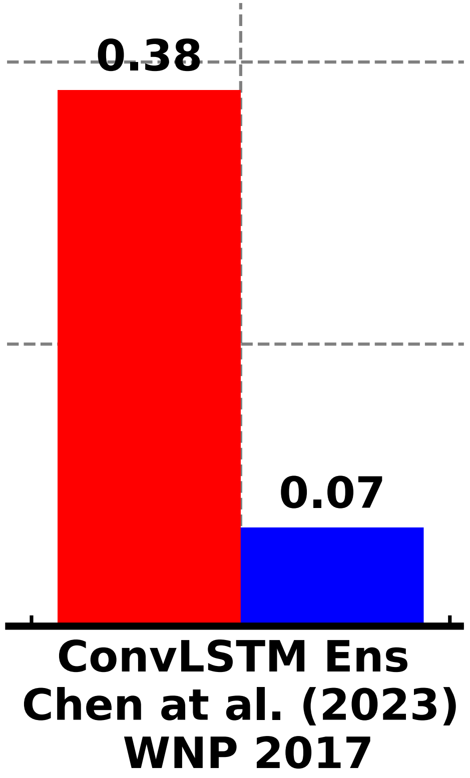
True Positive Rate

False Positive Rate

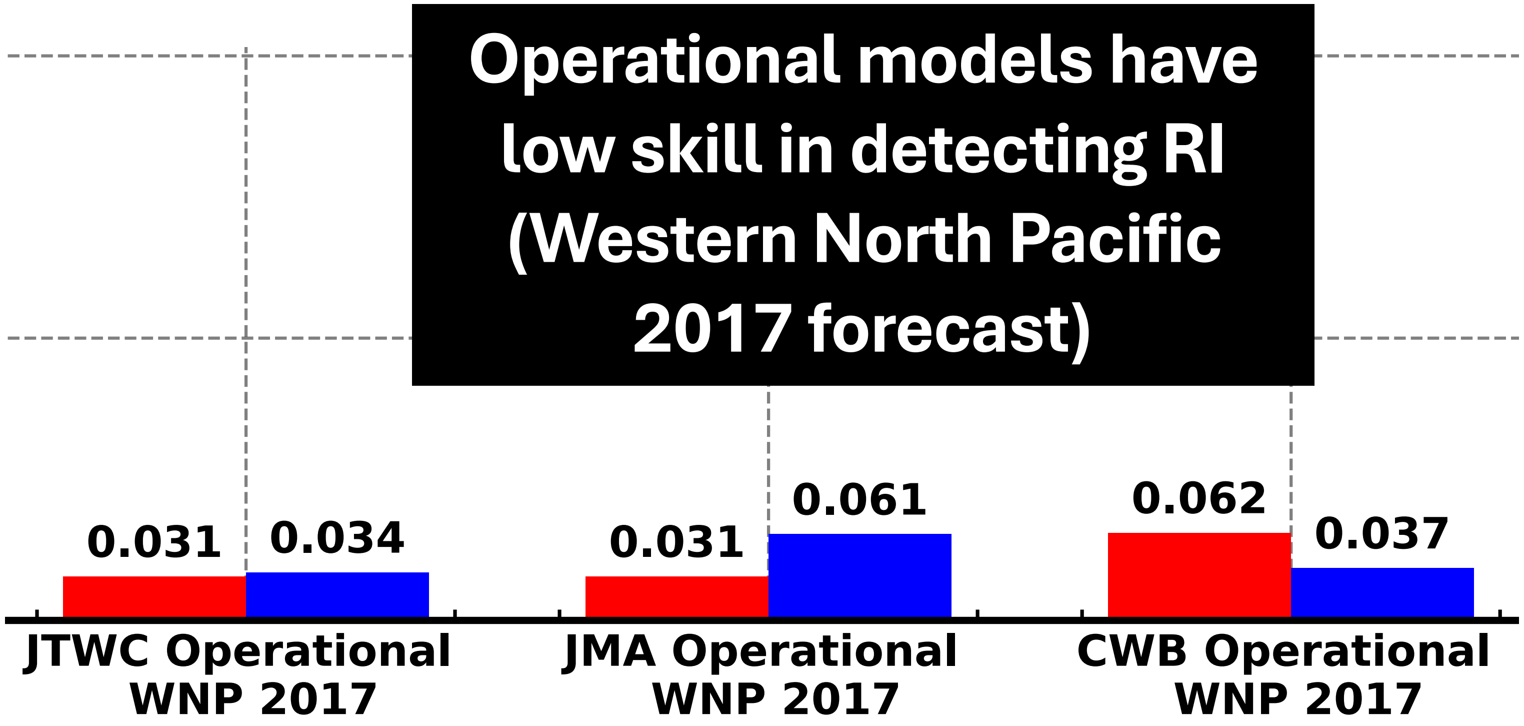
How did the ML.HIFS+P outperform the best operational model?

- ML techniques instead of statistical techniques
- Internal structures was integrated alongside environmental predictors leading to a more sophisticated representation of the RI process

BACKGROUND Rapid Intensification Prediction with ML



Operational models have low skill in detecting RI (Western North Pacific 2017 forecast)



Ensemble of Convolutional Long-Short Term Memory (ConvLSTM) utilized satellite images of infrared and water vapor, and scalar environmental conditions *(Chen et al., 2023)*.

Operational models utilize a combination of numerical models, ensemble forecasting techniques, and statistical methods to predict RI *(Wang et al., 2023)*.

BACKGROUND

Objectives

To demonstrate the **predictive capability of the newly defined wind structure or fullness parameters for TC RI forecasting in the Western Pacific basin** by incorporating it as the features of six machine learning classification models and then by evaluating its performance under different experiment setups.

- Synthetic Minority Oversampling Technique
- Adding time-lagged features
- Feature Selection (SelectKBest)

Objectives

Why Western Pacific basin?

- significant to the Philippines
- lack of ML-RI studies in the Western Pacific

Why Wind Structure?

- rapidly intensifying TCs can exhibit a unique wind structure (e.g., high fullness / size, moderate strength)
- wind structure features can be easily derived from best-track datasets (good for quick case studies)
- majority of ML-RI studies utilize environmental predictors and satellite data, which are hard to process and computationally expensive (file format in NetCDF and HDF5)

TABLE OF CONTENTS

Dataset (IBTrACS)

Understanding the Dataset – International Best Track Archive for Climate Stewardship (IBTrACS)

A global archive of tropical cyclone data that contains the most accurate and complete information about the tracks and intensity of these storms around the world.

It combines recent and historical tropical cyclone data from multiple agencies to create a consistent, publicly accessible best-track dataset that facilitates agency comparisons.

Reference:

https://www.ncei.noaa.gov/products/international-best-track-archive

METHODOLOGY Dataset (IBTrACS)

ORIGINAL DATASET:
rows: 243,483 instances of storms in the Western Pacific *(1884 – 2023)*
columns: 163 features / observed storm characteristics by weather stations around the world

CLEANED / TREATED DATASET: (using python pandas)
rows: 5784 instances of storms *(2001-2022)*
columns: 7 (6 as features; 1 target)

Domain knowledge-guided filtering: (backed by literature)

- recent storms (yr 2001+) – climatological characteristics
- intensifying cases (no weakening) – we are focused on intensification
- negative fullness values are omitted – to reduce noise and better data quality

Guo & Tan 2022

METHODOLOGY

Wind Structure Features

Features	Description
Vmax	1-min. Maximum Sustained Winds; the intensity or strength of a TC. *(units: kt)*
RMW	Radius of Maximum Wind; distance from TC center to its band of strongest winds. *(units: nmile)*
R34	Radius of the 34 kt Wnds; distance from the TC center to the gale-force winds. *(units: nmile)*
Derived Features	
TCF	Tropical Cyclone Fullness; the size of the outer annular wind ring (R34-RMW) relative to the outer-core size (R34), and the size part of fullness *(unitless)* Mathematically equivalent to: *1 – RMW/R34*
TCF0	Tropical Cyclone Critical Fullness; the intensity part of fullness. *(unitless)* Mathematically equivalent to: *1 – 33 kt / Vmax*
RF	The ratio of fullness; simply the ratio between TCF and TCF0. *(unitless)*

```
df['TCF'] = 1 - (df['USA_RMW']/df['USA_R34_NE'])
df['TCF0'] = 1 - (33/df['USA_WIND'])
df['Rf'] = df['TCF']/df['TCF0']
```

