AN ADVANCED ARTIFICIAL INTELLIGENCE SYSTEM FOR INVESTIGATING THE TROPICAL CYCLONE RAPID INTENSIFICATION

by

Yijun Wei

A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Computational Sciences and Informatics

Committee:

_____	Dr. Jason Kinser, Committee Chair
_____	Dr. Ruixin Yang, Committee Member
_____	Dr. Igor Griva, Committee Member
_____	Dr. Olga Gkountouna, Committee Member
_____	Dr. Jason Kinser, Department Chairperson
_____	Dr. Donna M. Fox, Associate Dean, Office of Student Affairs & Special Programs, College of Science
_____	Dr. Fernando R. Miralles-Wilhelm, Dean, College of Science
Date: _____	Fall Semester 2020 George Mason University Fairfax, VA

AN ADVANCED Artificial Intelligence System for Investigating the Tropical Cyclone Rapid Intensification

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

by

Yijun Wei
Master of Art
University of Michigan – Ann Arbor, 2013
Master of Science
University of Michigan – Ann Arbor, 2013

Director: Ruixin Yang, Associate Professor
Geography and Geoinformation Science

Fall Semester 2020
George Mason University
Fairfax, VA

DEDICATION

I dedicate this dissertation to my parents Wu Gao and Yong Wei, my advisor Dr. Ruixin Yang and my dear friends.

ACKNOWLEDGEMENTS

I would like to thank many friends, relatives, and supporters who have made this happen. Dr. Ruixin Yang, advised me and gave me great help for conducting research. My parents, Wu Gao and Yong Wei, supported me spiritually. Dr. Kinser, Dr. Griva, and Dr. Gkountouna were of invaluable help. Finally, thanks to all the people who help me along this long journey.

TABLE OF CONTENTS

LIST OF TABLES

Table

Page

LIST OF FIGURES

Figure

Page

ABSTRACT

AN ADVANCED ARTIFICIAL INTELLIGENCE SYSTEM FOR INVESTIGATING THE TROPICAL CYCLONE RAPID INTENSIFICATION

Yijun Wei, Ph.D.

George Mason University, 2020

Dissertation Director: Dr. Ruixin Yang

Tropical cyclones (TCs) can cause heavy casualties due to storm surge, high wind gusts, heavy rainfall and flooding, and landslides, so predicting TC is important. There are mainly two elements of TC forecasting: tracking prediction and intensity prediction. So far, it is found that tracking prediction is more mature than the intensity prediction. Various models are developed for TC intensity prediction and can be simple enough to run for a few seconds or complex enough to run for a couple of hours on a supercomputer. Although with so many models are developed, the intensity prediction accuracy is still very low, and one primary reason is the existence of Rapid Intensification (RI).

Currently, most RI prediction studies are conducted based on a subset of the SHIPS database using a relatively simple model structure. However, variables (features) in the SHIPS database are built upon expert knowledge in TC intensity studies, and the variable values are derived from gridded model outputs or satellite observations. Are there any more

important variables in TC intensity predictions but not identified in the SHIPS dataset? In this study, two AI-based techniques are used to extract new features from a widely used comprehensive gridded reanalysis data set. The original SHIPS data, and the newly derived features are used as inputs to an artificial intelligence (AI) for the RI prediction.

This study first constructs a complicated artificial intelligence (AI) system, the COR-SHIPS model, based on the complete SHIPS dataset that handles feature engineering and selection, imbalance, prediction, and hyper parameter-tuning simultaneously. The COR-SHIPS model is derived to improve the performance of the current researches in RI prediction and to identify other essential SHIPS variables that are ignored by previous studies with variable importance scores. COR-SHIPS is also used as the baseline model in the dissertation.

To distill new variables from vast amounts of gridded data, two models, with a similar structure to the COR-SHIPS model but with an additional data filters, are designed in the dissertation to identify new features related to TC intensity changes in general and RI in particular. Here, we adopt the Local linear embedding (LLE) and deep learning (DL) techniques respectively to filter the near center and large-scale spatial data of the European Centre for Medium-Range Weather Forecasts (ECMWF) ERA-Interim reanalysis data, one of the best reanalysis products at the moment, for identifying new variables related to RI, and term the corresponding LLE-SHIPS model and DL-SHIPS model, respectively.

The result of the three models outperforms most of the earlier studies by at least approximately 30%, 60%, and 75%, respectively. In addition to the well-known SHIPS database, we specify the 400 and 450 hPa wind speeds, identify 1000 hPa potential vorticity

and vertical pressure speed, and differentiate humidity southeast, vorticity north, and eastward wind north to the TC centers that could help the prediction and understanding the occurrence of RI.

CHAPTER 1 INTRODUCTION

1.1 Background

Tropical cyclones (TCs) can cause heavy casualties due to storm surge, high wind gusts, heavy rainfall and flooding, and landslides (Pacific Disaster Center. n.d.). On May 2, 2008, Cyclone Nargis sent a storm surge in Myanmar and killed at least 138,000 people (Enz et al. 2009). Prediction of the behavior of TCs can minimize deaths and losses. Therefore, skillful TC prediction is significant to mitigate risk by timely planning and preparation.

The first known TC forecast was conducted by Lt. Col. William Reid of the Corps of Royal Engineers in the western hemisphere in 1846, and barometric pressure was used as the basis for Reed's approach (Reid 1846). Most forecasts before 1900 were obtained by direct observation at weather stations through the telegraph. Significant changes were made in data collection since 1900, where radiosondes (1930), aircraft (1943), coast weather radar (1950), and weather satellite (1960) were introduced (Sheets 1990).

There are mainly two elements of TC forecasting: track forecasting and intensity forecasting. These two predictions are critical in disaster prevention, but the development of these two presents a difference. So far, it is found that tracking prediction is more mature than intensity prediction. DeMaria et al. (2007) examined the National Hurricane Center (NHC) and Joint Typhoon Warning Center operational TC intensity forecasts for the three major northern hemisphere TC basins (Atlantic, eastern North Pacific, and

western North Pacific) for the past two decades. The intensity forecasts were compared to the track forecasts for the same data sample. The performance of the two forecasts was comparable at 12 h, but the track forecasts were 2 to 5 times more skillful by 72 h, with the largest ratio in the western Pacific. As lead time increases, tracking prediction became more skillful than intensity prediction. Cangialosi and Franklin (2017) indicated that in the Atlantic and Pacific, the skill of track prediction is at least 3-7 times larger than that of intensity in 12, 24, 36 hours, and 10-40 times more skillful by 48, 72, 96, and 120 hours in 2016. Since the tracking forecasting is relatively accurate, and the intensity forecasting is with low skills, recent research on TC prediction mainly focuses on intensity forecasting on a time scale from 12 hours to 120 hours (Cangialosi and Franklin 2017).

Various models were developed for TC intensity prediction and can be simple enough to run for a few seconds or complex enough to run for a couple of hours on a supercomputer. Based on the mechanism, these models can be characterized as the dynamical model, the statistical model, and the statistical-dynamical model.

1.1.1 Dynamical models

Dynamical models, also known as the numerical models, consider complex physical processes and are used on the supercomputer to solve the ordinary and partial differential equations in physics. One of the most critical and influential models is the Geophysical Fluid Dynamics Laboratory (GFDL) model (Kurihara et al. 1998), which was used for a research purpose during 1973 and 1980. Encouraged by the performance of the GFDL model, the research model (GFDL) was converted to a comprehensive

prediction system that started in the mid-1980s. The process took about 15 years, and GFDL model became operational in 1995. The interpolated form of GFDL model (GFDI) became available for intensity in 1996, and a U.S. Navy's version of GFDI was added in 1999 (Rennick 1999). Another widely used dynamical model is the hurricane weather research and forecast (H-WRF) model, which became operational in 2007 (Miller 2007).

1.1.2 Statistical Model

A statistical model does not include the physics of the atmosphere but instead is based on the relationship between specific information and behavior of TCs. The first statistical model was developed in 1972 to help generate TC track forecasts, and the model was named Climatology and Persistence (CLIPER5). In 1979, the Statistical Hurricane Intensity Forecast (SHIFOR) model, which consisted of climatology and persistence variables, began operational for TC intensity prediction (Jarvinen and Neumann 1979). A 5-day SHIFOR version (SHIFOR5) was implemented in 2001 (Knaff et al. 2003). Decay-SHIFOR5 is a form of SHIFOR5 that includes a weakening component when TCs move inland, and Decay-SHIFOR5 modifies intensity over land using CLIPER track and climatological decay rate (Rhome 2007).

A well-known statistical model is the Statistical Hurricane Intensity Prediction Scheme (SHIPS) developed by DeMaria and Kaplan (1994). In SHIPS, different multiple regression models with persistence, synoptic, and climatological variables were derived to predict TC intensity changes in 12, 24, 36, 48, 72 hours for the Atlantic basin. The storm intensity, i.e., the dependent variable, is measured by the maximum 1-min sustained surface wind, and an independent variable (predictor) is considered significant

112 if the probability that the regression coefficient is different from 0 with exceeds 95%
113 confidence.

114 The candidate variables were displayed in Table 1.1. Among those, JDATE,
115 VMX, DVMX, LAT, LONG, USM, VSM, and CSM were climatological and persistence
116 variables, and POT, SHR, DSHR, REFC, PEFC, SIZE, and DTL were synoptic variables.
117 POT took the effect of sea surface temperature (SST) into account since SST is closely
118 related to TC intensification (Merrill 1987). SHR and DSHR were used to evaluate the
119 vertical shear, and plenty of studies have shown vertical shear of the horizontal wind has
120 a negative influence on TC intensification (e.g., Gray 1967; Merrill 1988). REFC and
121 PEFC are included to account for positive interactions between the TC and synoptic-scale
122 systems. SIZE was included as a measure of the extent of the outer circulation of the TC.
123 Although all landfall cases were eliminated from the data, the proximity to land might
124 still have a modifying influence on the storm intensity, and DTL was involved in the
125 model. A simple backward-stepping procedure was conducted for the variable selection
126 and POT, SHR, DVMX, REFC, PRFC, JDATE, LONG, DTL, SIZE, and DSHR were
127 selected as the variables for the multiple regression model. The model was tested using
128 the Jackknife procedure, and the result indicated that the intensity errors were 10% - 15%
129 smaller than the errors from a model that used only climatology and persistence
130 (SHIFOR5). However, the forecast only explained about 50% of the variability of the
131 observed intensity change, which indicates that a statistical model with large-scale
132 variables is not able to explain all types of storm process effects adequately (DeMaria and
133 Kaplan 1994).

Table 1.1: Candidate variables and their abbreviations included in DeMaria and Kaplan (1994).

Variable	Abbreviation
Absolute value of Julian date – 253	JDATE
Initial storm intensity	VMX
Intensity change during previous 12 h	DVMX
Initial storm latitude	LAT
Initial storm longitude	LONG
Eastward component of storm motion vector	USM
Northward component of storm motion vector	VSM
Magnitude of storm motion vector	CSM
Maximum possible intensity - initial intensity	POT
Magnitude of 850-200-mb vertical shear	SHR
Time tendency of vertical shear magnitude	DSHR
The 200-mb relative eddy angular momentum flux convergence	REFC
The 200-mb planetary eddy angular momentum flux convergence	PEFC
The 850-mb relative angular momentum	SIZE
Distance to nearest major landmass	DTL

1.1.3 Statistical-Dynamical Model.

Statistical-dynamical model blends the statistical model and the dynamical model (NHC Track and Intensity Models 2017). In other words, the statistical-dynamical model employs variables derived from the dynamical models. Although SHIPS proposed in DeMaria and Kaplan (1994) was regarded as a statistical or statistical-synoptic model. In 1997, SHIPS was converted to a statistical-dynamical model by using large-scale

144 variables in Global Forecast System (GFS) (DeMaria et al. 2005). Therefore, the SHIPS
145 after 1997 is regarded as a statistical-dynamical model.
146 DeMaria et al. (2005) described modifications to the NHC operational SHIPS
147 intensity model from 1997 to 2003, including an additional method to account for the use
148 of variables from the dynamical model in 1997, the storm decay over land in 2000, the
149 extension of the forecasts from 3 to 5 days in 2001, and the use of the GFS, a global
150 numerical computer model run by National Oceanic and Atmospheric Administration
151 (NOAA) in 2001. The study showed that SHIPS performs well in predicting 72 h
152 intensity in the Atlantic, and at 48 and 72 h in the east Pacific. The inclusion of the
153 effects of the decay over land beginning in 2000 reduces the short period Atlantic
154 intensity error but not for 72 h forecasting. An experimental version of SHIPS consisted
155 of satellite variables during the 2002 and 2003 seasons significantly improved skill in the
156 east Pacific forecasts by up to 7% at 12–72 h, and 3.5% through 72 h in Atlantic forecasts
157 (DeMaria et al. 2005).
158 The latest version of SHIPS, which has an inland decay component, was known
159 as Decay-SHIPS (DSHIPS). The DSHIPS typically provides more accurate TC intensity
160 forecasts when TCs encounter or interact with the land. Over open water with no land
161 interactions, the intensity forecasts from DSHIPS and SHIPS are identical (Rhome 2007).
162 As SHIPS model is mainly used in Atlantic and northeast Pacific, a similar model
163 known as the Statistical Typhoon Intensity Prediction Scheme (STIPS) was developed for
164 the northwest Pacific Ocean and Southern Hemisphere by Knaff et al. (2005).