$$TCF = 1 - \frac{RMW}{R34}$$

$$TCF0 = 1 - \frac{V34}{Vmax}$$

$$\mathrm{RF} = \frac{TCF}{TCF0}$$

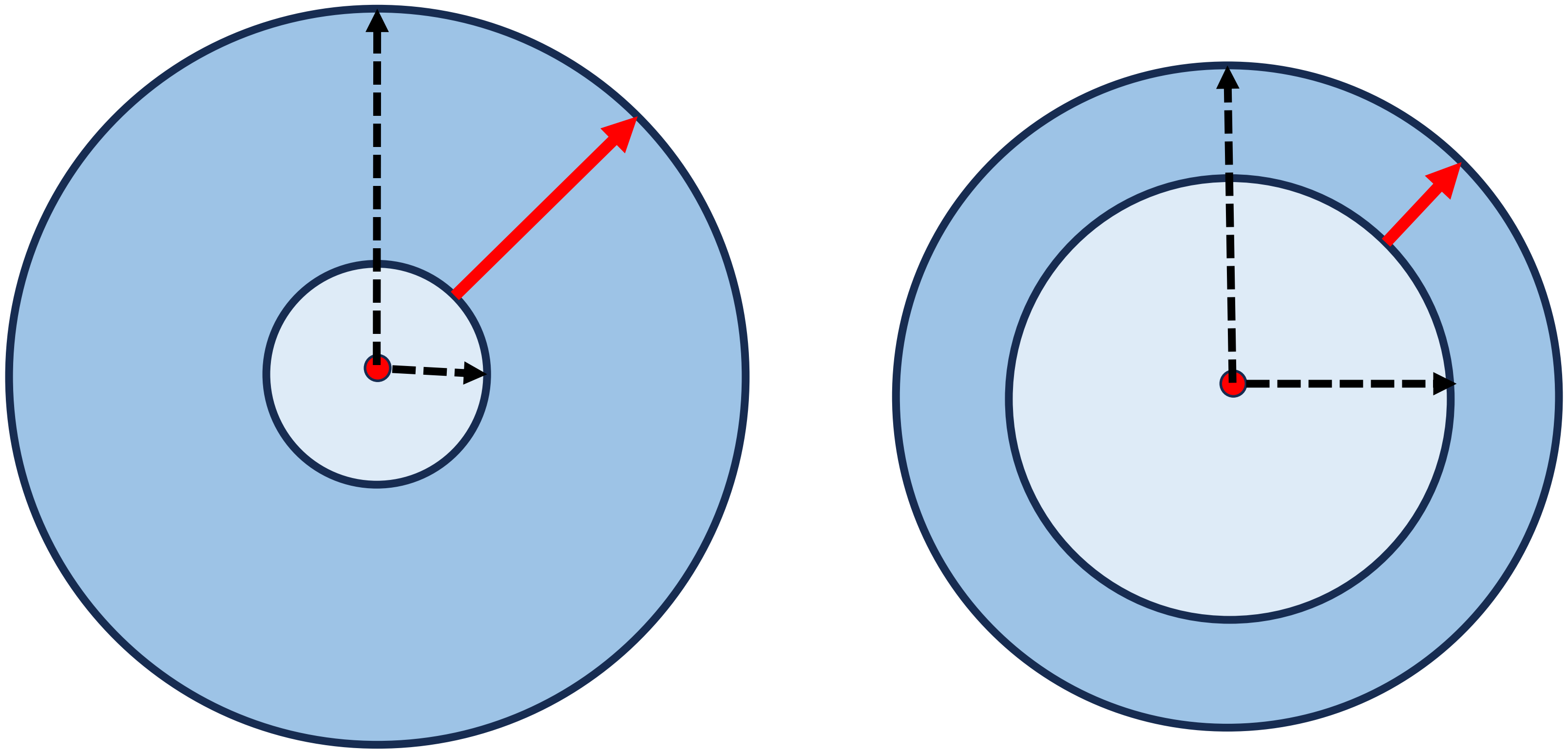
Guo & Tan 2022

METHODOLOGY

Wind Structure Features

Features (2)
Storm Size / TC Fullness
Storm Strength / Critical Fullness

Red Arrow = difference of the dotted black lines / difference of the two radii



"LARGE" Storm

LARGE RED ARROW

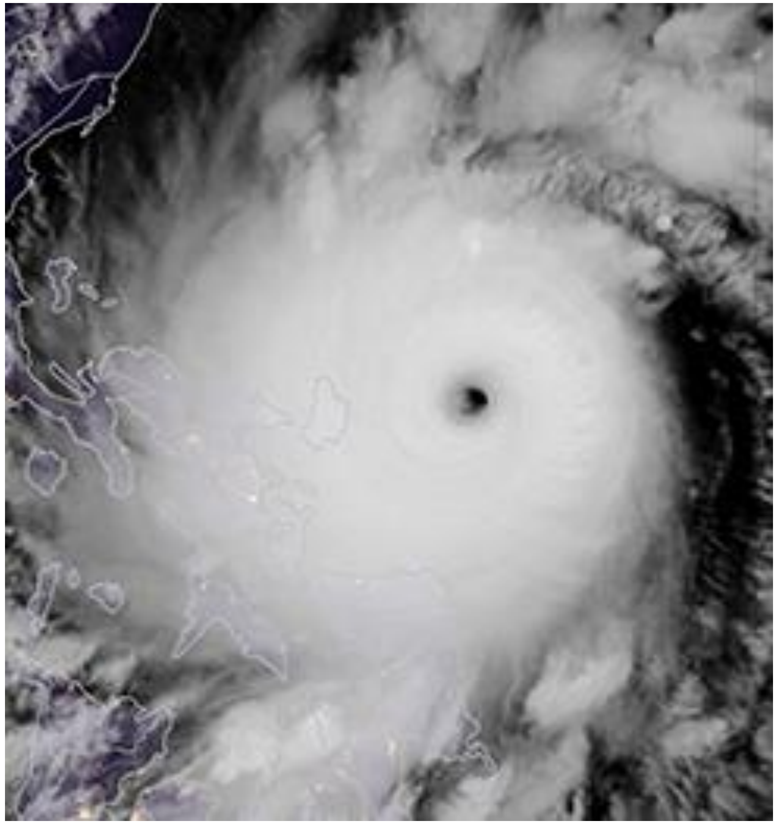
"Small" Storm

Small Red Arrow

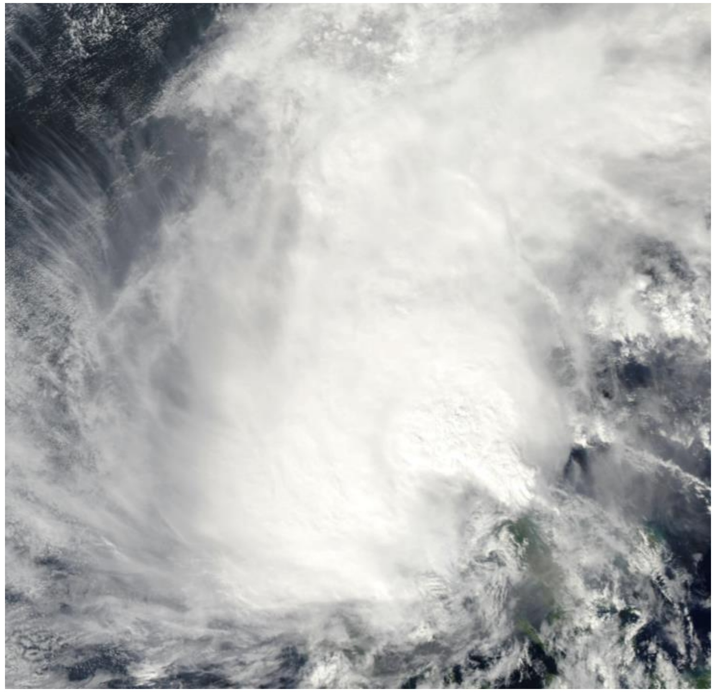
METHODOLOGY

Wind Structure Features

Features (2)
Storm Size / TC Fullness
Storm Strength / Critical Fullness



Super Typhoon Goni (2020)
Large Storm

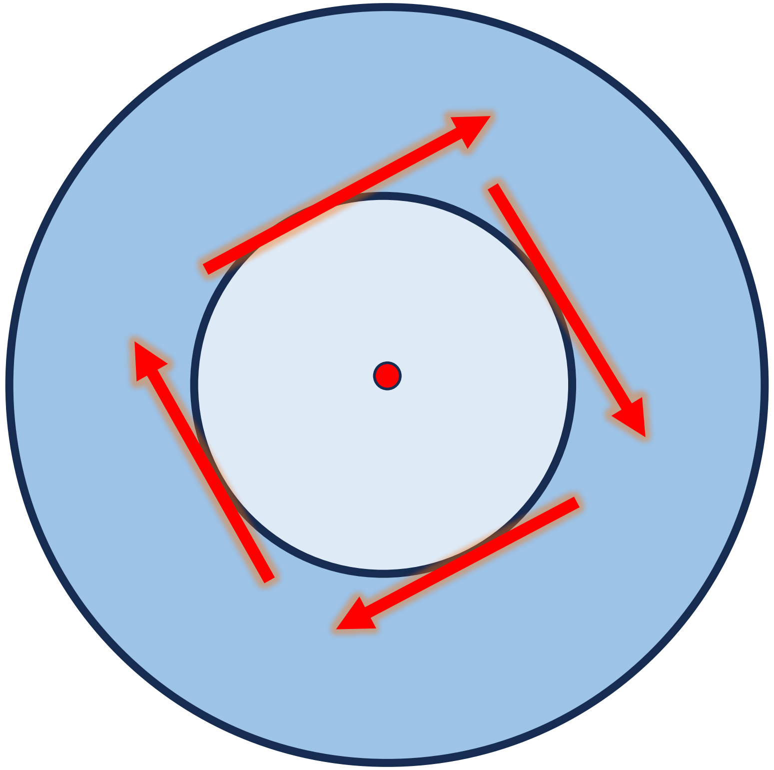


Tropical Depression Winnie (2004)
Small Storm

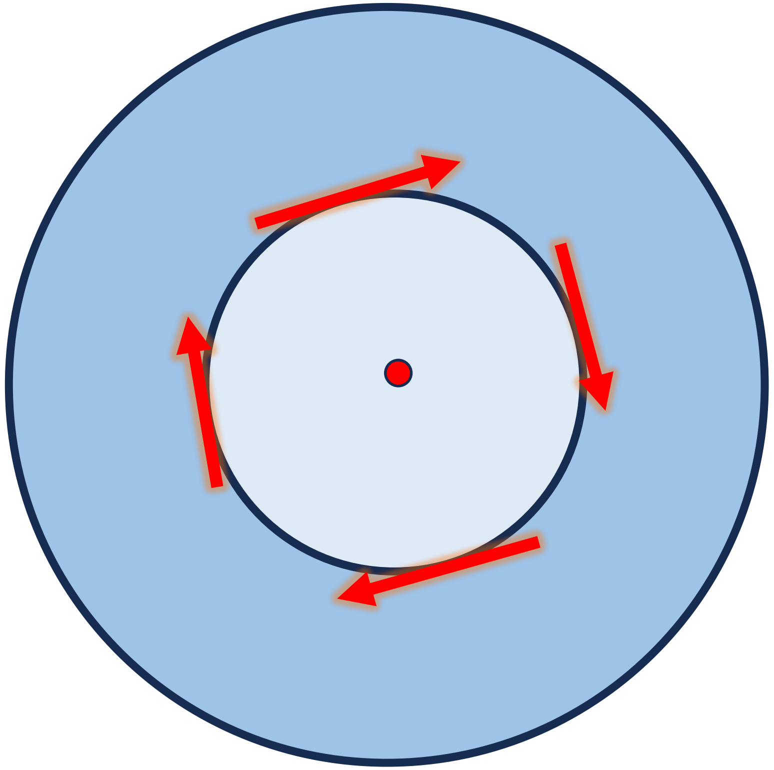
METHODOLOGY Wind Structure Features

Red Arrow = tangential winds

Features (2)
Storm Size / TC Fullness
Storm Strength / Critical Fullness



"Intense" Storm
Strong (Fast) Winds



"Weak" Storm
Weak (Slow) Winds

METHODOLOGY — Experiment Setup

Train Test Split: ***(Yang et al. 2020; Su et al. 2020)***

- **train (2001-2018), test (2019-2022)**
- **reason for this split:** temporal consistency (train on past, predict on future)

Number of Samples:

- **train: 4836** (Non-RI: 3605 & RI: 1231)
- **test: 948** (Non-RI: 699, RI: 249)

Resolve Class imbalance in the training set with SMOTE

- multiple TC RI studies have shown that resolving the class imbalance leads to better model performance ***(Yang 2017; Yang et al. 2020)***
- model tends to biased toward the majority class and hence, the cost function will prioritize minimizing errors on the majority class

$$-\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1-y_i) \cdot \log(1-p(y_i))$$

Experiment Setup

Models: *(Using Scikit-Learn; Pederegosa et al., 2011)*

- Logistic Regression (LogReg), k-Nearest Neighbors (kNN), Support Vector Machines (SVM-Linear, SVM-Polynomial), Gaussian Naive-Bayes (GaussNB), Extra Trees (ETrees)

No Hyperparameter Tuning

Evaluation: (Straightforward evaluation for interpretability, especially to the non-technical public)

- Probability of Detection **(POD; True Positive Rate)**
- False Alarm Ratio **(FAR; False Positive Rate)**

METHODOLOGY Experiment Setup

Experiment 1: Base Model, No Oversampling

- Vmax, RMW, R34, TCF, TCF0, RF

Experiment 2: **Base Model**, with **SMOTE** (oversampling)

- Vmax, RMW, R34, TCF, TCF0, RF

Experiment 3: **With additional features** (time-lagged), with **SMOTE**

- Vmax, RMW, R34, TCF, TCF0, RF, and their previous 6 and 12-hour changes

Experiment 4: **Top 10 features** by SelectKBest, with **SMOTE**

- dTCF0_12H, dVmax_6H, dTCF0_6H, dVmax_12H, RMW, RF, dRMW_12H, TCF0, dTCF_12H, dR34_12H

Experiment 5: **Top 5 features** by SelectKBest, with **SMOTE**

- dTCF0_12H, dVmax_6H, dTCF0_6H, dVmax_12H, RMW

Sample Code

Model Training

```python
# select base features
X_train = df_train[['USA_RMW', 'USA_R34_NE','USA_WIND','TCF', 'TCF0', 'Rf']]
X_train = X_train.astype(float)

X_test = df_test[['USA_RMW', 'USA_R34_NE','USA_WIND','TCF', 'TCF0', 'Rf']]
X_test = X_test.astype(float)

# standardize the features (mean = 0, sd =1 )
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# target
y_train= df_train['RI Category']
y_test= df_test['RI Category']
```

```python
# Apply SMOTE to the training dataset
X_train_exp2, y_train_exp2 = smote.fit_resample(X_train, y_train)

# IMPORTANT: we will not apply SMOTE to the testing set so that we would have a realistic prediction
```

```python
# creating model instance
lr_model = LogisticRegression(random_state = 34)
knn_model = KNeighborsClassifier(n_neighbors=5)
svm_model = SVC(kernel='linear', probability=True, random_state=34)
svmp_model = SVC(kernel='poly', probability=True, random_state=34)
gnb_model = GaussianNB()
et_model = ExtraTreesClassifier(random_state=34)
```

```python
# fitting the models
lr_model.fit(X_train_exp2, y_train_exp2)
knn_model.fit(X_train_exp2, y_train_exp2)
svm_model.fit(X_train_exp2, y_train_exp2)
svmp_model.fit(X_train_exp2, y_train_exp2)
gnb_model.fit(X_train_exp2, y_train_exp2)
et_model.fit(X_train_exp2, y_train_exp2)

# predicting on the test set
lr_model_preds = lr_model.predict(X_test)
knn_model_preds = knn_model.predict(X_test)
svm_model_preds = svm_model.predict(X_test)
svmp_model_preds = svmp_model.predict(X_test)
gnb_model_preds = gnb_model.predict(X_test)
et_model_preds = et_model.predict(X_test)
```

Sample Code

Model Evaluation

```python
print(f"lr Confusion: {lr_conf}")
print(f"knn Confusion: {knn_conf}")
print(f"svm Confusion: {svm_conf}")
print(f"svmp Confusion: {svmp_conf}")
print(f"gnb Confusion: {gnb_conf}")
print(f"et Confusion: {et_conf}")

# Extracting TP, FP, TN, FN from the confusion matrix
TP_lr = lr_conf[1, 1]
FP_lr = lr_conf[0, 1]
TN_lr = lr_conf[0, 0]
FN_lr = lr_conf[1, 0]

TP_knn = knn_conf[1, 1]
FP_knn = knn_conf[0, 1]
TN_knn = knn_conf[0, 0]
FN_knn = knn_conf[1, 0]

TP_svm = svm_conf[1, 1]
FP_svm = svm_conf[0, 1]
TN_svm = svm_conf[0, 0]
FN_svm = svm_conf[1, 0]
```

```python
# Probability of Detection
print(f"lr POD: {(TP_lr/(TP_lr + FN_lr))}")
print(f"knn POD: {(TP_knn/(TP_knn + FN_knn))}")
print(f"svm POD: {(TP_svm/(TP_svm + FN_svm))}")
print(f"svmp POD: {(TP_svmp/(TP_svmp + FN_svmp))}")
print(f"gnb POD: {(TP_gnb/(TP_gnb + FN_gnb))}")
print(f"et POD: {(TP_et/(TP_et + FN_et))}")

# False Alarm Ratio
print(f"lr FAR: {(FP_lr/(FP_lr + TN_lr))}")
print(f"knn FAR: {(FP_knn/(FP_knn + TN_knn))}")
print(f"svm FAR: {(FP_svm/(FP_svm + TN_svm))}")
print(f"svmp FAR: {(FP_svmp/(FP_svmp + TN_svmp))}")
print(f"gnb FAR: {(FP_gnb/(FP_gnb + TN_gnb))}")
print(f"et FAR: {(FP_et/(FP_et + TN_et))}")
```

Sample Code SelectKBest

```
# Looping through each estimator
for name, estimator in estimators.items():

    # Create the SelectKBest object and rank each feature
    selector = SelectKBest(score_func=f_classif, k=10)
    X_train_exp4_sel = selector.fit_transform(X_train_exp4, y_train_exp4)
    X_test_exp4_sel = selector.transform(X_test)

    # fitting the models
    estimator.fit(X_train_exp4_sel, y_train_exp4)

# predicting on the test set
lr_model_preds = estimators['lr_model'].predict(X_test_exp4_sel)
knn_model_preds = estimators['knn_model'].predict(X_test_exp4_sel)
svm_model_preds = estimators['svm_model'].predict(X_test_exp4_sel)
svmp_model_preds = estimators['svmp_model'].predict(X_test_exp4_sel)
gnb_model_preds = estimators['gnb_model'].predict(X_test_exp4_sel)
et_model_preds = estimators['et_model'].predict(X_test_exp4_sel)
```

Mean and SD

Exploratory Data Analysis

The mean and SD of the RI and Non-RI TCs were compared for both testing and training sets.

- For both sets, RI TCs generally have higher mean Vmax, RMW, R34, TCF, TCF0, and RF with less spread than Non-RI TCs.
  - This physically suggests that RI TCs, on average, are more intense, have smaller inner and outer cores, and exhibit larger fullness with narrower distributions.

- As for most delta (time-lagged) features, RI TCs generally have higher means with less spread for testing and training sets.
  - This also physically suggests that RI TCs, on average, are changing their intensities, size, and fullness more quickly than non-RI TCs.

Table 4. Mean and Standard Deviation of the training and testing sets for RI and Non-RI.

	RI				Non-RI			
Feature	**Train Mean**	**Train SD**	**Test Mean**	**Test SD**	**Train Mean**	**Train SD**	**Test Mean**	**Test SD**
Vmax	72.23	19.97	70.71	19.28	72.10	26.32	63.07	25.37
RMW	21.54	7.52	19.45	8.43	24.82	12.85	29.74	20.29
R34	102.21	36.25	116.90	46.11	110.57	43.85	118.98	55.64
TCF	0.75	0.14	0.80	0.13	0.74	0.17	0.70	0.20
TCF0	0.52	0.11	0.50	0.13	0.48	0.18	0.40	0.19
RF	1.50	0.44	1.73	0.74	1.79	1.18	2.32	1.97
dVmax_6H	7.06	4.89	7.10	6.55	3.74	5.69	3.62	5.97
dRMW_6H	-2.23	5.17	-3.22	6.87	-1.06	6.28	-2.67	9.06
dR34_6H	7.31	11.59	7.03	14.85	5.08	12.29	4.84	17.32
dTCF_6H	0.05	0.10	0.06	0.10	0.03	0.17	0.06	0.17
dTCF0_6H	0.05	0.04	0.06	0.04	0.03	0.04	0.03	0.05
dRF_6H	-0.07	0.52	-0.21	0.92	-0.05	0.96	-0.05	2.00
dVmax_12H	12.77	7.97	11.92	11.60	7.18	10.10	6.81	11.05
dRMW_12H	-4.73	7.81	-6.24	9.79	-2.38	9.71	-5.01	13.56
dR34_12H	14.94	18.47	14.15	24.89	10.41	21.44	11.10	26.98
dTCF_12H	0.13	0.21	0.14	0.24	0.08	0.23	0.14	0.30
dTCF0_12H	0.11	0.07	0.11	0.08	0.06	0.08	0.07	0.09
dRF_12H	-0.01	2.10	-0.34	2.58	-0.08	1.66	0.01	2.98

t-test

Exploratory Data Analysis

The **t-test revealed** that there is a statistically significant difference at the 0.05 confidence level for some features between the testing and training sets for both Non-RI and RI cases

- differences in the distribution of the features between the **two independent sets could be a potential source of error in the model's prediction** (probably due to temporal variability – different average atmospheric conditions)

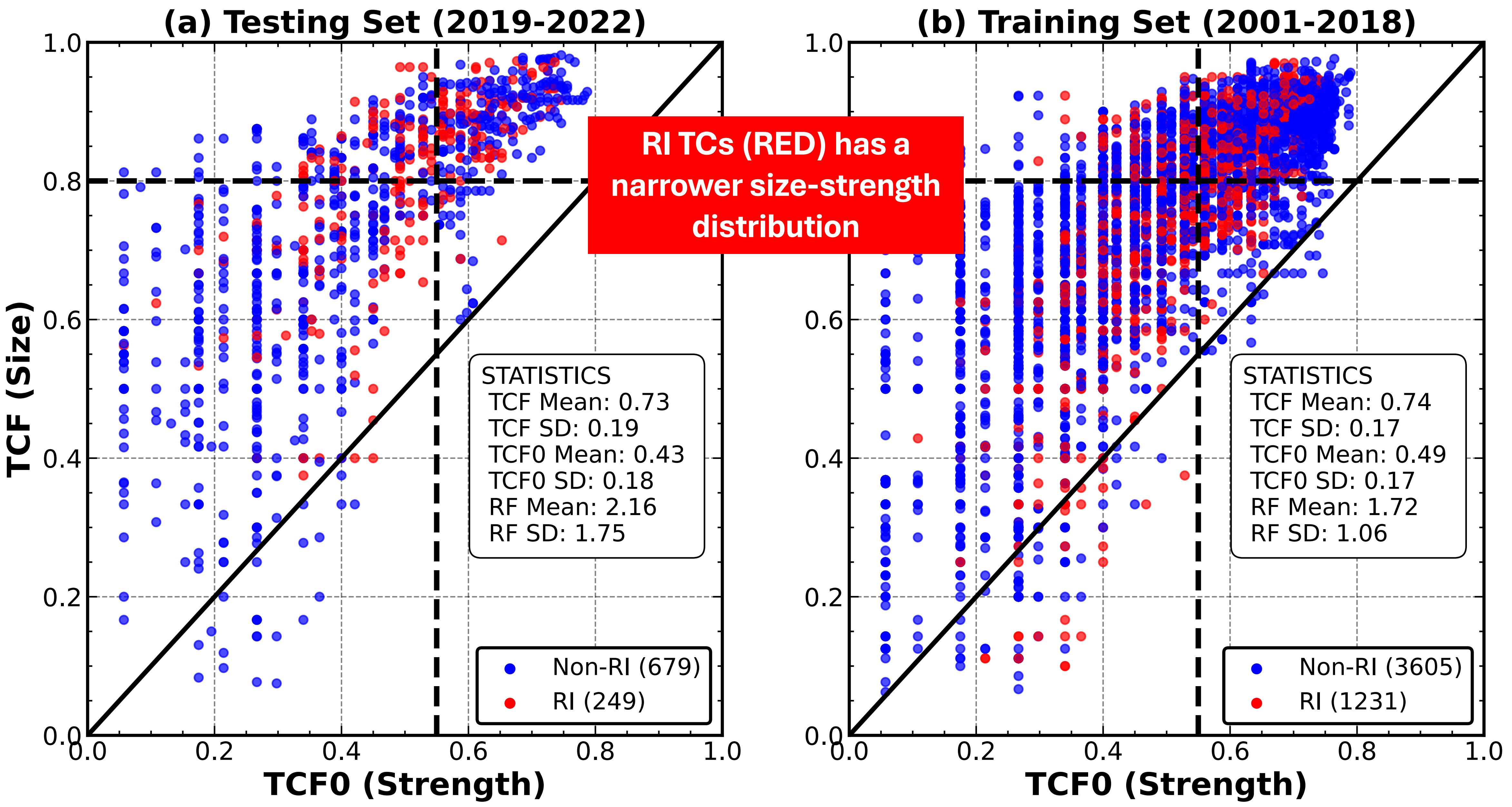
Some features exhibit a statistically significant difference at the 0.05 confidence level between Non-RI and RI TCs for both testing and training sets.

- **some features can better discriminate RI TCs from non-RI TCs**

Table 5. Statistically significant features from the t-test at 0.05 significance level.