The values of SHIPS variables, available openly, are considered as the SHIPS database. Every year, instances from the previous year and new variables from other sources may be added to the database, while some old variables may be removed. The development of the SHIPS database was described in DeMaria and Kaplan (1994, 1999), DeMaria et al. (2005), and Kaplan et al. (2010, hereafter, KDK10). The most recent version of the database is the SHIPS Developmental Data, a complete dataset with known different types of the parameter related to TC intensity changes (SHIPS 2018a). The SHIPS database was used by many TC intensity related types of research. One such example is the logistic growth equation model (LGEM) (DeMaria 2009), which was also a type of statistical-dynamical TC intensity model and used the same input as SHIPS but in the framework of a simplified dynamical prediction system. LGEM estimated the only parameter in LGEM - population growth rate, which is proportional to the maximum sustained wind, using four free parameters. These four parameters were the time-dependent growth rate, maximum potential intensity (MPI), and two constants that determine how quickly the intensity relaxes toward the MPI, i.e., vertical shear (S) and a convective instability parameter (C). LGEM was found to explain observed intensity variations better than SHIPS (DeMaria and Kaplan 1994). LGEM-MR, a version of LGEM, where the remaining parameters are determined by a multiple regression method using a subset of the SHIPS database, came to work in real-time from 2006. The average skill of LGEM-MR forecasts is up to 17% better than those from the SHIPS model (DeMaria 2009).

Since a large number of intensity forecasting models came into available, the consensus prediction model (ICON) was developed by Sampson et al. (2008). A three models consensus of DSHIPS, GFDI, and GFNI (The interpolated Navy version of GFDL hurricane model) were found outperform almost all the single intensity forecasting model in the Atlantic basin.

1.2 The problem

Although the statistical-dynamical models have been used since early 1990, the prediction accuracy is still not high. One primary reason is the existence of Rapid Intensification (RI) (Kaplan and DeMaria 2003; DeMaria et al. 2005; Yang et al. 2007).

1.2.1 SHIPS-RII model (KD03)

RI was defined in Kaplan and DeMaria (2003, hereafter, KD03), as the maximum sustained surface wind speed increase of 30 kt or more over a 24-h period, and KD03 derived the initial version of the Statistical Hurricane Intensity Prediction Scheme Rapid Intensification Index (SHIPS-RII) model for RI prediction for the Atlantic basin. A two-sided t-test was utilized in KD03 to determine if the 16 different variables, listed in Table 1.2, display significant differences in RI instances and non-RI instances.

Table 1.2: Candidate variables and their abbreviations included in KD03.

Variable	Abbreviation
Maximum sustained surface wind speed	VMAX
Latitude	LAT
Longitude	LON
Storm speed	SPD
Intensity change during the previous 12 h	DVMX
Storm motion	USTM
The absolute value of (Julian date - 253)	JDAY

Sea surface temperature	SST
Maximum possible intensity - initial intensity	POT
850–200-hPa vertical shear	SHR
200-hPa wind	U200
200-hPa temperature	T200
850–700-hPa relative humidity	RHLO
850-hPa relative vorticity	Z850
200-hPa relative eddy angular momentum flux	REFC
Steering layer	SLYR

The result indicated that

- 11 of 16 variables except for VMAX, SPD, JDAY, T200, and Z850 show significantly difference at the 95% significance level;
- And among them, 10 variables except for LON show significantly difference at the 99% significance level;
- 7 of them, i.e., LAT, DMAX, SST, RHLO, POT, SHR, and U200, show significantly difference at the 99.9% significance level

in RI instances and non-RI instances.

Variables that are significant at 95%, 99%, and 99.9% confidence level were used to evaluate the composite probability of RI, respectively, and variables at 99.9% level were found to have the highest probability. However, LAT and U200 were further removed because they were found to be highly correlated to SHR, and SST, and POT, respectively, and highly correlated variables do not give us much additional information. Therefore, DVMX, SHR, SST, POT, and RHLO were remained to achieve the highest composite probability of RI.

KD03 also found that RI instances tended to occur farther south and west than the non-RI instances. In addition, the RI instances had a more westerly component of motion and were intensifying more during the preceding 12h than the non-RI instances. Furthermore, the RI instances appeared farther from their maximum potential intensity and in regions of warmer water, higher lower-tropospheric relative humidity, lower vertical shear, and more easterly upper-tropospheric flow than the non-RI instances. Interestingly, RI was more likely to occur for systems that are in an environment where forcing from upper-level troughs or cold lows was weaker than the average of all.

1.2.2 Revised RII model (KDK10)

To employ a more sophisticated statistical method, compared to KD03, KDK10 used four more variables, one large-scale variable, and three satellite-derived variables,

- 200-hPa divergence from the 0–1000-km radius (D200),
- Percent area from 50 to 200 km covered by $\leq -30^{o}\mathrm{C}$ infrared (IR) imagery cloud-top brightness temperatures (PX30),
- Standard deviation of 50–200-km IR cloud-top brightness temperatures (SDBT), and
- Ocean heat content (OHC),

to conduct a linear discriminant analysis for RI prediction both in the Atlantic and in the eastern North Pacific based on TCs happened during 1995 and 2006. KDK10 evaluated the performance of the model in terms of probability of detection (POD) and false alarm ratios (FAR), and the model was found better than any other operational RI prediction models at that time. Meanwhile, D200, SHRD, and the PER were found to be the most

important variables in the Atlantic basin RI prediction. In contrast, PER, SDBT, and POT were found to be the most important variables in the eastern North Pacific basin.

1.2.3 Enhanced RII model (KRD15)

To improve the usefulness of the revised RII model, Kaplan et al. (2015, hereafter, KRD15) reevaluated the variables for RI with 20-55 knots intensity changes in 12 to 48 hours (seven combinations) for both the Atlantic basin and the eastern North Pacific and selected ten variables (replaced two and added two in comparison to those in KDK10). They then used the linear discriminant analysis technique to develop an enhanced SHIPS-RII. The enhanced SHIPS-RII model, along with the logistic regression model and the Bayesian classification models by Rozoff and Kossin (2011), were fed into a probabilistic model, and resulted in a better RI prediction.

Although KD03, KDK10, and KRD15 achieved certain prediction skills for the RI prediction, the test method used in those studies for variable selection is a one-by-one t-test, which is a trial-and-error process on individual factors. There are possibilities that a single variable may be insignificantly correlated to response, but multiple variables together may have a significant impact on the response's prediction skill (Trevor et al. 2009). Furthermore, only a few variables (usually less than 20) are selected for these studies, and many useful variables may be neglected. Therefore, more systematic methods are needed to conduct an exhaustive search for the most influential factors contributing to RI in a given set of factors. Efforts were made by Yang et al. (2008), and Yang et al. (2011), which employed the association rule for feature selection among the variables identified by KD03.

1.2.4 Systematic machine learning and data mining models

Association rule is an unsupervised and automatic data exploration method to explore multiple-to-one associations for discovering interesting relationships hidden in large databases (Yang et al. 2007). The strength of association rule can be measured concerning its support and confidence. Support determines how often a rule applies to a given data set, while confidence determines how frequently the rule happens (Tan 2015). Yang et al. (2007) adopted association rules with the 11 independent variables being discretized into two value ranges (High-Low) for each of the variables to predict RI. A three variables association rule (47.6% confidence, and 1.3% support) mined out has a higher RI probability than that with five variables (41% confidence, 0.7% support) identified by KD03. Yang et al. (2011) used association rule with more variables from KD03 database for the period 1997-2003. The result showed that the association rule reaches the support of 5.5% with an accuracy of no less than 70%. However, there are a large number of RI instances that do not follow the rule; a more generalized approach should be used.

Furthermore, Yang (2016, hereafter, Y16) employed WEKA (Holmes et al. 1994), a machine learning toolbox, to conduct an exhaustive and systematic examination for classification-based RI prediction with various models, subset features, and cost values for imbalance handling. Y16 split the entire dataset into a training dataset for model fitting and a test dataset for model evaluation. Although the performance of the best model in Y16 achieved a decent training result, the performance on test data was not as good. Apparently, the commonly known overfitting caused an accuracy discrepancy.

One way to improve the performance of Y16 is to tweak the so-called hyperparameters, the set of model parameters that do not change over the training process because in Y16, only the default hyperparameter setting is used. The other way is to improve the cost-effective approach used in Y16 that handles the highly imbalanced RI and non-RI instances.

So far, most RI prediction studies, including those introduced above, are conducted based on SHIPS database. However, variables in the SHIPS database are built upon human expertise in defining a relevant event based on hard and subjective thresholds. There are possibilities that those expert engineered variables in SHIPS may not be comprehensive enough, or some useful information may be ignored by the experts since the mechanism of TC intensification, and RI process is still not fully understood.

Therefore, other data sources should be employed in addition to SHIPS data to enhance the performance of the model. As one of the best reanalysis products at the moment, European Centre for Medium-Range Weather Forecasts (ECMWF) ERA-Interim reanalysis data can be a candidate. A large number of researches regarding TCs are conducted based on ECMWF ERA-Interim reanalysis data.

For example, in Wang et al. (2015), the relationship between the vertical wind shear (VWS) and the intensity change is analyzed based on ECMWF ERA-Interim reanalysis data. VWS was found negatively correlated with the intensity change, and furthermore, instead of commonly used shear between 200 and 850 hPa, the shear between 300 and 1000 hPa displays a higher negative correlation with the TC intensity change. Wang et al. (2015) also indicated that the probability for TC intensifies the

intensity and suffers RI increases when the VWS is falling and sea surface temperature (SST) is increasing. Other researches with the ERA-Interim data on TC related topics include Qian et al. (2016), Wang (2018), Astier et al. (2015), and Ferrara et al. (2017).

1.3 Proposed approach

To improve the performance of previous RI prediction researches and to identify the new additional essential variables from the prediction, this study constructs a well-tailored artificial intelligence (AI) system that uses a data filter to process the input data into attribute-relation format, adopts a customized data sampler for overcoming the imbalance, employs a very powerful state-of-the-art classifier, and tweaks the hyperparameters for optimal results. The structure of the proposed AI system is displayed in Figure 1.1. The input data could be a single data source, as well as multiple data sources, and for a single data source input, the input data will be processed to attribute-relation format by the data filter to be fed into the data sampler. If there are multiple input data sources, each data source will be processed into a separate attribute-relation table and are concatenated with each other before feeding into the data sampler. The data sampler will upsample RI (minority) instances and downsample non-RI (majority) instances accordingly, leading to a balanced augmented data set. The classifier will then classify the balanced data into RI or non-RI instances. Although the hyperparameter tuning is displayed as one component in Figure 1.1, the process could take place in multiple steps, either independently for one component, or in several components for the whole AI system.

Based on the AI system in Figure 1.1, three models, the COR-SHIPS model, the LLE-SHIPS model, and the DL-SHIPS model, are developed, and some details are elaborated below.

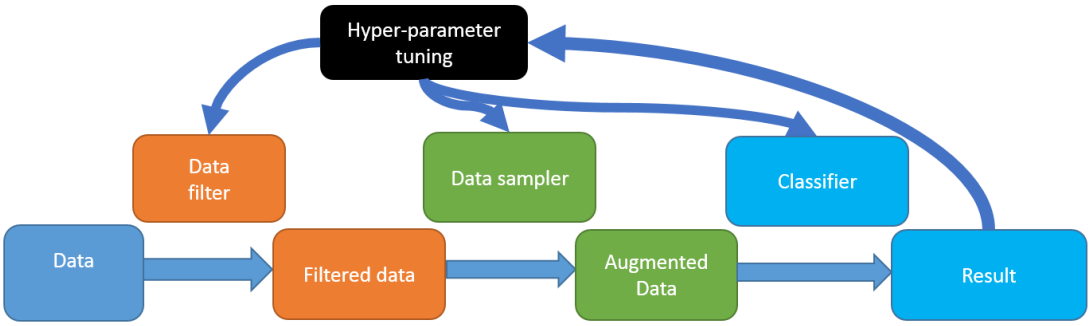


Figure 1.1: The Artificial Intelligence (AI) system structure designed in this study. One data filter is displayed in the Figure to process one data source, but if there are multiple input data sources, multiple data filters will be used, and each input data will be processed separately from each data filter into a separate attribute-relation table. All of the data filters' output is concatenated together before feeding into the data sampler.

1.3.1 COR-SHIPS model

COR-SHIPS model employs only SHIPS developmental data and is the continued work of Y16. Comparing with Y16, the COR-SHIPS model adopts a different data filter, upsamples RI instances, employs a more powerful classifier, and tunes their hyperparameters to improve the performance. Figure 1.2 displays the structure of the COR-SHIPS model, which consists of four components, SHIPS data filter, GMM-SMOTE data sampler, XGBoost classifier, and hyperparameter tuning component. The

SHIPS data filter is first used to convert the SHIPS developmental case-based data blocks to commonly used attribute-relation format and to filter the variables to generate a reduced variable set as the input for the data sampler. The GMM-SMOTE data sampler, XGBoost classifier, and hyperparameter tuning component are working the same as the corresponding component in Figure 1.1.