COMPARISON	FEATURES
Training Set (RI vs Non-RI)	RMW, R34, TCF, TCF0, RF, dVmax_6H, dVmax_12H, dRMW_6H, dRMW_12H, dR34_6H, dR34_12H, dTCF_6H, dTCF_12H, dTCF0_6H, dTCF0_12H
Testing Set (RI vs Non-RI)	RMW, Vmax, TCF, TCF0, dVmax_6H, dVmax_12H, dTCF0_6H, dTCF0_12H
RI (Training vs Testing Set)	R34, TCF, TCF0, RF, dRMW_6H, dRMW_12H, dRF_6H, dRF_12H
Non-RI (Training vs Testing Set)	RMW, R34, Vmax, TCF, TCF0, RF, dRMW_6H, dRMW_12H, dTCF_6H, dTCF_12H, dTCF0_6H, dTCF0_12H

Wind Structure | Exploratory Data Analysis

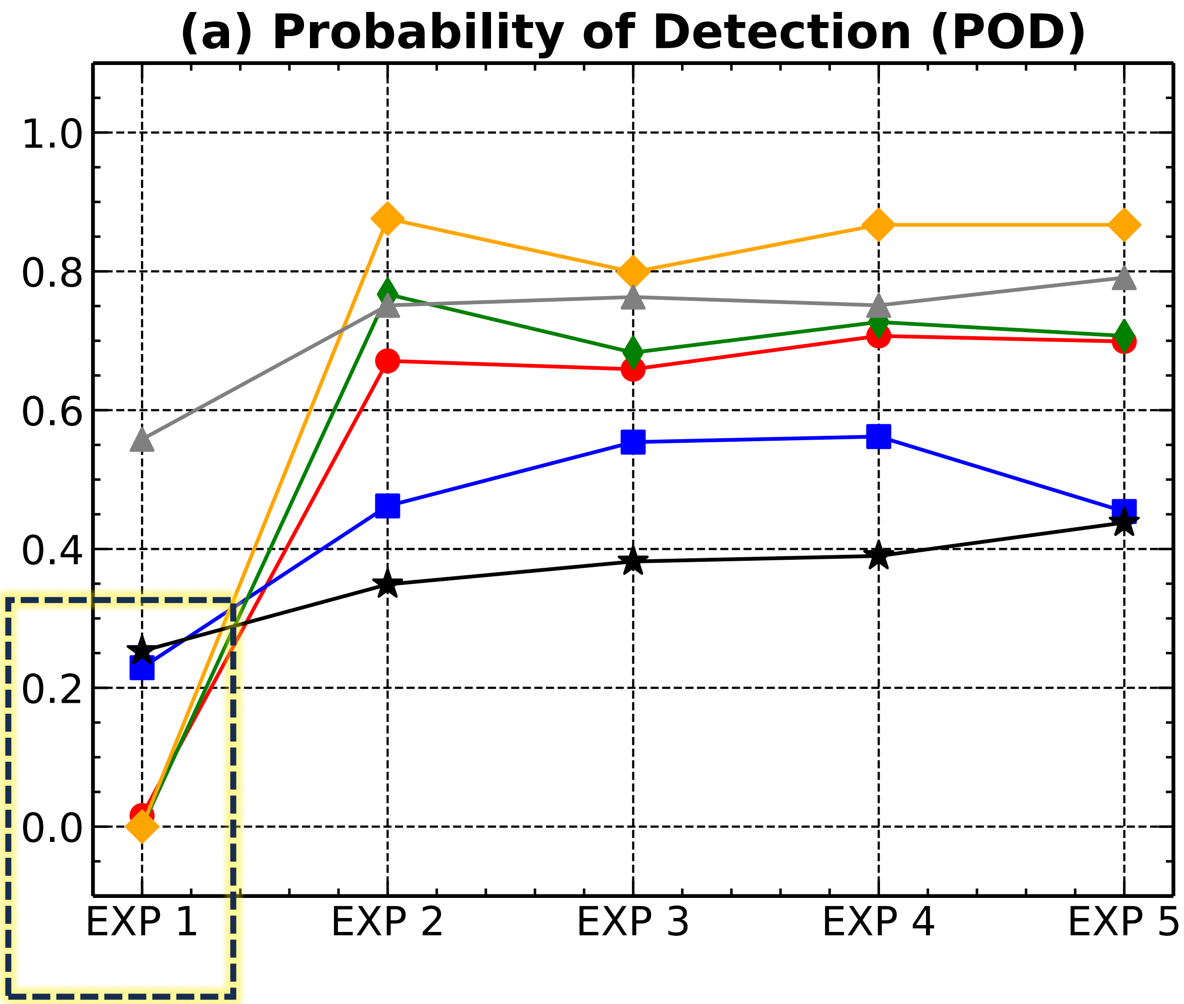


RI TCs (RED) has a narrower size-strength distribution

TABLE OF CONTENTS

FINDINGS

Model Evaluation



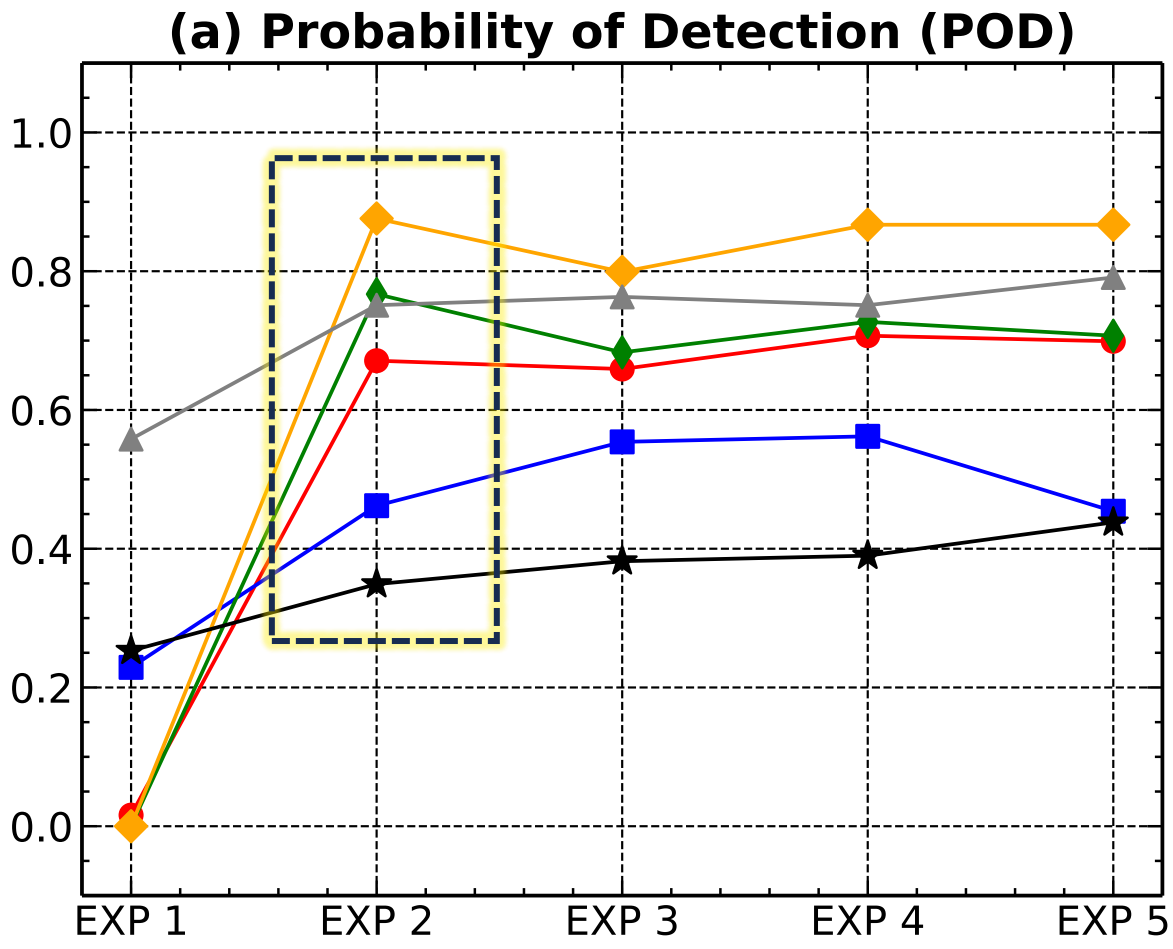
The base model (EXP 1), as expected, exhibited the worst performance due to class imbalance. The models might prioritize optimizing accuracy for the majority class at the expense of misclassifying instances for the minority class.

Employing SMOTE (EXP 2) led to improved performance across all models because the models now learned from balanced instances of Non-RI and RI.

Adding more features (EXP 3) slightly increased (decreased) the performance for GaussianNB, kNN, and ETrees (SVM-P, SVM-L, LogReg). The additional features may not have contained significant predictive information.

FINDINGS

Model Evaluation



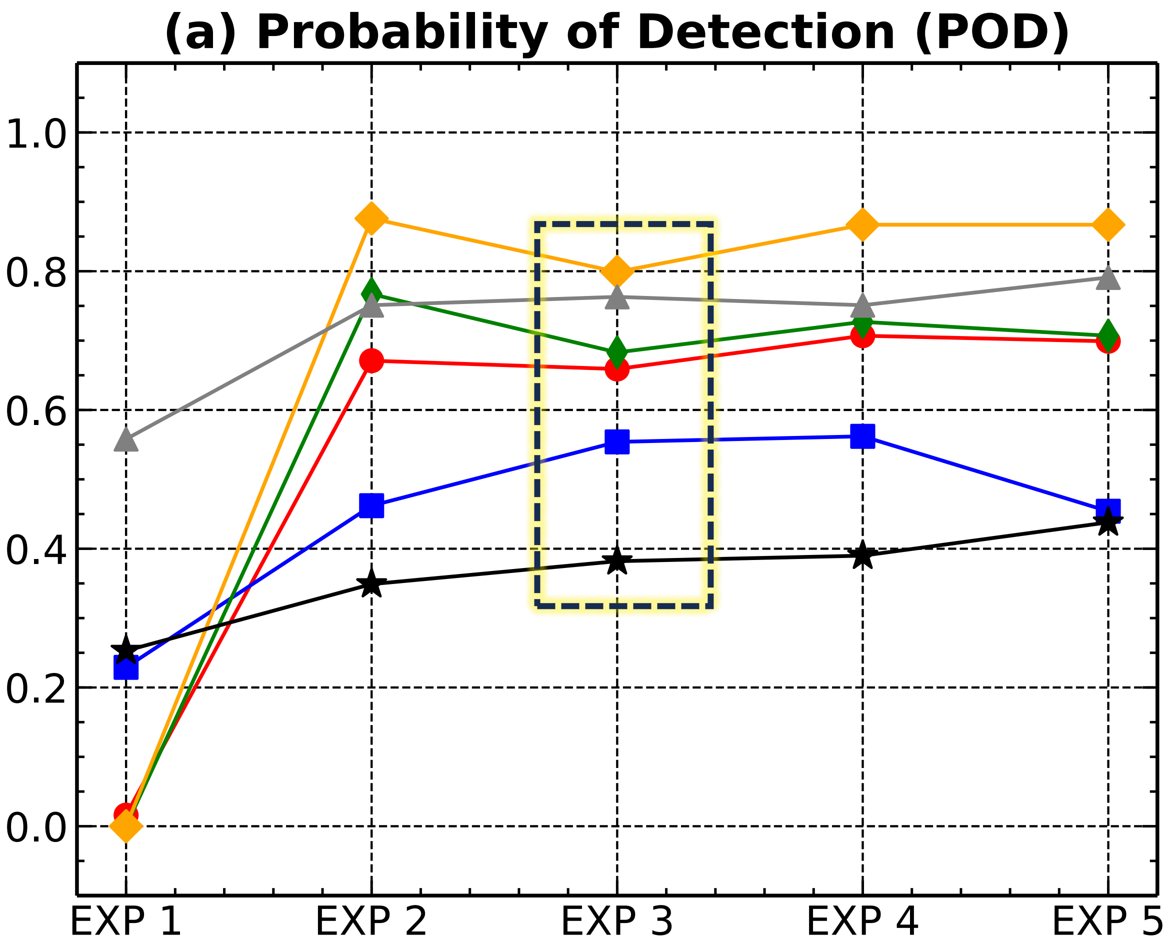
The base model (EXP 1), as expected, exhibited the worst performance due to class imbalance. The models might prioritize optimizing accuracy for the majority class at the expense of misclassifying instances for the minority class.

Employing SMOTE (EXP 2) led to improved performance across all models because the models now learned from balanced instances of Non-RI and RI.

Adding more features (EXP 3) slightly increased (decreased) the performance for GaussianNB, kNN, and ETrees (SVM-P, SVM-L, LogReg). The additional features may not have contained significant predictive information.