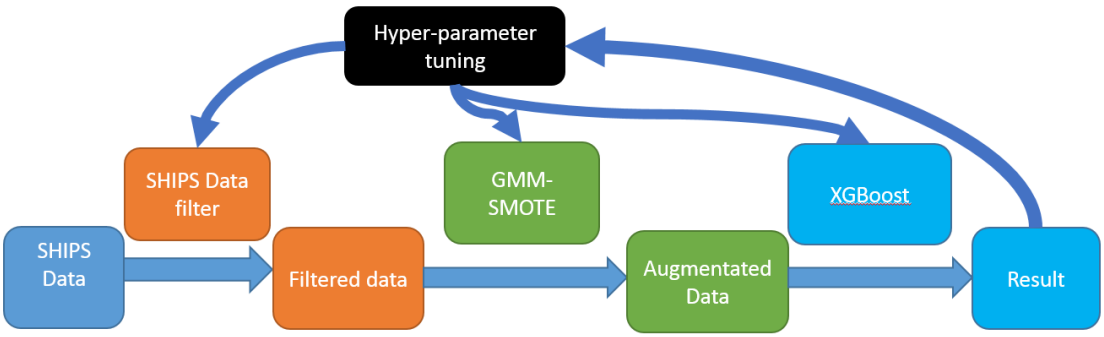


Figure 1.2: COR-SHIPS model structure.

1.3.2 LLE-SHIPS model and DL-SHIPS model

However, similar to Y16, the COR-SHIPS model still only employs the SHIPS dataset, which is largely based on expert experiences. Since the mechanism of the TC is unknown, the knowledge from the domain scientist may not be comprehensive, which may result in some important variables not being included in the SHIPS dataset. Therefore, an additional input data source, comprehensive ECMWF ERA-Interim reanalysis data, is used, and different ERA-Interim data filters are used to process the data into the attribute-relation format. Two models, the LLE-SHIPS model, and the DL-SHIPS model, are proposed based on the structure that is described in Figure 1.3 with

almost the same structure as that of the COR-SHIPS model in Figure 1.2 except two differences.

Firstly, in addition to using the SHIPS data filter, LLE-SHIPS model and DL-SHIPS model employ other ERA-Interim data filters to process ERA-Interim data. The ERA-Interim data filter in the LLE-SHIPS model is used to only value near the TC center (near center information) that is derived from the reanalysis data. Due to the limitation on LLE implementation, the DL-SHIPS model will be used to derive additional large-scale variables, up to 1,200 km from the TC centers.

Secondly, the hyperparameter tuning component of the LLE-SHIPS model and DL-SHIPS model tune of ERA-Interim data filter independently from the other components, which are similar to the COR-SHIPS model.

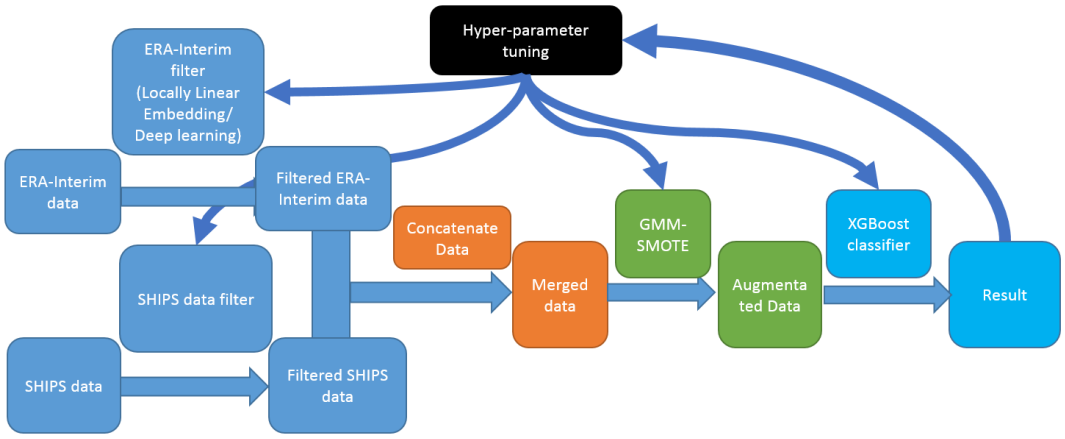


Figure 1.3: LLE-SHIPS and DL-SHIPS model structure.

In sum, this study unprecedently designs an AI system that automates data filtering, data augmentation, classification, and hyperparameter tuning for TC intensity prediction to improve the RI prediction performance and identify new variables that are critical in the prediction process. This is also one of the few attempts to explore how the state of art machine learning models perform in TC RI prediction.

The outline of the remainder of this dissertation is constructed as follows. The datasets, including but not limit to ERA-Interim and SHIPS for this study, are introduced in Chapter 2. Chapter 3 specifies data filters, and Chapter 4 describes the data sampler. The classifier and hyperparameter tuning components are discussed in Chapter 5. Chapter 6 delivers the result and discusses variable importance. Chapter 7 concludes the study and discusses future research.

CHAPTER 2 DATA

The variables used in this study originated from three datasets for the 1982 to 2017 period. The first data is SHIPS Developmental Data, the most complete dataset with known different types of the parameters related to TC intensity changes (SHIPS 2018a). Another data set is the ECMWF ERA-Interim pressure level reanalysis data, "a reanalysis of the global atmosphere covering the data-rich period since 1979 available every 6 hours over 37 different vertical pressure levels" (Berrisford et al. 2011). The last is the National Hurricane Center (NHC) best track data, which has "a comma-delimited, text format with six-hourly information on the location, maximum winds, central pressure, and (beginning in 2004) size of all known TCs and subtropical cyclones" (Landsea and Franklin 2013). To prepare for further analysis in this study, National Hurricane Center (NHC) best track data is first used to locate the center of the TC, and related spatial subsets around TC centers is cropped from ECMWF ERA-Interim pressure level reanalysis data. Then ERA-Interim variables are processed through the ERA-Interim data filter and then concatenated with the SHIPS variables processed from the SHIPS data filter.

2.1 SHIPS Developmental Data

At the moment, SHIPS Developmental Data (SHIPS 2018a), collected in ASCII text file (SHIPS 2018b), is the most complete dataset with known different types of the parameters related to TC intensity changes. Every year, instances from the previous year, and possibly new variables from other sources are added to the SHIPS Developmental Data while some old variables may be removed. The 2018 version of the SHIPS

Developmental Data used in this study had TC instances from 1982 to 2017 in the Atlantic basin.

SHIPS data consists of synoptic, climatological, persistence, geographical, satellite, experimental variables. Some of the variable examples are horizontal wind difference between 850 and 200 hPa (SHRD) [1], relative humidity between 850 and 700 hPa (RHLO), the maximum potential intensity from Kerry Emanuel equation (VMPI), previous 12-hour change intensity (BD12), the 200 hPa zonal wind (U200), the 200 hPa zonal wind with radius 0-200 km (U20C), and Reynolds SST (RSST) (DeMaria and Kaplan 1994; DeMaria and Kaplan 1999; DeMaria et al. 2005).

The values of these variables are derived from multiple types of data sources and are accumulated based on multiyear data processing. It is almost impossible to describe all of them in detail, so we describe some data source examples below (Readers are referred to the SHIPS data description file for more information (SHIPS 2018c)).

TPW (Total Precipitable Water), which is the volume of water vapor in a column that from the earth surface to the atmosphere, is a meteorological parameter used for heavy precipitation prediction. TPW is created by two satellite instruments, AMSU (Advanced Microwave Sounding Unit) on three NOAA satellites, and SSM/I (Special Sensor Microwave Imager) on three DMSP (Defense Meteorological Satellite Program) satellites (Kidder and Jones 2007), using a blending algorithm. In this study, 40 variables are TPW variables in different spatial scale related to TC center, including 21 variables in MTPW

[1] Original abbreviations used in SHIPS are used here. Readers are referred to SHIPS documentation (SHIPS 2018c) for details.

(explained in Table 2.1 and also will be discussed below), and 19 variables, PW01 to PW19, in different temporal scale of MTPW.

Infrared (IR) imagery is produced by sensing the electromagnetic radiations emitted or reflected from a target surface. IR imagery obtained from GEOS east and GEOS west has high temporal and spatial resolution and also is well known for its correlation with TC rapid intensification (Knaff et al. 2008). Variables in SHIPS created from GEOS infrared (IR) imagery are: IR00, IRXX, IRM3, and IRM1 are in different spatial and temporal scales. More details are explained in Table 2.1.

Table 2.1: List of one-time variables (SHIPS 2018a), explaining the details of each variable, and the values for each corresponding time column (Adopted from SHIPS (2018c)).

Variable Name	Variable description
HIST	Storm history variable. The number of 6 hour periods the storm max wind has been above 20, 25, …, 120 kt
IR00	Variables from GOES data (not time dependent). The 20 values in this record are as follows: TIME = 0: Time (hhmm) of the GOES image, relative to this case TIME = 6: Average GOES ch 4 brightness temp (deg C *10), r=0-200 km TIME = 12: Standard deviation of GOES BT (deg C*10), r=0-200 km TIME = 18: Same as 2) for r=100-300 km TIME = 24: Same as 3) for r=100-300 km TIME = 30: Percent area r=50-200 km of GOES ch 4 BT < -10 C TIME = 36: Same as 6 for BT < -20 C TIME = 42: Same as 6 for BT < -30 C TIME = 48: Same as 6 for BT < -40 C TIME = 54: Same as 6 for BT < -50 C TIME = 60: Same as 6 for BT < -60 C TIME = 66: max BT from 0 to 30 km radius (deg C*10) TIME = 72: avg BT from 0 to 30 km radius (deg C*10) TIME = 78: radius of max BT (km) TIME = 84: min BT from 20 to 120 km radius (deg C*10)

	TIME = 92: avg BT from 20 to 120 km radius (deg C*10) TIME = 98: radius of min BT (km) TIME = 102 to 120: Variables need for storm size estimation
IRXX	Same as IR00 above, but generated from other variables (not satellite data). These should only be used to fill in for missing IR00 if needed
IRM1	Same as IR00 but at 1.5 hours before initial time
IRM3	Same as IR00 but at three hours before initial time
PSLV	Pressure of the center of mass (hPa) of the layer where storm motion best matches environmental flow (t=0 only). Also, the information used to calculate the steering layer pressure. All fields are valid at TIME = 0, and those in the TIME = 6 to TIME = 102 columns include the following: TIME = 6 column: The observed zonal storm motion component (m/s *10) TIME = 12 column: The observed meridional storm motion component (m/s *10) TIME=18, TIME=24 columns: Same as t=6, 12 hr columns but for the 1000 to 100 hPa mass weighted deep layer environmental wind (m/s *10) t=30, t=36 columns: Same as t=6,12 columns but for the optimally weighted deep layer mean flow (m/s *10) TIME=42 column: The parameter alpha that controls the constraint on the weights from being not too “far” from the deep layer mean weights (non-dimensional, *100) TIME=48 to TIME=102 columns: The optimal vertical weights for p=100, 150, 200, 250, 300, 400, 500, 700, 850 and 1000 hPa (non-dimensional *1000)

MTPW	Total Precipitable Water (TPW) variables at t=0 from the GFS analysis. The 21 values in this record are as follows: TIME = 0: 0-200 km average TPW (mm * 10) TIME = 6: 0-200 km TPW standard deviation (mm * 10) TIME = 12: 200-400 km average TPW (mm * 10) TIME = 18: 200-400 km TPW standard deviation (mm * 10) TIME = 24: 400-600 km average TPW (mm * 10) TIME = 30: 400-600 km TPW standard deviation (mm * 10) TIME = 36: 600-800 km average TPW (mm * 10) TIME = 42: 600-800 km TPW standard deviation (mm * 10) TIME = 48: 800-1000 km average TPW (mm * 10) TIME = 54: 800-1000 km TPW standard deviation (mm * 10) TIME = 60: 0-400 km average TPW (mm * 10) TIME = 66: 0-400 km TPW standard deviation (mm * 10) TIME = 72: 0-600 km average TPW (mm * 10) TIME = 78: 0-600 km TPW standard deviation (mm * 10) TIME = 84: 0-800 km average TPW (mm * 10) TIME = 90: 0-800 km TPW standard deviation (mm * 10) TIME = 96: 0-1000 km average TPW (mm * 10) TIME = 102: 0-1000 km TPW standard deviation (mm * 10) TIME = 108: %TPW less than 45 mm, r=0 to 500 km in 90 deg azimuthal quadrant centered on up-shear direction TIME = 114: 0-500 km averaged TPW (mm * 10) in 90 deg up-shear quadrant TIME = 120: 0-500 km average TPW (mm * 10)

446

447 The NCODA system is “an oceanographic version of the multivariate optimum
448 interpolation (MVOI) technique widely used in operational atmospheric forecasting
449 systems. The ocean analysis variables in NCODA are temperature, salinity, geopotential
450 (dynamic height), and velocity” (Cummings 2005). Related variables are NSST (SST
451 from the NCODA analysis), NTMX (Max ocean temperature in the NCODA vertical
452 profile), NDFR (Depth of the lowest model level in the NCODA analysis), NTFR (Ocean
453 temperature at the lowest level in the NCODA analysis), NOHC (Ocean heat content
454 from the NCODA analysis relative to the 26 degree C isotherm), NO20 (Same as NOHC

with respect to the 20 degree C isotherm), and XNST-XO20 (Climatological values of the NCODA variables with relate to the depth of the 30, 28, …, 16 deg C isotherms).

The original format for SHIPS data is an ASCII text file, which consists of a large number of blocks, and each block involves variable names and their values from the current time up to 120 hours in a 6-hour interval. Some satellite and count variables only have one time only, and a few other variables are with values up to 12 hours before the current time.

Figure 2.1 displays an example of one data block, which has 141 lines with the line name at the end of each row with two special notation HEAD/LAST for the start/end of the block. TIME indicates the relative hour to the current time, and 9999 is filled when the value of a variable is not available. Based on the contents of the data block, all the lines can be divided into three categories:

- Special notations: HEAD, LAST, and TIME. The detailed information of HEAD, LAST, and TIME are displayed in Table 2.2.
- One-time variables: HIST, IRXX, IR00, IRM1, IRM3, PSLV, and MTPW: Those variables have only one-time values such as satellite and count variables with each TC instance, and the values on a particular column are with different meanings rather than time-dependent values. In Figure 2.1, 0 to 120 hours are corresponding to 21 values. The detailed information is displayed in Table 2.2, which also indicates that within each line, each element (column) only presents a different variable rather than providing values of variables at 6 hour interval. For example, that MTPW at TIME 6 column is 44 implies 0-200 km Total

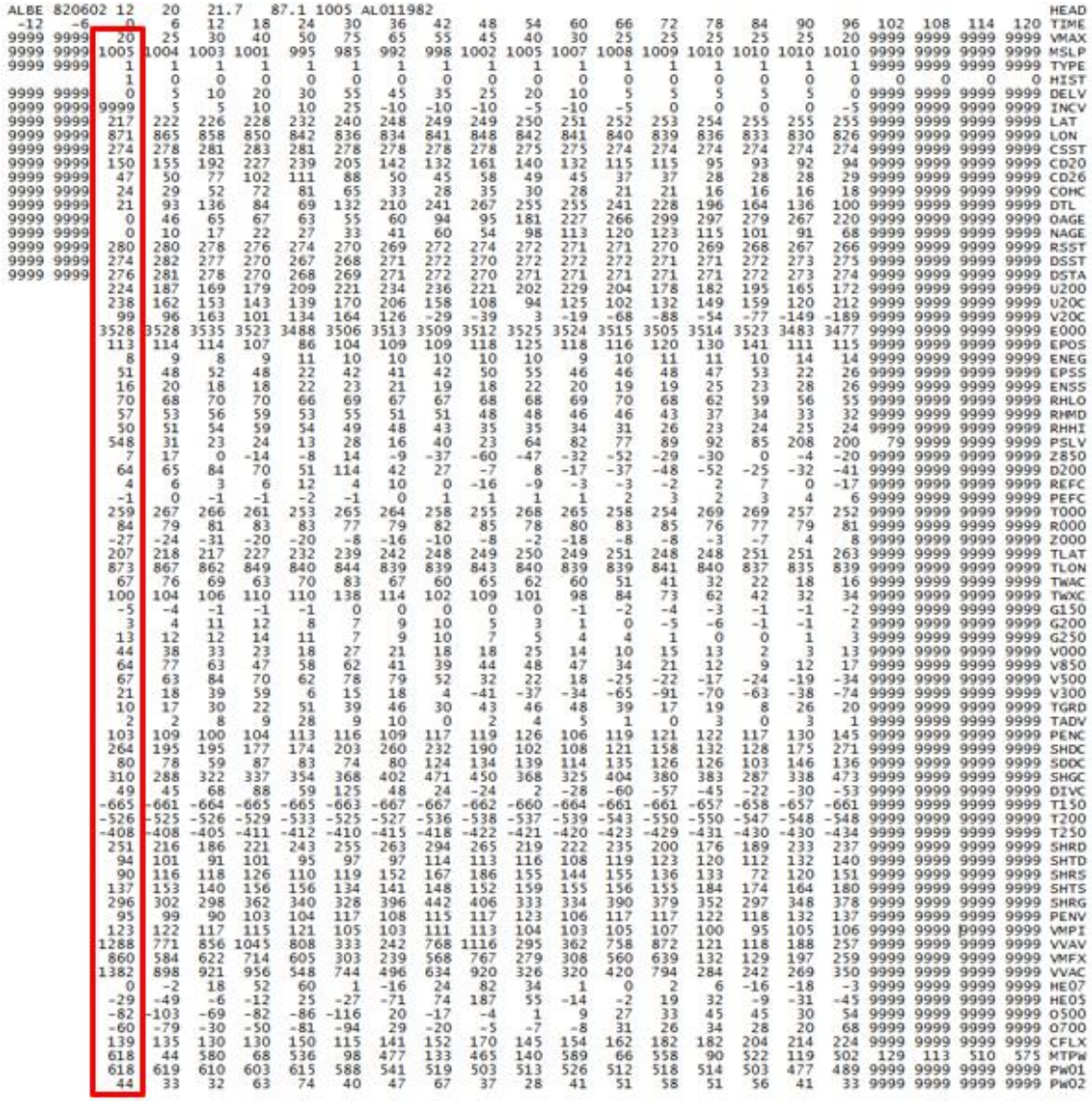
Precipitable Water standard deviation (mm * 10) is 44 based on Figure 2.1 and Table 2.1.

- Time-dependent variables: all other lines (variable names): Each line provides values of the corresponding variable at 6 hour interval at most from past 12 hours (-12 in TIME line) up to 120 hours (120 in TIME line) in the future as displayed in Figure 2.1. For example, VMAX at TIME 6 is 25 implies that 6 hours later from the moment of the block the VMAX is 25 (kt) based on Figure 2.1.

Table 2.2: List of the special notations (SHIPS 2018a), explaining the details of each variable (Adopted from SHIPS (2018c)).

Variable Name	Variable description
HEAD	Header line (1st four letters of storm name, 2-digit year, month, day, and UTC time, maximum winds, lat, lon, minimum sea level pressure, and ATCF ID number (e.g., AL011982) at t=0 of the current case)
TIME	Time away from current
LAST	The last line for this case

(a)



ALBE	820602	12	20	21.7	87.1	1005	AL011982														HEAD
-12	-6	0	6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96	102	108	114	120	TIME
9999	9999	20	25	30	40	50	75	65	55	45	40	30	25	25	25	25	25	20	9999	9999	9999	9999	VMAX
9999	9999	1005	1004	1003	1001	995	985	992	998	1002	1005	1007	1008	1009	1010	1010	1010	1010	9999	9999	9999	9999	MSLP
9999	9999	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	9999	9999	9999	9999	TYPE
		1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	HIST
9999	9999	0	5	10	20	30	55	45	35	25	20	10	5	5	5	5	5	0	9999	9999	9999	9999	DELV
9999	9999	9999	5	5	10	10	25	-10	-10	-10	-5	-10	-5	0	0	0	0	-5	9999	9999	9999	9999	INCV
9999	9999	217	222	226	228	232	240	248	249	249	250	251	252	253	254	255	255	255	9999	9999	9999	9999	LAT
9999	9999	871	865	858	850	842	836	834	841	848	842	841	840	839	836	833	830	826	9999	9999	9999	9999	LON
9999	9999	274	278	281	283	281	278	278	278	278	275	275	274	274	274	274	274	274	9999	9999	9999	9999	CSST
9999	9999	150	155	192	227	239	205	142	132	161	140	132	115	115	95	93	92	94	9999	9999	9999	9999	CD20
9999	9999	47	50	77	102	111	88	50	45	58	49	45	37	37	28	28	28	29	9999	9999	9999	9999	CD26
9999	9999	24	29	52	72	81	65	33	28	35	30	28	21	21	16	16	16	18	9999	9999	9999	9999	COHC
9999	9999	21	93	136	84	69	132	210	241	267	255	255	241	228	196	164	136	100	9999	9999	9999	9999	DTL
9999	9999	0	46	65	67	63	55	60	94	95	181	227	266	299	297	279	267	220	9999	9999	9999	9999	OAGE
9999	9999	0	10	17	22	27	33	41	60	54	98	113	120	123	115	101	91	68	9999	9999	9999	9999	NAGE
9999	9999	280	280	278	276	274	270	269	272	274	272	271	271	270	269	268	267	266	9999	9999	9999	9999	RSST
9999	9999	274	282	277	270	267	268	271	272	270	272	272	272	271	271	272	273	275	9999	9999	9999	9999	DSST
9999	9999	276	281	278	270	268	269	271	272	270	271	271	271	271	271	272	273	274	9999	9999	9999	9999	DSTA
		224	187	169	179	209	221	234	236	221	202	229	204	178	182	195	165	172	9999	9999	9999	9999	U200
		238	162	153	143	139	170	206	158	108	94	125	102	132	149	159	120	212	9999	9999	9999	9999	U20C
		99	96	163	101	134	164	126	-29	-39	3	-19	-68	-88	-54	-77	-149	-189	9999	9999	9999	9999	V20C
		3528	3528	3535	3523	3488	3506	3513	3509	3512	3525	3524	3515	3505	3514	3523	3483	3477	9999	9999	9999	9999	E000
		113	114	114	107	86	104	109	109	118	125	118	116	120	130	141	111	115	9999	9999	9999	9999	EPOS
		8	9	8	9	11	10	10	10	10	10	9	10	11	11	10	14	14	9999	9999	9999	9999	ENEG
		51	48	52	48	22	42	41	42	50	55	46	46	48	47	53	22	26	9999	9999	9999	9999	EPSS
		16	20	18	18	22	23	21	19	18	22	20	19	19	25	23	28	26	9999	9999	9999	9999	ENSS
		70	68	70	70	66	69	67	67	68	68	69	70	68	62	59	56	55	9999	9999	9999	9999	RHLO
		57	53	56	59	53	55	51	51	48	48	46	46	43	37	34	33	32	9999	9999	9999	9999	RHMD
		50	51	54	59	54	49	48	43	35	35	34	31	26	23	24	25	24	9999	9999	9999	9999	RHHI
		548	31	23	24	13	28	16	40	23	64	82	77	89	92	85	208	200	79	9999	9999	9999	PSLV
		7	17	0	-14	-8	14	-9	-37	-60	-47	-32	-52	-29	-30	0	-4	-20	9999	9999	9999	9999	Z850
		64	65	84	70	51	114	42	27	-7	8	-17	-37	-48	-52	-25	-32	-41	9999	9999	9999	9999	D200
		4	6	3	6	12	4	10	0	-16	-9	-3	-3	-2	2	7	0	-17	9999	9999	9999	9999	REFC
		-1	0	-1	-1	-2	-1	0	1	1	1	1	2	3	2	3	4	6	9999	9999	9999	9999	PEFC
		259	267	266	261	253	265	264	258	255	268	265	258	254	269	269	257	252	9999	9999	9999	9999	T000
		84	79	81	83	83	77	79	82	85	78	80	83	85	76	77	79	81	9999	9999	9999	9999	R000
		-27	-24	-31	-20	-20	-8	-16	-10	-8	-2	-18	-8	-8	-3	-7	4	8	9999	9999	9999	9999	Z000
		207	218	217	227	232	239	242	248	249	250	249	251	248	248	251	251	263	9999	9999	9999	9999	TLAT
		873	867	862	849	840	844	839	839	843	840	839	839	841	840	837	835	839	9999	9999	9999	9999	TLON
		67	76	69	63	70	83	67	60	65	62	60	51	41	32	22	18	16	9999	9999	9999	9999	TWAC
		100	104	106	110	110	138	114	102	109	101	98	84	73	62	42	32	34	9999	9999	9999	9999	TWXC
		-5	-4	-1	-1	-1	0	0	0	0	0	-1	-2	-4	-3	-1	-1	-2	9999	9999	9999	9999	G150
		3	4	11	12	8	7	9	10	5	3	1	0	-5	-6	-1	-1	2	9999	9999	9999	9999	G200
		13	12	12	14	11	7	9	10	7	5	4	4	1	0	0	1	3	9999	9999	9999	9999	G250
		44	38	33	23	18	27	21	18	18	25	14	10	15	13	2	3	13	9999	9999	9999	9999	V000
		64	77	63	47	58	62	41	39	44	48	47	34	21	12	9	12	17	9999	9999	9999	9999	V850
		67	63	84	70	62	78	79	52	32	22	18	-25	-22	-17	-24	-19	-34	9999	9999	9999	9999	V500
		21	18	39	59	6	15	18	4	-41	-37	-34	-65	-91	-70	-63	-38	-74	9999	9999	9999	9999	V300
		10	17	30	22	51	39	46	30	43	46	48	39	17	19	8	26	20	9999	9999	9999	9999	TGRD
		2	2	8	9	28	9	10	0	2	4	5	1	0	3	0	3	1	9999	9999	9999	9999	TADV
		103	109	100	104	113	116	109	117	119	126	106	119	121	122	117	130	145	9999	9999	9999	9999	PENC
		264	195	195	177	174	203	260	232	190	102	108	121	158	132	128	175	271	9999	9999	9999	9999	SHDC
		80	78	59	87	83	74	80	124	134	139	114	135	126	126	103	146	136	9999	9999	9999	9999	SDDC
		310	288	322	337	354	368	402	471	450	368	325	404	380	383	287	338	473	9999	9999	9999	9999	SHGC
		49	45	68	88	59	125	48	24	-24	2	-28	-60	-57	-45	-22	-30	-53	9999	9999	9999	9999	DIVC
		-665	-661	-664	-665	-665	-663	-667	-667	-662	-660	-664	-661	-661	-657	-658	-657	-661	9999	9999	9999	9999	T150
		-526	-525	-526	-529	-533	-525	-527	-536	-538	-537	-539	-543	-550	-550	-547	-548	-548	9999	9999	9999	9999	T200
		-408	-408	-405	-411	-412	-410	-415	-418	-422	-421	-420	-423	-429	-431	-430	-430	-434	9999	9999	9999	9999	T250
		251	216	186	221	243	255	263	294	265	219	222	235	200	176	189	233	237	9999	9999	9999	9999	SHRD
		94	101	91	101	95	97	97	114	113	116	108	119	123	120	112	132	140	9999	9999	9999	9999	SHTD
		90	116	118	126	110	119	152	167	186	155	144	155	136	133	72	120	151	9999	9999	9999	9999	SHRS
		137	153	140	156	156	134	141	148	152	159	155	156	155	184	174	164	180	9999	9999	9999	9999	SHTS
		296	302	298	362	340	328	396	442	406	333	334	390	379	352	297	348	378	9999	9999	9999	9999	SHRG
		95	99	90	103	104	117	108	115	117	123	106	117	117	122	118	132	137	9999	9999	9999	9999	PENV
		123	122	117	115	121	105	103	111	113	104	103	105	107	100	95	105	106	9999	9999	9999	9999	VMPI
		1288	771	856	1045	808	333	242	768	1116	295	362	758	872	121	118	188	257	9999	9999	9999	9999	VVAV
		860	584	622	714	605	303	239	568	767	279	308	560	639	132	129	197	259	9999	9999	9999	9999	VMFX
		1382	898	921	956	548	744	496	634	920	326	320	420	794	284	242	269	350	9999	9999	9999	9999	VVAC
		0	-2	18	52	60	1	-16	24	82	34	1	0	2	6	-16	-18	-3	9999	9999	9999	9999	HE07
		-29	-49	-6	-12	25	-27	-71	74	187	55	-14	-2	19	32	-9	-31	-45	9999	9999	9999	9999	HE05
		-82	-103	-69	-82	-86	-116	20	-17	-4	1	9	27	33	45	45	30	54	9999	9999	9999	9999	O500
		-60	-79	-30	-50	-81	-94	29	-20	-5	-7	-8	31	26	34	28	20	68	9999	9999	9999	9999	O700
		139	135	130	130	150	115	141	152	170	145	154	162	182	182	204	214	224	9999	9999	9999	9999	CFLX
		618	44	580	68	536	98	477	133	465	140	589	66	558	90	522	119	502	129	113	510	575	MTPW
		618	619	610	603	615	588	541	519	503	513	526	512	518	514	503	477	489	9999	9999	9999	9999	PW01
		44	33	32	63	74	40	47	67	37	28	41	51	58	51	56	41	33	9999	9999	9999	9999	PW02