FINDINGS

Model Evaluation



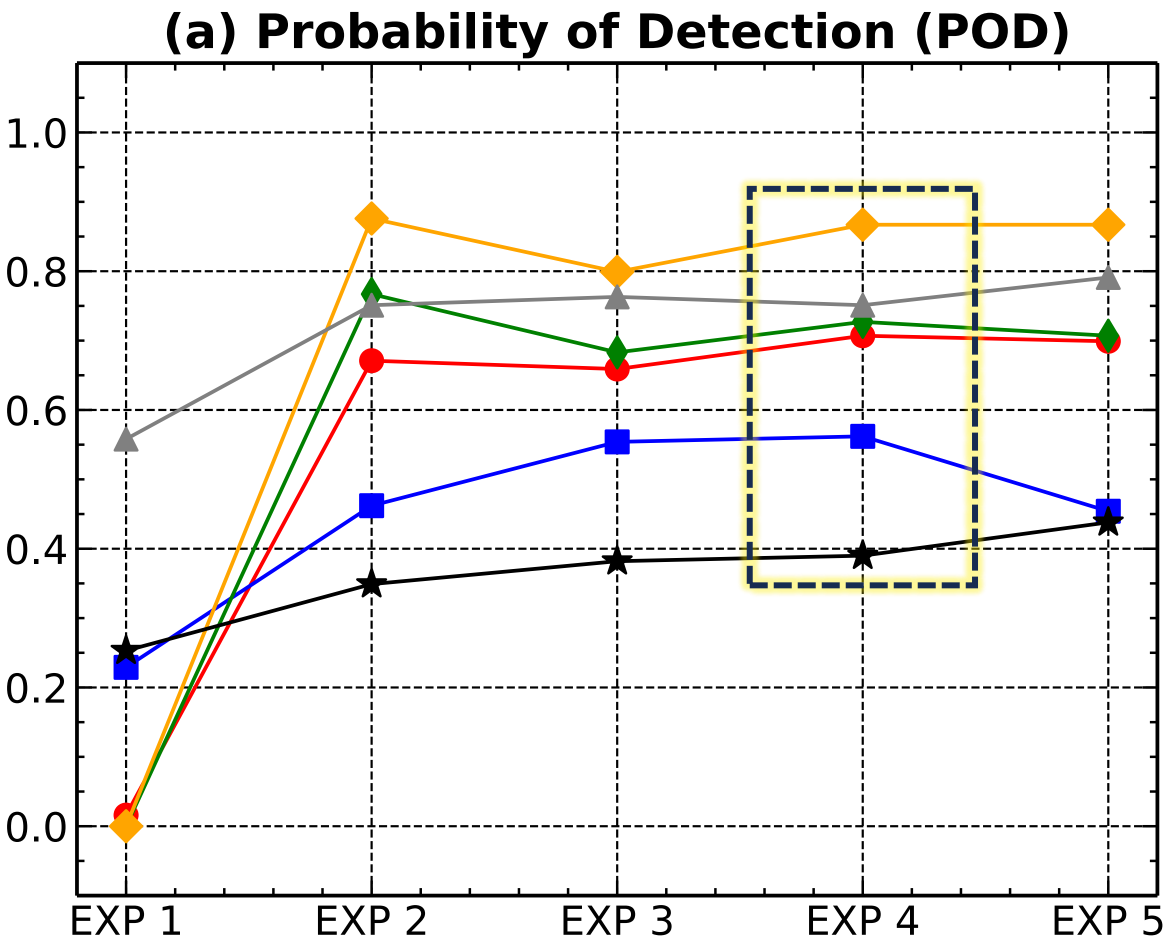
The base model (EXP 1), as expected, exhibited the worst performance due to class imbalance. The models might prioritize optimizing accuracy for the majority class at the expense of misclassifying instances for the minority class.

Employing SMOTE (EXP 2) led to improved performance across all models because the models now learned from balanced instances of Non-RI and RI.

Adding more features (EXP 3) slightly increased (decreased) the performance for GaussianNB, kNN, and ETrees (SVM-P, SVM-L, LogReg). The additional features may not have contained significant predictive information.

FINDINGS

Model Evaluation



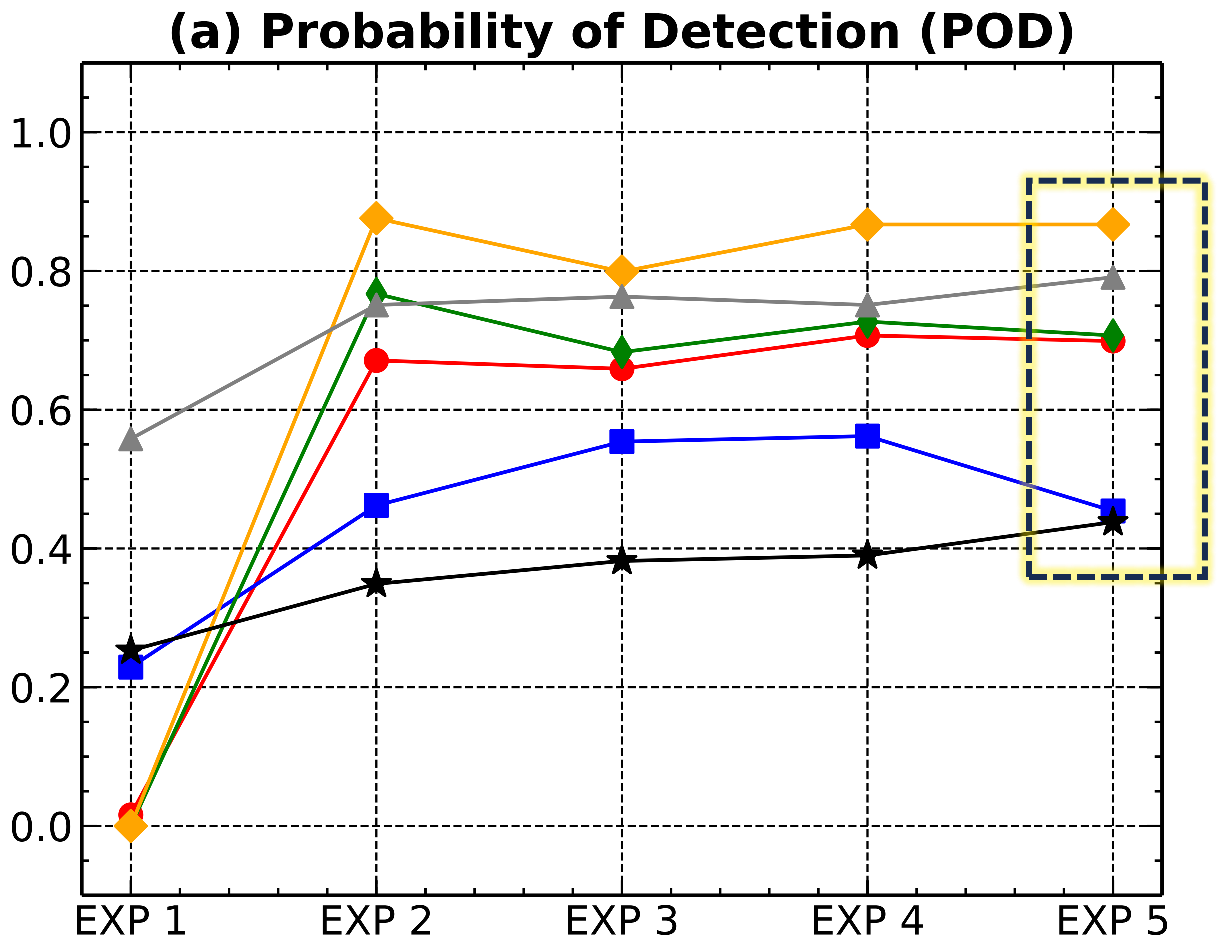
To reduce potential overfitting and increase model generalization, the ten best features were used as predictors in EXP 4. **The POD performance slightly increased for all models except for GaussNB.**

Further decreasing the features to the five best (EXP 5), only GaussNB and ETrees have seen improvements in POD performance.

- The GaussNB assumes feature independence; hence, the smaller subset of features might be beneficial.
- ETrees also benefitted from the reduced subset of features since there is potentially less overfitting.

FINDINGS

Model Evaluation



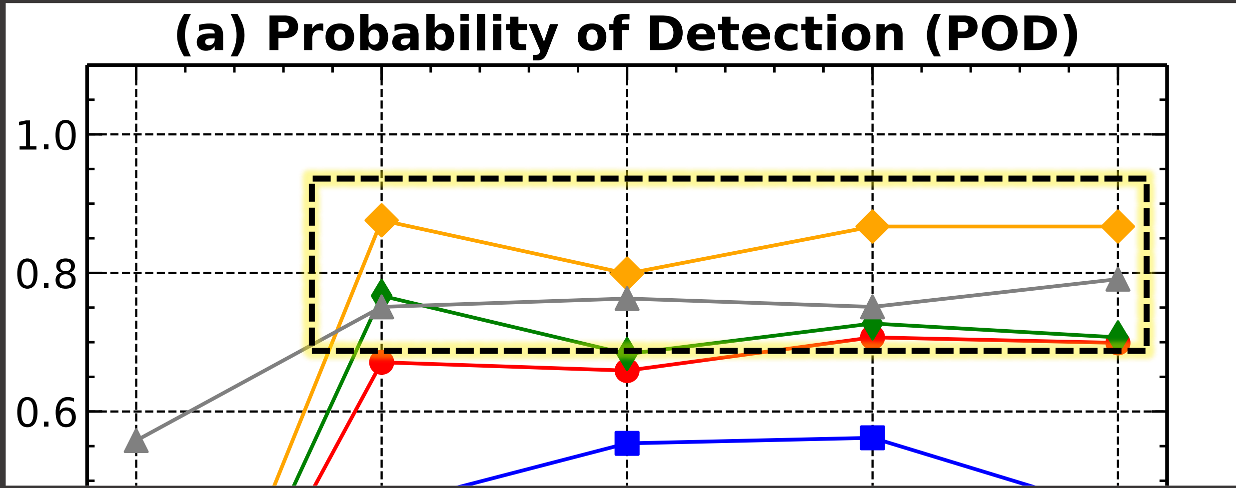
To reduce potential overfitting and increase model generalization, the ten best features were used as predictors in EXP 4. The POD performance slightly increased for all models except for GaussNB.

Further decreasing the features to the five best (EXP 5), only GaussNB and ETrees have seen improvements in POD performance.

- The GaussNB assumes feature independence; hence, the smaller subset of features might be beneficial.
- ETrees also benefitted from the reduced subset of features since there is potentially less overfitting.

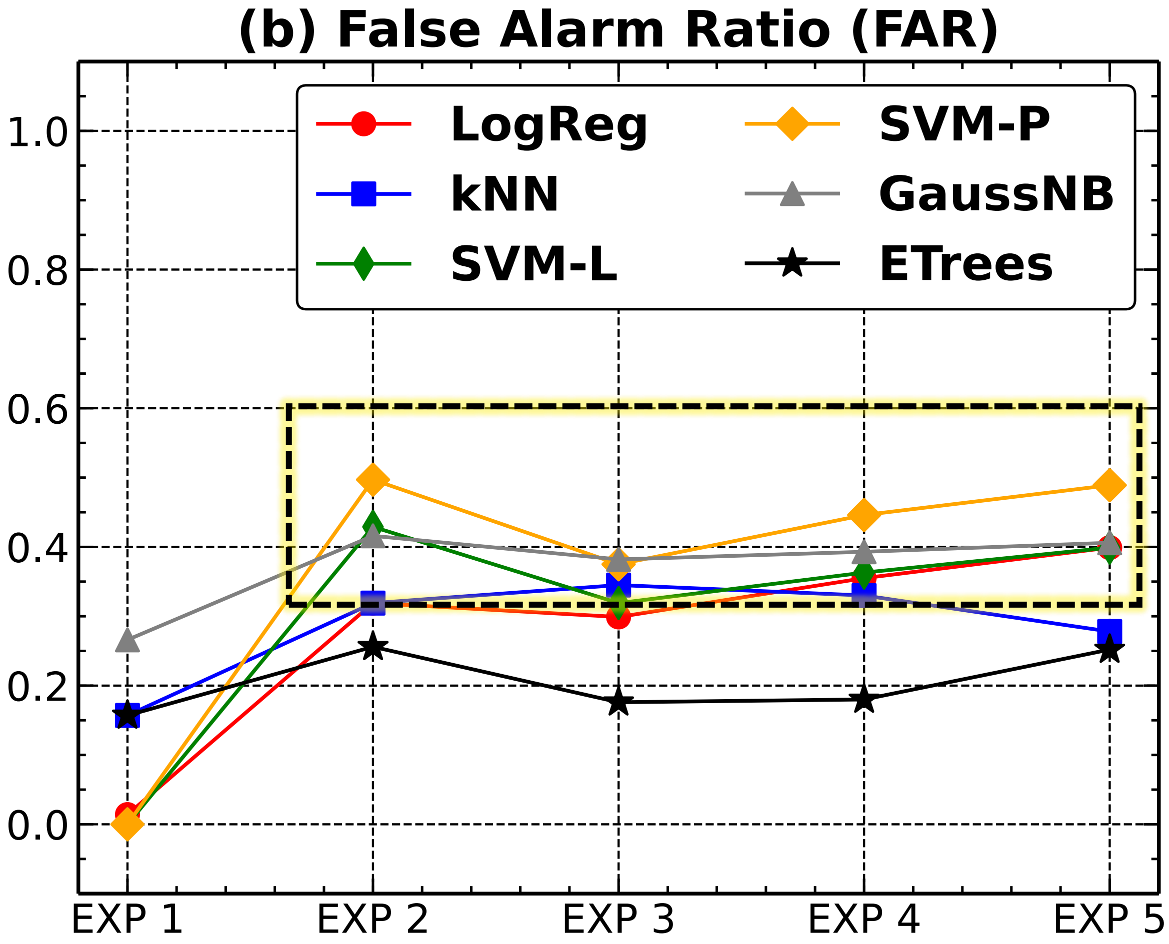
FINDINGS

Model Evaluation



The **SVM-P** and **GaussNB** have the highest POD among the models, and they also have the highest FAR.