(b)

580	572	578	562	533	541	513	499	489	502	513	501	494	478	477	452	444	9999	9999	9999	9999	PW03
68	61	68	95	102	61	52	73	69	47	76	76	77	72	78	69	57	9999	9999	9999	9999	PW04
536	521	534	529	494	493	487	493	483	489	486	491	477	459	457	439	429	9999	9999	9999	9999	PW05
98	91	88	87	94	78	75	83	82	69	86	78	75	72	91	91	95	9999	9999	9999	9999	PW06
477	477	507	512	480	483	486	481	480	485	489	483	454	436	441	420	414	9999	9999	9999	9999	PW07
133	115	103	94	87	87	88	90	91	71	77	73	91	103	96	106	111	9999	9999	9999	9999	PW08
465	459	485	484	475	490	516	479	471	476	481	454	431	421	424	402	389	9999	9999	9999	9999	PW09
140	138	131	113	97	80	72	84	97	73	72	81	101	104	111	121	117	9999	9999	9999	9999	PW10
589	583	585	571	553	552	520	504	492	505	516	504	500	486	483	458	454	9999	9999	9999	9999	PW11
66	59	63	90	103	60	53	72	63	43	70	71	74	69	74	64	57	9999	9999	9999	9999	PW12
558	547	556	548	520	519	501	498	487	496	499	496	487	471	469	447	440	9999	9999	9999	9999	PW13
90	85	83	91	102	77	68	79	74	60	80	75	75	73	85	81	82	9999	9999	9999	9999	PW14
522	516	534	532	503	503	495	491	484	491	495	490	472	456	457	435	429	9999	9999	9999	9999	PW15
119	106	95	94	98	84	78	84	82	65	79	74	84	89	91	94	97	9999	9999	9999	9999	PW16
502	496	516	514	493	499	503	486	479	485	490	477	457	443	445	423	414	9999	9999	9999	9999	PW17
129	122	113	104	99	82	76	84	88	69	77	79	93	96	100	106	106	9999	9999	9999	9999	PW18
113	83	116	439	758	328	494	700	695	418	778	746	847	848	942	885	860	9999	9999	9999	9999	PW19
510	526	549	461	411	476	448	442	431	446	413	418	411	408	389	377	376	9999	9999	9999	9999	PW20
575	564	570	555	534	535	512	503	488	501	505	499	491	479	475	452	446	9999	9999	9999	9999	PW21
0	-105	200	-122	253	38	30	23	18	14	13	-54	-94	13	-173	-103	77	67	112	115	9999	IRXX
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	IR00
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	IRM1
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	IRM3
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	PC00
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	PCM1
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	PCM3
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	RD20
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	RD26
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	RHCN
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	NSST
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	NTMX
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	NDTX
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	NDML
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	ND30
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	ND28
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	ND26
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	ND24
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	ND22
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	ND20
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	ND18
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	ND16
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	NDFR
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	NTFR
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	NOHC
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	NO20
270	281	284	284	284	281	278	279	280	278	278	277	277	276	276	276	278	9999	9999	9999	9999	XNST
271	281	285	284	284	281	278	279	280	279	278	278	277	276	276	276	278	9999	9999	9999	9999	XTMX
0	0	2	5	4	1	1	1	1	1	1	1	1	0	1	1	1	9999	9999	9999	9999	XDTX
15	15	25	38	34	18	12	14	19	14	14	13	13	12	11	11	11	9999	9999	9999	9999	XDML
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	XD30
9999	10	21	29	27	13	9999	9999	11	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	XD28
21	37	62	87	78	40	23	27	38	28	26	24	23	20	19	19	20	9999	9999	9999	9999	XD26
50	65	99	128	119	67	44	49	65	50	47	45	43	40	38	39	41	9999	9999	9999	9999	XD24
86	97	133	166	156	96	66	72	94	73	70	67	65	61	61	64	65	9999	9999	9999	9999	XD22
123	131	171	214	201	126	93	100	125	101	98	95	92	89	90	90	81	9999	9999	9999	9999	XD20
166	170	222	273	261	164	122	132	164	133	129	126	123	120	122	116	9999	9999	9999	9999	9999	XD18
240	233	287	347	333	215	161	172	213	175	169	166	163	162	161	140	9999	9999	9999	9999	9999	XD16
1183	1929	2941	2958	3000	2963	2662	3000	3000	3000	3000	3000	3000	2635	1021	387	91	9999	9999	9999	9999	XDFR
57	46	44	43	44	44	45	45	44	44	45	45	45	47	90	143	205	9999	9999	9999	9999	XTFR
7	20	39	56	50	24	13	15	21	15	14	13	12	11	10	11	12	9999	9999	9999	9999	XOHC
177	223	326	421	392	226	151	167	219	169	161	154	147	138	136	140	142	9999	9999	9999	9999	XO20
266	280	283	281	281	278	274	275	278	275	275	274	274	272	272	272	273	9999	9999	9999	9999	XDST
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	LAST

Figure 2.1: An example of data block of original SHIPS ASCII text file with 141 lines. a) the first 70 lines of the block. b) the rest lines with the first line duplicated with that in a).

Only special notations and one-time variables are discussed in detail above because all the time-dependent variable values are organized in the same format.

2.2 ECMWF ERA-Interim reanalysis data

ECMWF was founded in 1973 and is a research institute sponsored by several countries to produce numerical weather prediction to its member countries. ERA-Interim reanalysis data are generated by ECMWF every 6 hours and global wise covered with a horizontal resolution approximately 80km from its forecast numerical model to "improve on various technical aspects of reanalysis such as data selection, quality control, bias correction, and performance monitoring, each of which can have a major impact on the quality of the reanalysis products." (Dee et al. 2011). In short, the ERA-Interim reanalysis data is derived from the assimilating atmospheric model and can be regarded as the observed data. ERA-Interim reanalysis data has five data products, model level dataset, potential temperature dataset, potential velocity dataset, pressure level dataset, and surface dataset. Among them, the pressure level dataset is the most frequently used in TC researches. For example, Wang et al. (2015) adopted ERA-Interim pressure level data product to evaluate how vertical wind shear is influencing TC intensity change; Li et al. (2017) derived vorticity analysis from identifying TCs' track in Northwest Pacific Ocean Region based on ERA-Interim pressure level data product.

As the primary focus of this study is the TC in the Atlantic basin, ERA-Interim pressure level dataset, stored in a netCDF format, in the Atlantic basin is used. The Temporal and spatial coverage, the pressure levels, as well as the variable abbreviations of ERA-Interim pressure level dataset, are listed in Table 2.3, and the variables are explained in Table 2.4.

532

533 **Table 2.3: Temporal and spatial coverage of the ERA-interim pressure level data**
534 **and its available pressure levels and variables.**

Temporal Coverage	Four times daily, midnight, 6 am, noon, and 6 pm, January 1979 to two months delay of the moment
Spatial coverage	Global grid, 0.75 degree resolution
Pressure levels (37 to 1)	1, 2, 3, 5, 7, 10, 20, 30, 50, 70, 100, 125, 150, 175, 200, 225, 250, 300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 775, 800, 825, 850, 875, 900, 925, 950, 975, 1000 hPa
Variables (short names explained in Table 2.4)	cc, ciwc, clwc, d, z, w, o3, pv, r, q, t, u, v, and vo

535

536 **Table 2.4: Variable names, abbreviations, units, and description for the 14 variables**
537 **in the ERA-Interim pressure level dataset.**

Variable Name	**Abbreviation**	**Units**	**Description**
Fraction of cloud cover	cc	percentage	Horizontal fraction of the grid box covered by cloud
Cloud ice water content	ciwc	kg kg^{-1}	Grid-box mean specific cloud ice water content (mass of condensate / mass of moist air)
Cloud liquid water content	clwc	kg kg^{-1}	Grid-box mean specific cloud liquid water content (mass of condensate / mass of moist air)
Divergence	d	s^{-1}	Relative divergence
Geopotential	z	m^2 s^{-2}	At the surface: orography
Vertical velocity	w	Pa s^{-1}	Pressure vertical velocity dp/dt. In the model equations it is usually denoted by the Greek letter omega
Ozone mass mixing ratio	o3	kg kg^{-1}	Mass mixing ratio of Ozone
Potential vorticity	pv	K m^2 $kg^{-1}s^{-1}$	The ability of air to rotate in the atmosphere. "conservation equation directly ties together the dynamics and the heating" (Molinari 1989)

Relative humidity	r	percentage	Relative humidity is defined with respect to saturation of the mixed phase, i.e. with respect to saturation over ice below -23°C and with respect to saturation over water above 0°C. In the regime in between a quadratic interpolation is applied
Specific humidity	q	kg kg^{-1}	Grid box mean (mass of water vapour / mass of moist air)
Temperature	t	K	Temperature
U component of wind	u	m s^{-1}	West to east flow (eastward wind)
V component of wind	v	m s^{-1}	South to north flow (northward wind)
Vorticity (relative)	vo	s^{-1}	Measure of the rotation of air in the horizontal

As shown in Table 2.3, there are 14 variables and 37 pressure levels in the pressure level dataset, and some of these variables are the same as those used in SHIPS database, for example, vo in 850-hpa, t in 200-hPa, and v in 850-hpa to 200-hpa (Kaplan and DeMaria 2003).

Table 2.4 displays the 14 variables with their explanations. Fraction of cloud cover (cc), cloud ice water content (ciwc), and cloud liquid water content (clwc) are similar variables since cc presents the proportional of a grid box covered either by liquid cloud or ice cloud, while ciwc presents in each grid box the mass of cloud ice particles per kilogram of the total mass of dry air, water vapor, cloud liquid, cloud ice, rain, and falling snow, and clwc is almost the same as ciwc except that cloud ice particles is replaced by cloud liquid water droplets. Divergence (d) implies the horizontal divergence rate of the velocity that the air is spreading out from a point, and d is positive when the air is

spreading out and negative vice versa. Geopotential (z) indicates the amount of work to lift a unit of air from the mean sea level to a certain point against the gravity. Vertical velocity (w) indicates the speed of air moving upward or downward. Ozone mass mixing ratio (o3) indicates "the mass of ozone per kilogram of air." Potential vorticity (pv) indicates the ability of air to rotate in the atmosphere and usually is used to look for places where wind storm likely to occur. Relative humidity (r) indicates water vapor pressure as a percentage of a fixed vapor pressure value when water vapor becomes liquid water or ice. Specific humidity (q) is the same as ciwc, but the mass of cloud ice particles per kilogram is replaced with the mass of water vapor per kilogram. Temperature (t) indicates the temperature in the atmosphere. U component of wind (u) and V component of wind (v) indicates "the horizontal speed of air moving towards the east, in meters per second" and "the horizontal speed of air moving towards the north, in meters per second," respectively. u and v can be combined to calculate the speed and direction of the wind. The negative sign of u and v shows that air moving to the west and south. Relative vorticity (vo) is a clockwise (positive) or counter clockwise (negative) air spin.

2.3 NHC best track data

To determine which part of the ERA-interim data should be used, the center of TCs need to be located. The NHC best track (HURDAT2) data, available every 6 hours (midnight (UTC 0), 6 am (UTC 600), noon (UTC 1200), and 6 pm (UTC 1800)), including the time, longitude, latitude, maximum sustained wind speed of the TCs will be used. Table 2.5 is a concise version from the original NHC best track data and shows a

TC cataloged as AL011982, indicating the TC is the 1st TC in 1982 that occurred in the Atlantic Ocean and had seventeen records named ALBERTO. The first record indicates the TC was recorded for June 2nd, 1982 at 1200 UTC. The TC was centered at 21.7°N and 87.1°W with an intensity of 20 knots and minimum pressure 1005 millibars. RI occurs if and only if sustained wind speed in the next 24 hours increases 30 knots or more. Therefore, TC AL011982 undergoes RI at 1200UTC and 1800 UTC on June 2nd 1981, and at 0 UTC and 600 UTC on June 3rd 1981.

Table 2.5: A TC record in NHC best track data.

AL011982			ALBERTO	17		
Date	Time	System status	Latitude	Longitude	Maximum sustained wind speed (in knots)	Minimum Pressure (in millibars)
19820602	1200	TD	21.7N	87.1W	20	1005
19820602	1800	TD	22.2N	86.5W	25	1004
19820603	0	TD	22.6N	85.8W	30	1003
19820603	600	TS	22.8N	85.0W	40	1001
19820603	1200	TS	23.2N	84.2W	50	995
19820603	1800	HU	24.0N	83.6W	75	985
19820604	0	HU	24.8N	83.4W	65	992
19820604	600	TS	24.9N	84.1W	55	998
19820604	1200	TS	24.9N	84.8W	45	1002
19820604	1800	TS	25.0N	84.2W	40	1005
19820605	0	TD	25.1N	84.1W	30	1007
19820605	600	TD	25.2N	84.0W	25	1008
19820605	1200	TD	25.3N	83.9W	25	1009
19820605	1800	TD	25.4N	83.6W	25	1010
19820606	0	TD	25.5N	83.3W	25	1010
19820606	600	TD	25.5N	83.0W	25	1010
19820606	1200	TD	25.5N	82.6W	20	1010

NHC best track data is used to identify TC's center, and its related information from the entire ERA-Interim reanalysis data. The two datasets, with the same temporal resolution, will be processed through the ERA-Interim data filter, and details will be discussed in Chapter 3.

CHAPTER 3 DATA FILTER

As indicated in Figure 1.1, the COR-SHIPS model, LLE-SHIPS model, and DL-SHIPS model, although differ in certain parts, share the same logical structure. They share three of the four components, data sampler, classifier, and hyperparameter tuning process and differ only in the data filter. COR-SHIPS model employs only the SHIPS data filter, while the LLE-SHIPS model and DL-SHIPS model adopt extra ERA-Interim data with filters based on local linear embedding (LLE) and deep learning (DL), respectively, in addition to the SHIPS data filter.

3.1 SHIPS data filter

3.1.1 ASCII text to attribute-relation table

For the RI classification analysis by the XGBoost classifier, the input data model is attribute-relation tables commonly used for relational databases, and therefore, each SHIPS instance block should be transformed into one entry in an attribute-relation table. One sample block of the SHIPS ASCII text file is displayed in Figure 2.1, which will be converted to one entry in an attribute-relation table, as shown in Figure 3.1. During the conversion, special notation, one-time variables, and time-dependent variables are handled differently and are described below in detail. The number of variables for each category is also listed for tracking purposes only.

1. Special notations (3 lines in total: HEAD, LAST, and TIME): The HEAD line has ten elements: TC name (NAME); two-digit year (YEAR), month (MONTH), date (DATE); UTC; maximum surface wind; center latitude; center longitude; the

minimum sea level pressure; and the ATCF ID number (ATCF). Because the maximum surface wind; center latitude; center longitude; the minimum sea level pressure are included in the time dependent variables, only the rest six variables are extracted from the HEAD line: NAME, YEAR, MONTH, DATE, UTC, and ATCF. No information is retrieved from the TIME and LAST lines (result in 6 variables).

2. One-time variables (7 lines: HIST, IRXX, IR00, IRM1, IRM3, PSLV, and MTPW): As indicated in section 2.1, since these seven lines contain values for many one-time variables, and index is added to denote and to distinguish the corresponding variables. Therefore, HIST_0, HIST_1, …, HIST_20 are created using values in the HIST line. Similarly, IRXX_1, …, IRXX_20, IR00_1, …, IR00_20, IRM1_1, …, IR M1_20, IRM3_1, …, IRM3_20 (when TIME = 0, IRXX, IR00, IRM1, and IRM3 present the relative time of GOES image relative to the instance, not related to the problem in the study, and therefore, are removed), PSLV_1, …, PSLV_18 (values in 108, 114, and 120 hour are filled with 9999), MTPW_0, …, MTPW_20 are created using values in the corresponding lines (result in 140 variables).
3. Time dependent variables (the remaining 131 lines[2]): The current value of a time dependent parameter (t=0) is associated with the corresponding TC instance. The values for other times (from previous 12 hours to future 120 hours) may be used to derive other variables or simply ignored (result in 131 variables, total 277 variables).

[2] Three lines, PC00, PCM1, PCM3 were mistreated as time-dependent initially. See details in the main text.

Table 3.1: One row of the attribute-relation table converted from original SHIPS data showed in Figure 2.1. Number 1 to 277 corresponds to the 1st to the 277th columns in the attribute-relation table is added for notation only here.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
NAME	YEAR	MONTH	DATE	TIME	ATCF	VMAX	MSLP	TYPE	DELV	INCV	LAT	LON	CSST	CD20	CD26
ALBE	1982	6	2	12	AL011982	20	1005	1	0	9999	217	871	274	150	47
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
COHC	DTL	OAGE	NAGE	RSST	DSST	DSTA	U200	U20C	V20C	E000	EPOS	ENEG	EPSS	ENSS	RHLO
24	21	0	0	280	274	276	224	238	99	3528	113	8	51	16	70
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
RHMD	RHHI	Z850	D200	REFC	PEFC	T000	R000	Z000	TLAT	TLON	TWAC	TWXC	G150	G200	G250
57	50	7	64	4	-1	259	84	-27	207	873	67	100	-5	3	13
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
V000	V850	V500	V300	TGRD	TADV	PENC	SHDC	SDDC	SHGC	DIVC	T150	T200	T250	SHRD	SHTD
44	64	67	21	10	2	103	264	80	310	49	-665	-526	-408	251	94
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
SHRS	SHTS	SHRG	PENV	VMPI	VVAV	VMFX	VVAC	HE07	HE05	O500	O700	CFLX	PW01	PW02	PW03
90	137	296	95	123	1288	860	1382	0	-29	-82	-60	139	618	44	580
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
PW04	PW05	PW06	PW07	PW08	PW09	PW10	PW11	PW12	PW13	PW14	PW15	PW16	PW17	PW18	PW19
68	536	98	477	133	465	140	589	66	558	90	522	119	502	129	113
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112
PW20	PW21	PC00	PCM1	PCM3	RD20	RD26	RHCN	NSST	NTMX	NDTX	NDML	ND30	ND28	ND26	ND24
510	575	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999
113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128
ND22	ND20	ND18	ND16	NDFR	NTFR	NOHC	NO20	XNST	XTMX	XDTX	XDML	XD30	XD28	XD26	XD24
9999	9999	9999	9999	9999	9999	9999	9999	270	271	0	15	9999	9999	21	50
129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144
XD22	XD20	XD18	XD16	XDFR	XTFR	XOHC	XO20	XDST	HIST_1	HIST_2	HIST_3	HIST_4	HIST_5	HIST_6	HIST_7
86	123	166	240	1183	57	7	177	266	1	0	0	0	0	0	0
145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
HIST_8	HIST_9	HIST_10	HIST_11	HIST_12	HIST_13	HIST_14	HIST_15	HIST_16	HIST_17	HIST_18	HIST_19	HIST_20	HIST_21	PSLV_1	PSLV_2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	548	31
161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176
PSLV_3	PSLV_4	PSLV_5	PSLV_6	PSLV_7	PSLV_8	PSLV_9	PSLV_10	PSLV_11	PSLV_12	PSLV_13	PSLV_14	PSLV_15	PSLV_16	PSLV_17	PSLV_18
23	24	13	28	16	40	23	64	82	77	89	92	85	208	200	79
177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192
MTPW_1	MTPW_2	MTPW_3	MTPW_4	MTPW_5	MTPW_6	MTPW_7	MTPW_8	MTPW_9	MTPW_10	MTPW_11	MTPW_12	MTPW_13	MTPW_14	MTPW_15	MTPW_16
618	44	580	68	536	98	477	133	465	140	589	66	558	90	522	119
193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208
MTPW_17	MTPW_18	MTPW_19	MTPW_20	MTPW_21	IRXX_1	IRXX_2	IRXX_3	IRXX_4	IRXX_5	IRXX_6	IRXX_7	IRXX_8	IRXX_9	IRXX_10	IRXX_11
502	129	113	510	575	0	-105	200	-122	253	38	30	23	18	14	13
209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224
IRXX_12	IRXX_13	IRXX_14	IRXX_15	IRXX_16	IRXX_17	IRXX_18	IRXX_19	IRXX_20	IR00_1	IR00_2	IR00_3	IR00_4	IR00_5	IR00_6	IR00_7
-54	-94	13	-173	-103	77	67	112	115	9999	9999	9999	9999	9999	9999	9999
225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
IR00_8	IR00_9	IR00_10	IR00_11	IR00_12	IR00_13	IR00_14	IR00_15	IR00_16	IR00_17	IR00_18	IR00_19	IR00_20	IRM1_1	IRM1_2	IRM1_3
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256
IRM1_4	IRM1_5	IRM1_6	IRM1_7	IRM1_8	IRM1_9	IRM1_10	IRM1_11	IRM1_12	IRM1_13	IRM1_14	IRM1_15	IRM1_16	IRM1_17	IRM1_18	IRM1_19
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999
257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272
IRM1_20	IRM3_1	IRM3_2	IRM3_3	IRM3_4	IRM3_5	IRM3_6	IRM3_7	IRM3_8	IRM3_9	IRM3_10	IRM3_11	IRM3_12	IRM3_13	IRM3_14	IRM3_15
9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999
273	274	275	276	277											
IRM3_16	IRM3_17	IRM3_18	IRM3_19	IRM3_20											
9999	9999	9999	9999	9999											

In SHIPS, three lines, PC00, PCM1, PCM3, give the first nine principal components (PCs) for IR imageries at the current time, 1.5 hours before, and 3 hours before. Initially, we misinterpret the variables as time-dependent variables and keep the current values, PC00_1, PCM1_1, and PCM3_1, only. Although it is not a correct treatment for those values, fortunately, the kept values are for the 2^{nd} principal components, which are the only important components identified by previous studies (KRD15).

With those information extractions and format conversion, each block (a TC instance) is converted to a row similar to the form in Figure 3.1, with a total of 277 variables. In Figure 3.1, the first row is added to denote the column series numbers from 1 to 277 for notation purposes only. The second row indicates the names of the variables or headers, and the third row gives the values of the variables for one instance. And all values for all other instances are stacked together to constitute an attribute-relation table with all TC instances for this study.

3.1.2 Preprocessing of the SHIPS data in attribute-relation table

The raw attribute-relation table obtained above is hard to be used directly due to variable natures, i.e., irrelevant variables, heavy missing values (9999 in original SHIPS data), scaling issue, and the inter-correlation between variables. In addition, some potential variables are not available directly from the simple conversion procedure. As a result, a preprocessing is performed on the raw attribute-relation table.