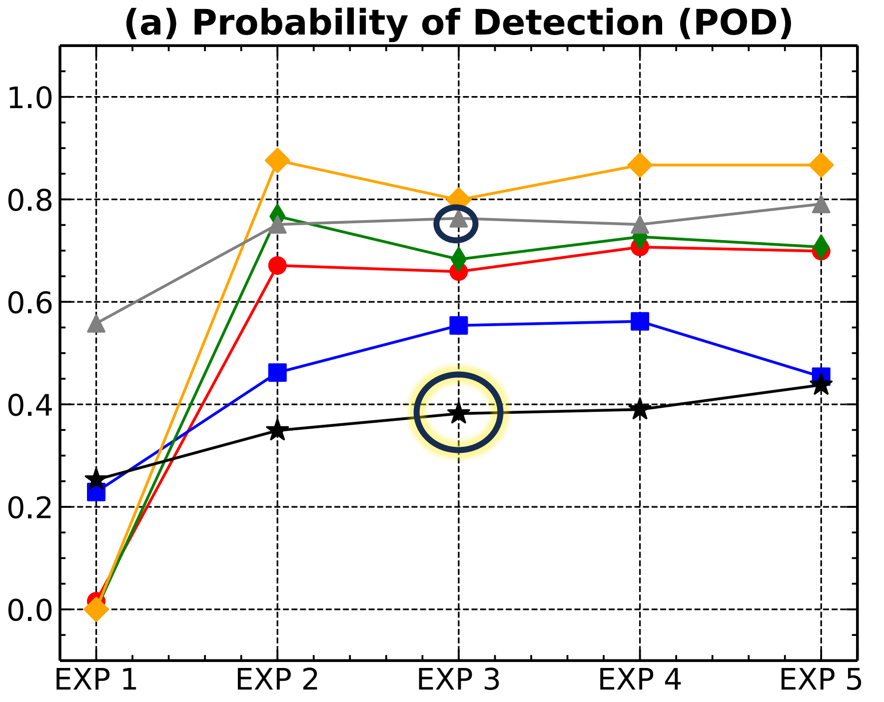
When POD increases, FAR also tends to increase, similar to multiple studies (Yang, 2016; Su et al., 2020, Chen et al., 2023), and is particularly common in predictive modeling.

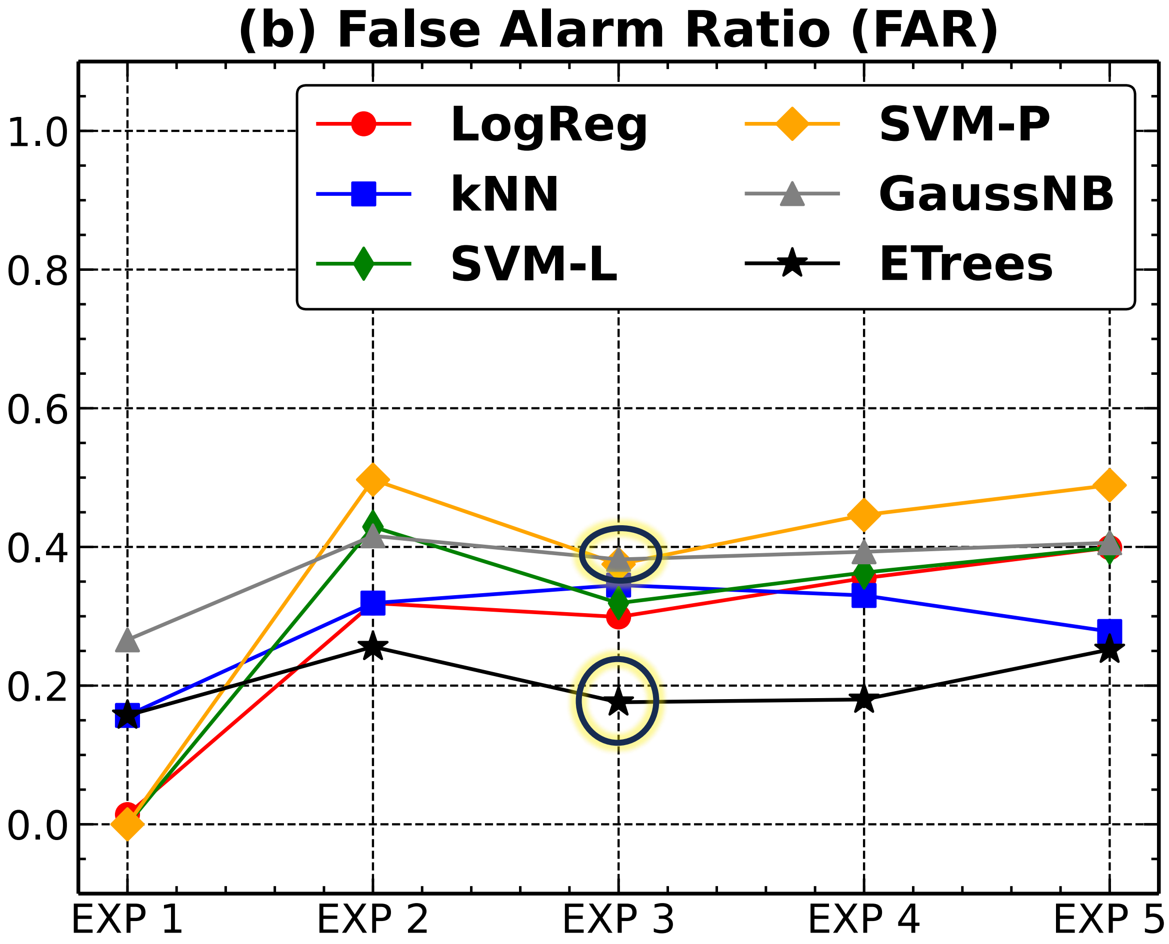


The model is still not robust if it also detects RI when it should not.

FINDINGS

Model Evaluation





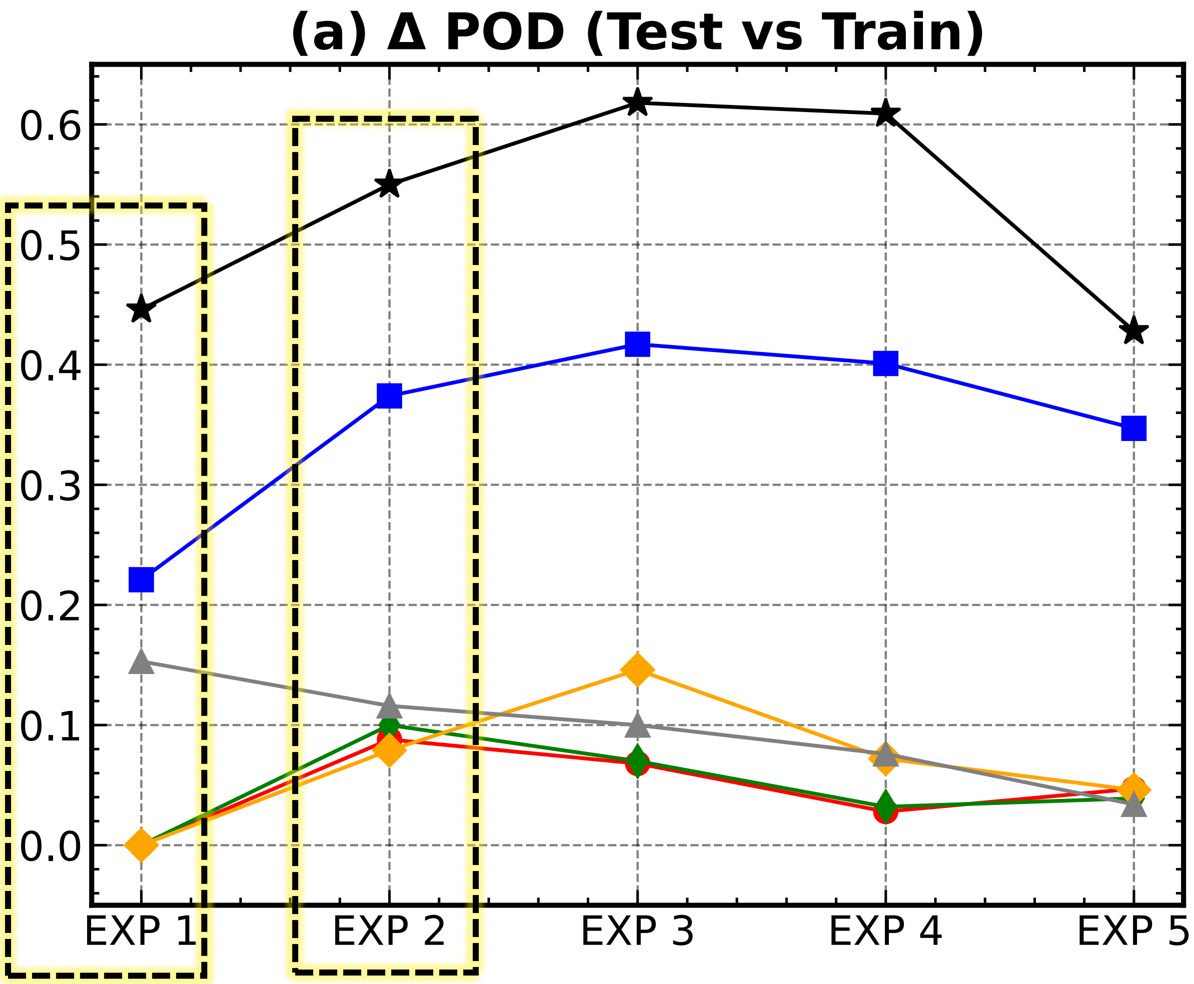
ETrees and **GaussNB** in EXP 3 have shown that a **simultaneous increase in POD and a decrease in FAR is possible.**

ETrees and **GaussNB** can potentially benefit from more features (EXP 3) to reduce model bias and increase fit.

The model is still not robust if it also detects RI when it should not.

FINDINGS

Model Evaluation



To further explore which model has the best generalization, the difference between POD and FAR is shown for Testing vs Training set.