3.1.2.1 Adding additional variables

Based on previous studies (e.g., DeMaria et al. 2005), intensity change is critical for rapid intensification prediction. Therefore, the previous 6-hour intensity change

(BD06) is calculated as subtracting the current intensity by the intensity 6 hours before, and BD06 for the first instance of each TC is set as missing. Previous 12-hour intensity change (BD12) and 18-hour intensity change (BD18) are calculated similarly. The first two (three) instances of each TC for BD12 (BD18) are set as missing. Since BD06 and BD12 contain the information of DELV and INCV, the latter two are removed (3 variables added and 2 removed resulting in 1 more variable, so 278 variables remained in total).

To include temporal variation of RI, annual Julian date is created to combine the information of MONTH and DATE, while MONTH, DATE, and UTC are removed. In addition, TYPE (storm type) should not have any influence of RI prediction and hence is also removed (1 variable added and 4 removed leading to 3 variables less, so 275 variables remained in total).

3.1.2.2 Variable removal

Some variables, such as the first four letters of storm name (NAME), are unique information used for tracing back the specific TC, and are unrelated to the RI prediction, which should be removed. ATCF and YEAR are also such variables and hence are removed. In addition, PSLV1 to PSLV18 represent storm motion information, where PSLV1 to PSLV8 are the storm motion component, and PSLV_9 to PSLV_18 indicates the optimal vertical weights for various pressures levels. Therefore, PSLV_9 to PSLV_18 are not related to TC rapid intensification and hence is removed (13 variables removed, so 262 variables remained in total).

IRXX should only be used when IR00 values are missing. Therefore, missing values in IR00_1, …, IR00_20, are replaced with the corresponding values in IRXX_1, …, IRXX_20 respectively and then IRXX_1, …, IRXX_20 are removed (20 variables removed, so 242 variables remained in total).

Missing values in a variable do not provide much information for that variable. The more missing values in a variable, the less information it has. Table 3.1 displays the missing value percentage in the SHIPS data (SHIPS 2018b) for variables with at least one missing filling. XD30 contains more than 95% missing values, while the next nine variables have more than 50% missing values. Other variables have less than 45% missing percentages. Since variables with more than 50% are not expected to give much information in the RI prediction, XD30, NDML, ND30, ND28, ND24, ND22, ND18, ND16, NO20, and XD28 are removed. After this removal, the remaining missing values are coded as NA; a notation can be easily handled later for sampling and classification (10 variables removed, so 232 variables remained in total).

Table 3.2: Variables with missing value and the missing percentage in SHIPS data. Variables are sorted according to the percentage. Variables without missing values are not listed.

Variable	Percentage	Variable	Percentage	Variable	Percentage
XD30	98.08%	RHCN	34.18%	XDML	5.05%
NDML	61.98%	PCM1	21.86%	XDTX	4.39%
ND30	61.84%	XD26	20.46%	BD06	4.30%
ND28	61.84%	IRM1_2 to IRM_20	14.12%	XDFR	2.41%
ND24	61.84%			CD26	2.10%
ND22	61.84%	PC00	14.06%	MSLP	1.08%

ND18	61.84%	IRM1_1	13.84%	TWAC	0.88%
ND16	61.84%	IRM3_2 to IRM2_20	12.93%	TWXC	0.88%
NO20	61.84%			DIVC	0.88%
XD28	51.78%	BD18	12.88%	COHC	0.48%
RD20	41.41%	XD24	12.87%	XNST	0.36%
RD26	41.41%	PCM3	12.84%	XTMX	0.36%
NDFR	39.57%	IRM3_1	12.57%	XTFR	0.36%
NTFR	37.62%	XD22	10.32%	XOHC	0.36%
NSST	37.31%	XD20	9.23%	XO20	0.36%
NTMX	37.31%	IR00_1 to IR00_20	9.12%	XDST	0.36%
NDTX	37.31%			DSST	0.03%
ND26	37.31%	XD18	8.96%	PSLV_2 to PSLV_8	0.01%
ND20	37.31%	XD16	8.65%		
NOHC	37.31%	BD12	8.60%		

In addition to variables with high missing value percentages, there are variables with a very high percentage of a single value, and those variables are of less value in the RI prediction. Table 3.2 shows variables with its largest percentages (greater than 50) of single values, and PSLV_8, the constraint on the weight, has more than 99.999% of instances with a single value, 40, is removed by assuming a threshold of 90% (1 variable removed, so 231 variables remained in total).

Table 3.3: Variables with higher than 50% single values in the SHIPS Data, the single values, and the percentages.

Variable	Value	Percentage
PSLV_8	40	99.99%
HIST_21	0	89.76%
HIST_20	0	88.10%
HIST_19	0	86.39%

HIST_18	0	83.84%
HIST_17	0	80.59%
HIST_16	0	78.77%
HIST_15	0	75.79%
HIST_14	0	73.81%
HIST_13	0	70.94%
HIST_12	0	66.30%
IRM3_1	-245	63.46%
PCM3	-245	63.27%
HIST_11	0	61.81%
IR00_1	15	58.89%
PC00	15	58.75%
HIST_10	0	56.41%
HIST_9	0	50.62%

705

706 The above components, data conversion from ASCII blocks to the attribute-
707 relation table, irrelevant variable removal, missing value, and single value handling, new
708 attribute creators together construct the SHIPS data filter. Through this filter, the original
709 ASCII based block SHIPS dataset (SHIPS, 2018c) is filtered into an attribute-relation
710 table for all TC instances with one TC instance as one row with 231 attributes (columns).

711 ***3.1.2.3 Rescale variables between 0 and 1.***

712 To make the data internally consistent, numerical values for all variables are
713 rescaled except for missing values (NA). All values are rescaled to between 0 and 1 using

714 $$\frac{Value_{variable} - Min_{variable}}{Max_{variable} - Min_{variable}} \quad (3.1)$$

where $\text{Value}_{\text{variable}}$ represents the value of a particular variable that needs to be standardized, $Max_{variable}$ and $Min_{variable}$ represent the maximum and minimum values of the particular variable across all instances.

3.1.3 Removal of highly correlated variables

Highly correlated input variables could influence the accuracy of the variable importance evaluation. Therefore, among highly correlated variables in the SHIPS dataset, only one variable should be kept while others being removed.

The definition of "highly correlated" depends on a predefined correlation threshold, which is related to the number of variables to be removed (or kept). This correlation threshold is one of the so-called hyperparameters.

As the first step, pairwise correlations of all the variables are calculated and compared with the correlation threshold. For each variable, its highly correlated variables (correlation higher than the threshold) are identified and placed in a group started with that variable. Therefore, there is a group that started with each of the 231 variables. All groups are sorted in descending order based on the number of variables they have, and if the length of the two groups is the same, the sorting is alphabetically based on the leading variable names. Then starting from the first group, all following groups starting with any current group members are eliminated, and then member variables are also removed from all other remaining groups to guarantee that each variable will appear only once. This process continues until the last group is reached. After that, the first variables in all remained groups are selected as filtered variables.

For example, Table 3.3 displays a correlation matrix between BD06, BD12, and BD18, and the selection threshold is defined as 0.8. BD12 has two correlated variables, BD06 and BD18, while BD06 and BD18 have only one correlated variable, BD12 each. Therefore BD12's group with length two is placed before BD06's, and BD18's of length one in the sorted list (not shown). In the removal process, groups started with BD06, and BD18 are removed because both are in the group starting BD12. If there were more groups (not this case), BD06 and BD18 should also be removed in all appearance. For this extremely simplified example, only variable BD12 is kept.

Table 3.4: Correlation matrix for pairs among BD06, BD12, and BD18, and the highly correlated group lists leading with each variable based on a 0.8 correlation threshold.

	BD06	**BD12**	**BD18**	**Highly correlated variable (>=0.8)**
BD06	1.00	0.86	0.75	BD06, BD12
BD12	0.86	1.00	0.92	BD12, BD06, BD18
BD18	0.75	0.92	1.00	BD18, BD12

The final result from the above procedure should be sensitive to the to-be-tuned hyperparameter correlation threshold. If the threshold is too low, important variables could be removed, and the accuracy of the model is reduced; if the threshold is too high, the variable importance will be evaluated inaccurately because the variable importance score is largely influenced by highly correlated variables (multicollinearity). In this study,

0.7, *0.8*, *0.9*, and *0.95* are predetermined as the searching space to tune (find the best value for) this hyperparameter. The details will be discussed in Chapter 6.

3.2 ERA-Interim data filters

The SHIPS data filter converts ASCII SHIPS data from the instance block format to an attribute-relation table and is the only data filter used in the COR-SHIPS model. To identify new features beyond SHIPS and to improve the RI prediction performance, the ERA-Interim data set is used together with SHIPS data in this study. As for the SHIPS data, this gridded pressure level data should also be filtered into the attribute-relation table format.

ERA data is filtered in two different ways to inspect the near core information and large-scale effects, respectively. As indicated in Figure 3.2, the average of the blue 3*3 grid boxes is used to present the near center information, and the local linear embedding (LLE) model is employed to filter the near center values, leading to the LLE-SHIPS model. For large-scale features, 33*33 grid boxes around the TC centers are filtered with a deep learning (DL) model, and the result is the DL-SHIPS model. Comparing with the LLE data filter, which can only process small scale near core information, DL can extract more information in the large-scale, including the smaller core information processed by the LLE model, but also has the risk of overfitting with the complicated structure. Moreover, the structure of the DL model is very complicated; therefore, it is much more difficult to evaluate the feature importance for the DL-SHIPS model than for the LLE-SHIPS model.

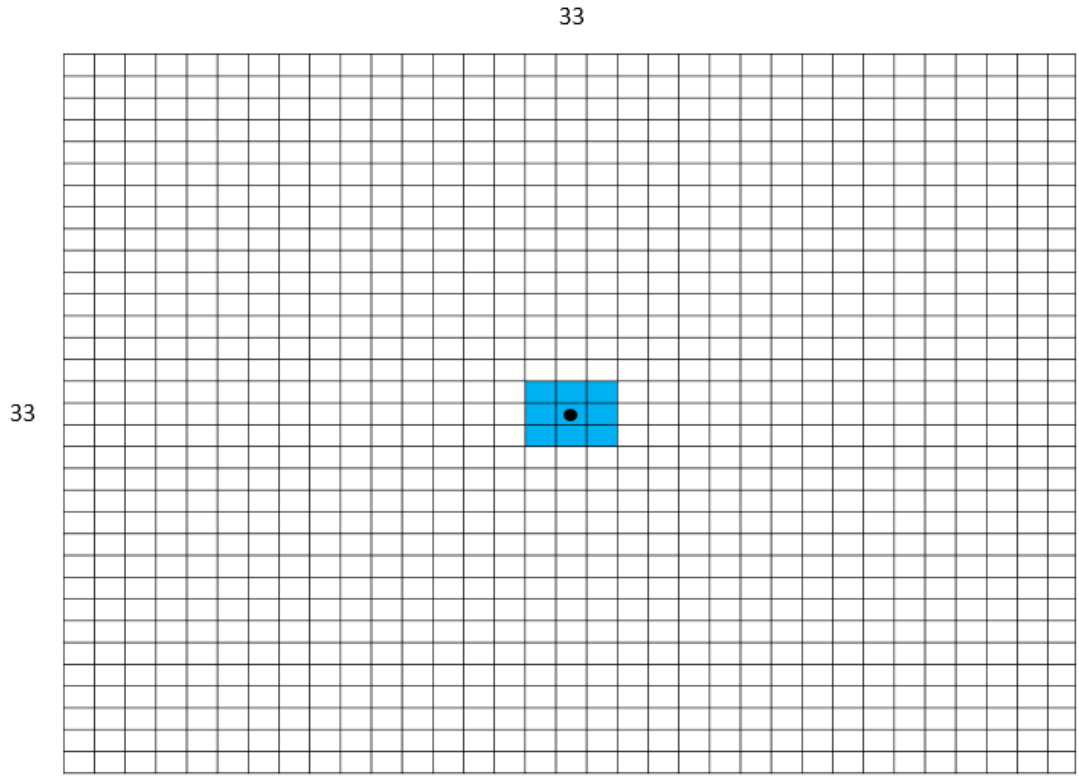


Figure 3.1: The 33*33 grid boxes centered at the grid box consisting of the center of a TC, denoted as the black dot, and the blue area presents the near core grids.

3.2.1 Local Linear Embedding (LLE) for filtering near core ERA-Interim data

The nine grid boxes around a TC center together with an approximated 240km*240km size are considered as the near center area, and values for 14 variables at 37 pressure levels are averaged over the nine boxes to have 37*14 representative values.

As well known, the RI status is not only based on the current moment but also that of the last 18 hours. Therefore, for each instance, we include data from the previous 18 hours to current, and therefore have 4*37*14 = 2072 features (variables). The features are labeled in a “time_variable_level” format. 18 hours before, 12 hours before, 6 hours

787 before, and at present are represented as NT18, NT12, NT06, and NT00. Level 1 (1000
788 hPa) to level 37 (1 hPa) are represented as l1, …, l37. For example, NT18_pv_l37
789 represents the horizontally averaged value of pv (potential vorticity) at pressure level 37
790 (1000 hPa) at 18 hours before the TC instance time.