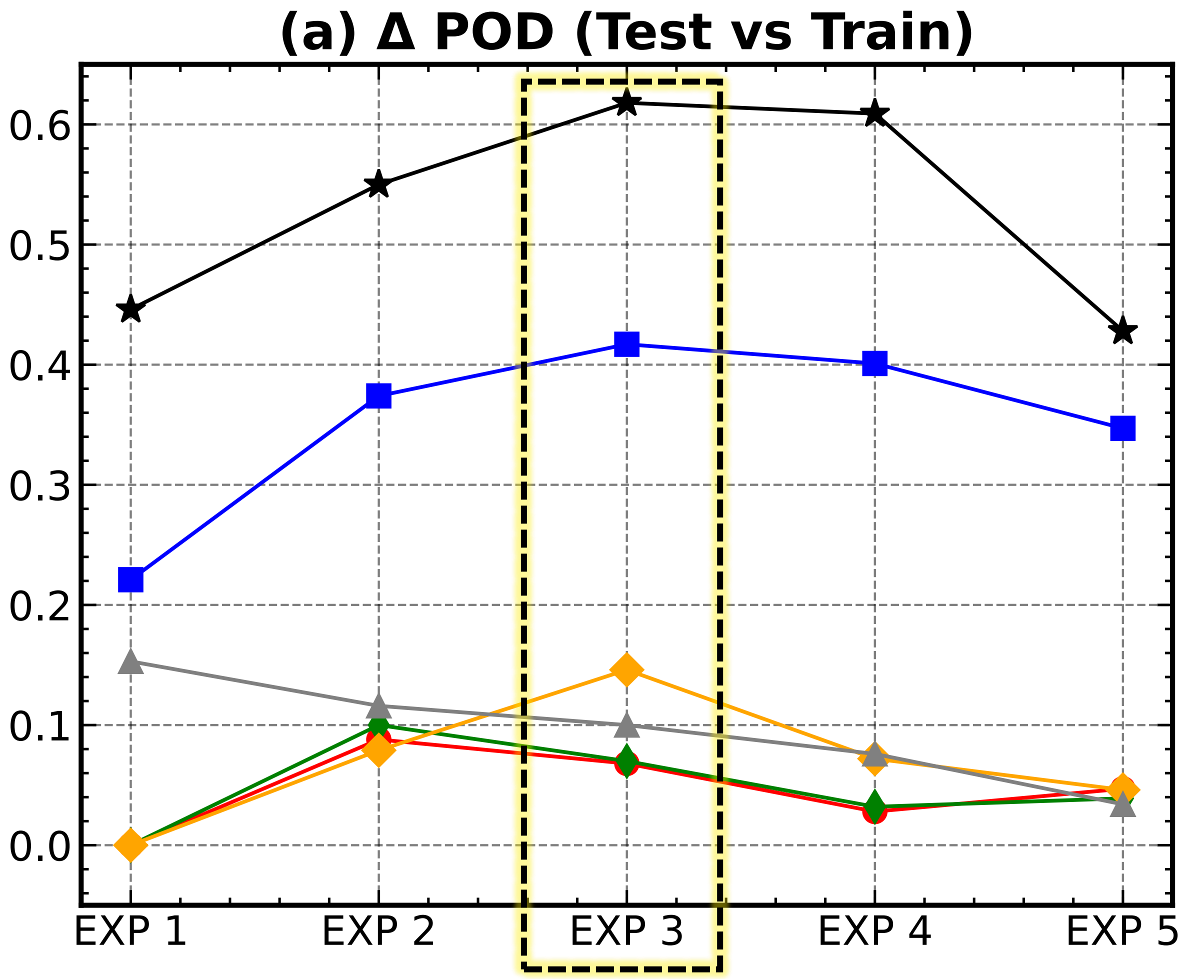
The most minor POD difference for all models except for GaussNB is in EXP 1.

After applying SMOTE (EXP 2) to deal with class imbalance, the POD difference increased for all models except again for GaussNB.

Adding more features (EXP 3) resulted in an increase in POD difference for some models (ETrees, kNN, SVM-P), which indicates overfitting, but decreased for the others (GaussNB, SVM-L, LogReg), which instead indicates better model generalization.

FINDINGS

Model Evaluation



To further explore which model has the best generalization, the difference between POD and FAR is shown for Testing vs Training set.

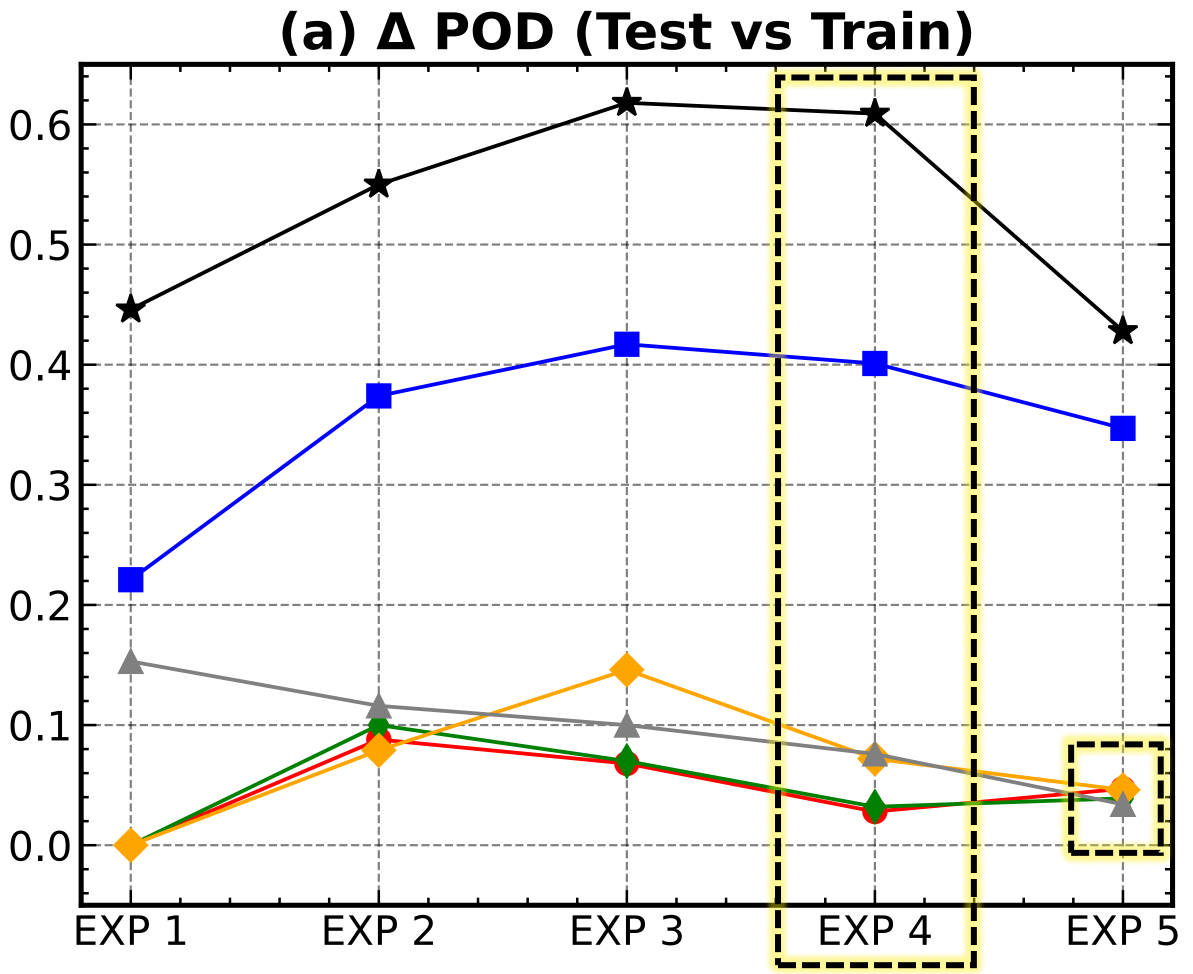
The most minor POD difference for all models except for GaussNB is in EXP 1.

After applying SMOTE (EXP 2) to deal with class imbalance, the POD difference increased for all models except again for GaussNB.

Adding more features (EXP 3) resulted in an increase in POD difference for some models (ETrees, kNN, SVM-P), which indicates overfitting, but decreased for the others (GaussNB, SVM-L, LogReg), which instead indicates better model generalization.

FINDINGS

Model Evaluation



When feature selection (ten best features) was applied (EXP 4), the POD difference decreased for all models, indicating that all models may generalize better.

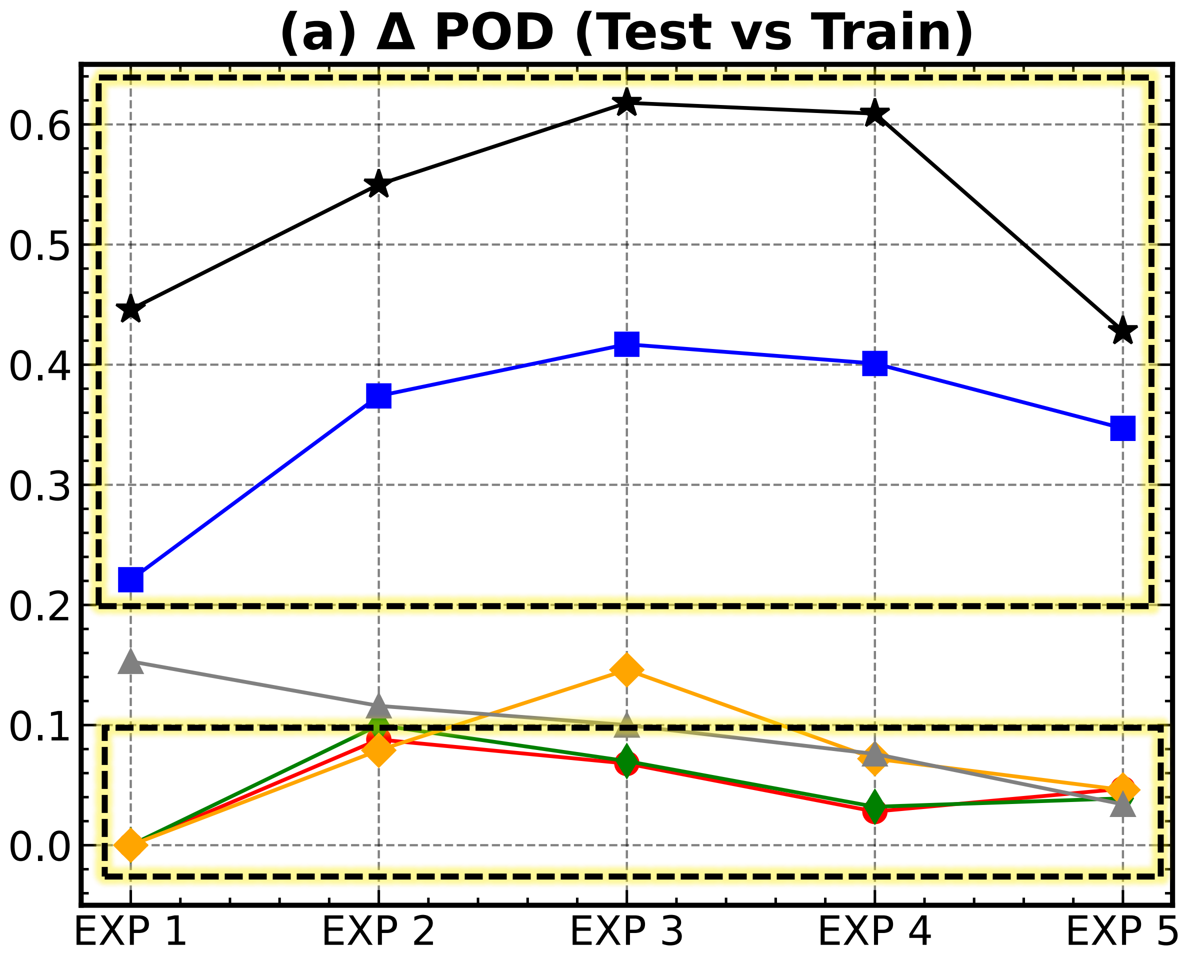
Further reducing features down to five best (EXP 5) led again to a reduction of POD for all models except for SVM-L and LogReg.

The models with the highest POD differences across all experiments are ETrees and kNN (maybe more susceptible to overfitting due to their complexity and sensitivity to local variations in the data)

POD difference of under 0.10 was observed for SVM-L and LogReg across all experiments.

FINDINGS

Model Evaluation



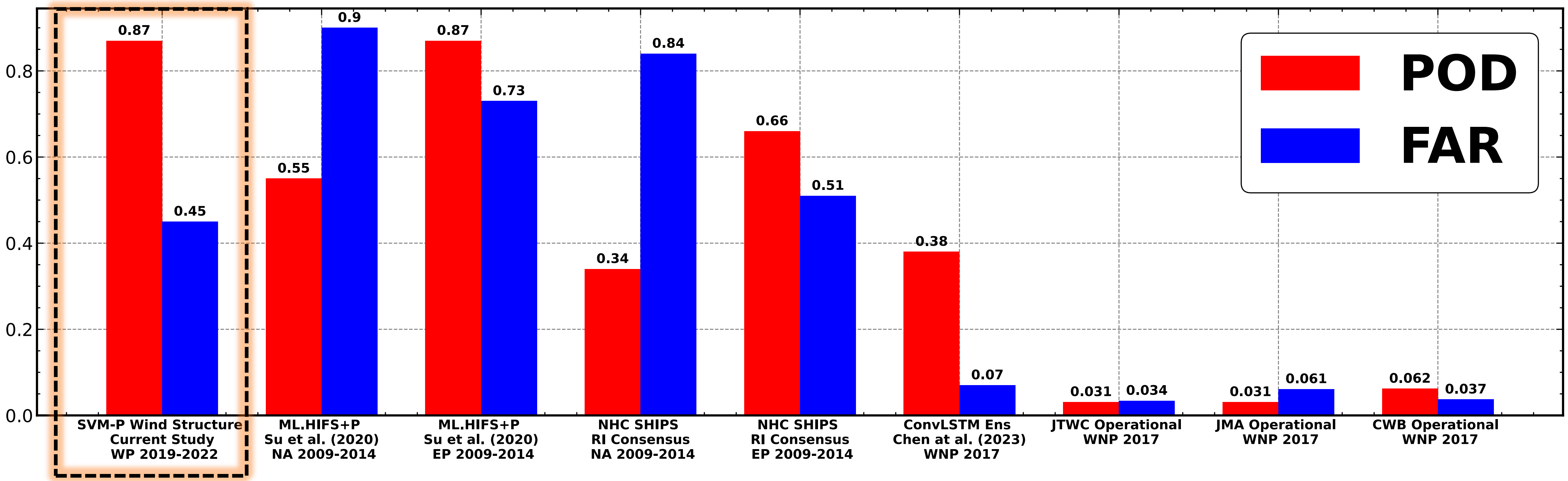
When feature selection (ten best features) was applied (EXP 4), the POD difference decreased for all models, indicating that all models may generalize better.

Further reducing features down to five best (EXP 5) led again to a reduction of POD for all models except for SVM-L and LogReg.

The models with the highest POD differences across all experiments are ETrees and kNN (maybe more susceptible to overfitting due to their complexity and sensitivity to local variations in the data)

POD difference of under 0.10 was observed for SVM-L and LogReg across all experiments (simple models tends to capture less variance).

FINDINGS
Model Evaluation



The best model of this study performed better than operational models in terms of detecting RI. However, the high FAR is still a concern. It is important to note that this is not a fair comparison since these models were tested on different time periods and ocean basins.

- Different time periods can exhibit different TC climatological characteristics
- Ocean basins each have regional TC characteristics (Kaplan et al. 2010)
- The testing set used in this study is filtered for intensifying cases only
- To realistically compare our model with operational ones, we need to validate our model with respect to operational forecasts

TABLE OF CONTENTS

CONCLUSIONS FUTURE WORK

- The models should be validated using an unfiltered dataset to **reflect real forecasting situations**
- Other **oversampling, feature engineering, and feature selection techniques, and using a validation set** should also be considered to achieve a good tradeoff between the model's ability to detect (POD) and misclassify RI (FAR)
- Wind structure features can be **integrated into a more sophisticated model** that includes environmental predictors, satellite observations, and internal dynamical features for a more holistic TC RI forecasting model
- **Explore ensemble ML models** (e.g., Random Forest, Adaptive Boosting, Extreme Gradient Boosting) and Neural Network architectures such as Long-Short Term Memory models to capitalize on the temporal dependency of the evolution of TCs

CONCLUSIONS

Experiment 1: Base model with no oversampling

- All models struggled due to the class imbalance, resulting in the extremely poor detection of RI events (except for GaussNB)

Experiment 2: Base model, with SMOTE (oversampling)

- Incorporating SMOTE resulted in improved performance across all models. Oversampling has enabled the models to better learn from the minority class.
- Increased skill in detecting RI (higher POD) also came with an increase in misclassification (higher FAR)

CONCLUSIONS

Experiment 3: Adding additional time-lagged features, with SMOTE

- Models like GaussianNB, kNN, and ETrees saw a slight improvement in POD, while a decrease in POD was observed for models such as LogReg, SVM-L, and SVM-P.
- Reduced the bias for GaussianNB, kNN, and ETrees while it increased the variance of LogReg, SVM-L, and SVM-P

Experiments 4 & 5: Top ten & five best features with SelectKbest, with SMOTE

- feature selection generally enhanced model performance.
- further reducing features led to poorer model performance due to less complexity and underfitting.
- GaussNB and ETrees models exhibited the most improvements with the least features.

Best Model in terms of detecting RI:

- SVM-P (EXP 2), POD: 0.87, FAR: 0.45

AUTHOR BACKGROUND

Henric R. Jandoc
Legazpi City, Albay
Sorsogon City, Sorsogon

EDUCATION

- Bachelor of Science in Meteorology – BICOL UNIVERSITY
  - 2019-2023

WORK EXPERIENCE

- **Part Time Lecturer** – Bicol University (Physics & Meteorology Department)
  - **Phys 134:** Waves and Optics (currently)
  - **Phys 133:** Electricity, Magnetism, & Optics
  - **Meteo108:** Computational Methods for Meteorology
  - **Meteo114:** Mesoscale Meteorology

- **External Thesis Advisor for Undergraduate**
  - **Study 1:** Flood Risk Mapping of Davao City using Ensemble Machine Learning Models
  - **Study 2:** Wind Structure and Deep Convective Clouds of Intensifying Tropical Cyclones in the Western North Pacific

AUTHOR BACKGROUND

Henric R. Jandoc
Legazpi City, Albay
Sorsogon City, Sorsogon

RESEARCH PUBLICATIONS / PROJECTS

[2023] Environmental Conditions and Internal Dynamical Processes during the Rapid Intensification of Super Typhoon Goni (2020) – LEAD AUTHOR
Proceedings of the Philippine Meteorological Society Volume 6 – Recent Advances and Applications of Meteorology and its Allied Sciences in the Philippines *https://philmetsoc.com/PMS/publications*, **ISSN 2599-5537**

[2023] On the Intensification of Super Typhoon Goni (2020), Part 2: Internal Dynamical Processes (ORAL PRESENTATION) – LEAD AUTHOR
7th Taiwan-Philippines Earth Sciences International Conference – National Cheng Kung University @ Tainan, Taiwan
http://www.edsrc.ncku.edu.tw/tpesic/Abstract.pdf

[TBD] On the Intensification of Super Typhoon Goni (2020): Environmental Conditions, Deep Convective Clouds, Precipitation, Cloud Microphysics, and Wind Structure – LEAD AUTHOR
To be submitted to the Asia-Pacific Journal of Atmospheric Sciences as LEAD AUTHOR

REFERENCES

Bhatia, K., Baker, A., Yang, W., Vecchi, G., Knutson, T., Murakami, H., Kossin, J., Hodges, K., Dixon, K., Bronselaer, B., & Whitlock, C. (2022). A potential explanation for the global increase in tropical cyclone rapid intensification. *Nature Communications, 13(1)*. https://doi.org/10.1038/s41467-022-34321-6

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16, 321-357*. https://doi.org/10.1613/jair.953

Chen, B. F., Kuo, Y. T., & Huang, T. S. (2023). A deep learning ensemble approach for predicting tropical cyclone rapid intensification. *Atmospheric Science Letters, 24, e1151*. https://doi.org/10.5555/5555

DeMaria, M., Sampson, C. R., Knaff, J. A., & Musgrave, K. D. (2014). Is tropical cyclone intensity guidance improving? *Bulletin of the American Meteorological Society, 95(3), 387-398*. https://doi.org/10.1175/bams-d-12-00240.1

Guo, X., & Tan, Z.-M. (2022). Tropical cyclone intensification and fullness: The role of storm size configuration. *Geophysical Research Letters, 49(16), e2022GL098449*. https://doi.org/10.1029/2022GL098449

Hendricks, E. A., Peng, M. S., Fu, B., & Li, T. (2010). Quantifying environmental control on tropical cyclone intensity change. *Monthly Weather Review*, *138*(8), 3243-3271. https://doi.org/10.1175/2010mwr3185.1

Kaplan, J., & Demaria, M. (2003). Large-scale characteristics of rapidly intensifying tropical cyclones in the North Atlantic basin. Weather and Forecasting, 18(6), 1093–1108. https://doi.org/10.1175/1520-0434(2003)018<1093:lcorit>2.0.co;2

Kaplan, J., Demaria, M., & Knaff, J. A. (2010). A revised tropical cyclone rapid intensification index for the Atlantic and eastern North Pacific basins. Weather and Forecasting, 25(1), 220–241. https://doi.org/10.1175/2009waf2222280.1

Knapp, K. R., Diamond, . J., Kossin, J. P., Kruk, M. C., & Schreck, C. J. (2018). International Best Track Archive for Climate Stewardship (IBTrACS) Project, Version 4. NOAA

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830. https://doi.org/10.48550/arXiv.1201.0490

Ruan, Z., & Wu, Q. (2018). Precipitation, convective clouds, and their connections with tropical cyclone intensity and intensity change. Geophysical Research Letters, 45(2), 1098–1105. https://doi.org/10.1002/2017gl076611

Su, H., Wu, L., Jiang, J. H., Pai, R., Liu, A., Zhai, A. J., et al. (2020). Applying satellite observations of tropical cyclone internal structures to rapid intensification forecast with machine learning. Geophysical Research Letters, 47(17), e2020GL089102. https://doi.org/10.1029/2020gl089102

Wang, W., Zhang, Z., Cangialosi, J. P., Brennan, M., Cowan, L., Clegg, P., Takuya, H., Masaaki, I., Das, A. K., Mohapatra, M., Sharma, M., Knaff, J. A., Kaplan, J., Birchard, T., Doyle, J. D., Heming, J., Moskaitis, J., Komaromi, W. A., Ma, S., … Blake, E. (2023). A review of recent advances (2018–2021) on tropical cyclone intensity change from operational perspectives, Part 2: Forecasts by operational centers. Tropical Cyclone Research and Review, 12(1), 50-63. https://doi.org/10.1016/j.tcrr.2023.05.003

Wilks, D. S. (2006). Statistical Methods in the Atmospheric Sciences (2nd ed.). Academic Press.

Yang, Q., Lee, C. Y., & Tippett, M. K. (2020). A long short-term memory model for global rapid intensification prediction. Weather and Forecasting, 35(4), 1203–1220. https://doi.org/10.1175/WAF-D-19-0199.1

Yang, R. (2016). A systematic classification investigation of rapid intensification of Atlantic tropical cyclones with the SHIPS database. Weather and Forecasting, 31, 495–513. https://doi.org/10.1175/WAF-D-15-0029.1

FINDINGS

SUPPLEMENTAL SLIDE

As for the FAR difference, most cases (models + experiments) were under 0.20. Similar trends for LogReg, SVM-L, and SVM-P were observed across all experiments.

The ETrees model, which has the highest POD difference across all experiments, was also the model with the highest FAR difference.