791 For a classification problem, 2072 is a very high dimension number, and overfitting
792 cannot be avoided if all of the 2072 features are used in a model. Therefore, we need to
793 reduce the feature dimension, which can be achieved using the principal component
794 analysis (PCA), designed to reduce the feature dimensions while keeping as much
795 statistical information as possible. Pearson (1901) first came up with the idea of PCA,
796 and later, this idea was independently proposed by Hotelling (1933). PCA is applied in a
797 large number of areas such as exploratory analysis (Li and Ralph 2019), dimension
798 reduction approach (Ron 2000; Labib and Vemuri 2006), and Geostationary Operational
799 Environmental Satellite (GOES-East and GOES-West) infrared imagery variables
800 creation (Kaplan et al. 2015).

801 However, the principal components are constructed with a linear combination of the
802 original features; hence nonlinear structure between these features is missed. To break
803 this limit, kernel PCA, which uses a non-linear kernel to transfer the original feature
804 space to a kernel Hilbert space and, therefore, to account for the nonlinear structure
805 between features, was proposed (Schölkopf et al. 1998; Yang et al. 2006). Local linear
806 embedding (LLE) is a type of kernel PCA and was first introduced in Roweis and Saul
807 (2000) for dimension reduction. In traditional dimension reduction approaches such as
808 the regular kernel PCA or multidimensional scaling (MDS), when the new reduced space

809 is searched, the geometry distance between different observations are not preserved. That
810 is, far away observations in the original feature space may be mapped to their
811 neighborhood in the new reduced feature space. By contrast, LLE preserves the global
812 geometry structure from locally linear fits in the new space. In other words, low
813 dimensional representation of the high dimensional data is discovered, where these local
814 relationships are best preserved (Géron 2017).

815 LLE is used to handle the near core ERA data in this study and leads to the LLE-
816 SHIPS model. As indicated above, each instance has 4*37*14 = 2072 features. Those
817 features are rescaled to numbers between 0 and 1 using $(value - \min)/(max - min)$
818 where value presents the raw value while min and max imply the minimum and
819 maximum value within 18 hours over all 37 levels for each of the 14 variables, similar to
820 rescaling process in Equation (3.1).

821 There are two steps in LLE itself. The first step is to evaluate "how each training
822 instance linearly relates to its closest neighbors," and then in the second step, "looking for
823 a low dimensional representation of all training instances, where these local relationships
824 are best preserved" (Géron 2017).

825 Mathematically, we can elaborate the first step of LLE details with the following
826 equations:

$$\widehat{W} \sim \underset{W}{argmin} \sum_{i=1}^{m} ||x^{(i)} - \sum_{j=1}^{k} w_{i,j} x^{(j)}||^2 \tag{3.2}$$

827 subject to

$$\begin{cases} w_{i,j} = 0 \quad if\ x^{(j)}\ is\ not\ one\ of\ the\ k\ nearest\ neighbor\ of\ x^{(i)}, \\ \sum_{j=1}^{m} w_{i,j} = 1. \end{cases} \tag{3.3}$$

In the above equations, there are *m* instances in the entire dataset ($x^{(i)}, i = 1, \dots, m$), where each instance $x^{(i)}$ ($i = 1, \dots, m$) is a vector of dimension 2072. *k* is a to-be-tuned integer defining the neighborhood size. Then $w_{i,j},\ j = 1, \dots, k$ are the weights for the *k* nearest neighbors of $x^{(i)}$, and $w_{i,j}$ is summed up to 1 over all neighbors and set as 0 when $x^{(j)}$ is not the neighbor of $x^{(i)}$. $w_{i,j}$ are trained to minimize the sum of the square distance between $x^{(i)}$ and its weighted neighbors' sum, $\sum_{j=1}^{k} w_{i,j} x^{(j)}$. $\widehat{W}$ is the solution of the weight matrix W (the matrix form of $w_{i,j}$), that satisfies Equation (3.2).

In the second step, after the trained weights are calculated, instances in the entire dataset are mapped to a *d*-dimensional space ($d < 2072$, which is undefined, and its tuning will be discussed later together with *k*) while preserving the relationship between instances as much as possible. $z^{(i)}$ is the image of $x^{(i)}$ in the *d*-dimension space, $i = 1, \dots, m$. The weight $\widehat{W}$ derived from step 1 is fixed, and the sum of the squared distance between $z^{(i)}$ and its weighted neighbors should be minimized to look for $z^{(j)}$ ($j = 1, \dots, m$). That is,

$$\hat{Z} = \underset{Z}{argmin} \sum_{i=1}^{m} ||z^{(i)} - \sum_{j=1}^{k} w_{i,j} z^{(j)}||^2 \tag{3.4}$$

subject to

$$\begin{cases} \sum_{i=1}^{m} z^{(i,j)}=0 \quad j=1,\ldots,d \\ \frac{1}{m} z' z=I_d \end{cases} \tag{3.5}$$

where each the $z^{(i,j)}$ $(i = 1, \ldots, m;\ j = 1, \ldots, d)$ represents *i-th* instances in *j-th* dimension, and Z is the matrix form of $z_{i,j}$. $z^{(i)}$ is summed to 0 over all *d* dimensions, and the covariance matrix of Z be the (*d*-dimensional) I_d, where I_d indicates the identity matrix with *d*d* dimension (Roweis and Saul 2000). $\hat{Z}$ is the solution of the weight matrix Z that satisfies Equation (3.4).

Based on Ginsburg et al. (2016), LLE can be derived as a kernel PCA with kernel $K = \lambda_{max} I - (1 - \widehat{W})(1 - \widehat{W}^T)$. More technical details of PCA and Kernel PCA are given in Appendix 1.

In this work, LLE is used to reduce the original space with 2072 dimensions to the new *d*-dimensional space while preserving the maximal global geometry structure. How to define *d* is very subjective; if *d* is too large, the reduction of dimension is light, and the possibility of overfitting is not reduced much. And if *d* is too small, some important information about the original space may be lost in the new reduced space. Therefore, *d* is the hyperparameter of LLE that needs to be determined. Since SHIPS data filter outputs approximately 72 variables, 10 and 90 are defined as the lower bound and the upper bound to search for *d*.

Another hyperparameter that describes how much geometry information should be kept is the number of the nearest neighbors identified for each observation, *k*. If *k* is too low, less geometry information is kept, and the new variables lose too much information

from the original variables. If k is too high, the computational cost is too high, and overfitting cannot be avoided. Therefore, to keep the balance, 5 and 15 are defined as the lower bound and the upper bound for searching *k*.

Although *k* and *d* are independent of each other, the number of dimension (*d*) is usually larger than the number of neighbors (*k*). The search range for k and d are summarized in Table 3.4, and the tuning details will be discussed in Chapter 6.

Table 3.5: Hyperparameters for the LLE and their searching range defined by the Min(imum) and Max(imum).

Hyperparameter	Explanation	Min	Max
no_neighbors (*k*)	The number of the nearest neighbors	5	15
no_dimension (*d*)	The dimensions in the reduced space	10	90

3.2.2 Deep learning (ERA-Interim data filter for DL-SHIPS model)

The large-scale range area with 33*33 grid cells cannot be processed by LLE because even a supercomputer cannot handle the computational cost. Correspondingly, an alternative data filter based on deep learning (DL), a well-known technique for its capacity to handle a large amount of data, is used to process information in the large-scale range.

DL is a kind of Artificial Neural Network (ANN) model, which is designed to solve learning tasks by imitating the human biological neural network. The first functional ANN like model was proposed by Hodgkin and Huxley (1952), who had used non-linear

features and multiple layers to develop a model. However, the ANN model was inefficient until 1985, when backpropagation was first employed in ANN (Holyoak 1987). ANN became popular after 2006 when Hinton (2007) proposed the concept of "deep learning," an architecture with many more layers than ANN. Hinton (2007) also proved that backpropagation works efficiently in multilayer ANN learning.

Although deep learning becomes very popular and a variety of implementations were developed since then, one significant breakthrough of deep learning was Alexnet deep learning model (Krizhevsky et al. 2012), which won the first prize and achieved exciting accuracy in ImageNet 2012 challenge and marked the start of the broad implementation of deep learning. Alexnet was the first end to end deep objective classification learning system and achieved 15.3% top-5 classification error rate for the ImageNet 2012 challenge. Later work such as VGG (Simonyan and Zisserman 2014) and GoogleNet (Szegedy et al. 2015), which reach 7.3% and 6.7% top-5 error rate separately, are derived from Alexnet.

All of those works are based on Convolutional Neural Network (CNN), one of the significant components in deep learning that extracts features, i.e., variables, directly from pixel-based images. CNN is an ANN-based network that is mainly used for processing natural images with 3 RGB channels, and it significantly outperforms all other data mining techniques (Krizhevsky and Hinton 2012; Simonyan and Zisserman 2014; Szegedy et al. 2015). To be specific, CNN can be viewed as a 2D version of ANN, where the one dimensional hidden layer is replaced by multiple 2D layers.

In addition to the astonishing accuracy in image object classification, CNN is successfully applied in other areas like text classification (Karpathy et al. 2014; Lai et al. 2015), sentiment analysis (Santos and Gatti 2014), and extreme weather prediction (Liu et al. 2016; Racah et al. 2016).

Liu et al. (2016) built an Alexnet alike CNN model to classify three extreme types of weather, TCs, atmospheric rivers, and weather fronts based on the CAM5.1 historical run, ERA-interim reanalysis, 20-century reanalysis, and NCEP-NCAR reanalysis data. The overall accuracy achieves more than 88%, and the TC detection rate reaches 98%.

Although regular CNN achieves excellent accuracy in tasks like image classification, CNN cannot handle problems with temporal information involved. Tran et al. (2015) proposed a 3D CNN aiming at handling video analysis problem by adding another temporal dimension on to CNN.

The large-scale ERA-Interim dataset consists of 14 variables of 33*33 gridded data with a temporal coverage from the previous 18 hours to the current time and 37 pressure levels. A 3D CNN can be used to extract features from each individual variable in such an arrangement. The 37 pressure levels are viewed as the 37 channels similar to RGB channels of video, the gridded data as images, and the temporal coverage as image sequence of a video.

Another important structure of deep learning is the auto-encoder network, which is "a type of ANN that is trained to attempt to copy its input to its output. Internally, it has a hidden layer that describes a code used to represent the input. The network may be

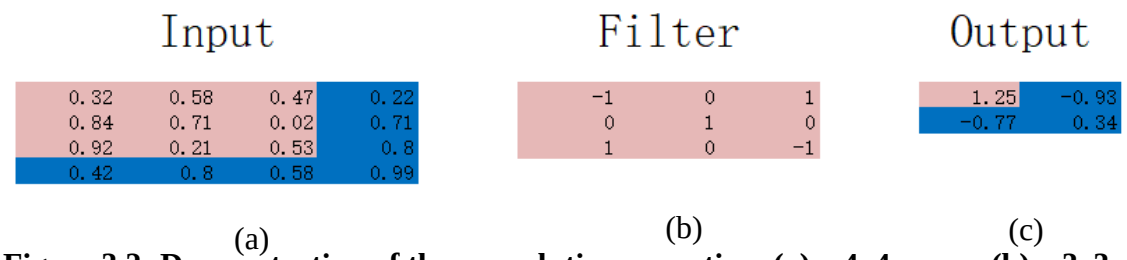
viewed as consisting of two parts: an encoder represents a feature extracted process and a decoder that produces an input reconstruction" (Wei et al. 2018). Auto-encoder is used for dimension reduction when the original data space dimension is too large and is also used for classification and prediction (Gogna and Majumdar 2019).

Racah et al. (2016) proposed an auto-encoder CNN architecture for a semi-supervised classification on the extreme weather. Since there are a large number of unlabeled extreme weather images, and to expand the training dataset, Racah et al. (2016) employed a bounding box technique to recognize the location of extreme weather, and the classification is based on those data. Although the classification performance in Racah et al. (2016) still needs improvement, it reveals that there are many promises to consider deep learning techniques in the weather community.

And in this work, the standard deep learning model is used to filter the large-scale ECMWF ERA-Interim reanalysis data, the data associated with all grids in Figure 3.2. Each instance has *4(*-18h, -12h, -6h, 0h*)*37(*pressure level*)* 33(*grid vertical dimension*)*33(*grid horizontal dimension*)* dimensions (values) (instead of 4*37*14 in LLE-SHIPS model), and the values are scaled to between 0 and 1 again, as did for LLE.

In a 2D convolutional layer, the same learnable filter is applied to each group of nearby pixels to extract features. The filter is defined as a $p*q$ (p, q are integers) size rectangle that can be convolved through the entire input array with the $m*n$ dimension. The dot product is computed between the filter weights and the input, and producing an $(m-p+1)*(n-q+1)$ output array after scanning assuming a stride of 1. Figure 3.3 displays

an example of the convolution operation. A 3*3 filter is convolved through a *4∗4* array and output a *2∗2* array with values calculated by the dot product of the sliding filter and the original data value. If the input array has more than one channel, as in a natural image with RGB channels, there will be the 3rd dimension (depth) added to the previous two-dimension filer, and the output array will still be two-dimension with value summing over the depth dimensions. Figure 3.4 shows a multi-channel example with a 4*4 image with 3 channels (Figure 3.4a). A three-dimension filter (Figure 3.4b) are designed and each is applied to the corresponding channel, and the result will be 3*2*2 outputs (Figure 3.4c). Then these 3 outputs will be simply summed up together, leading to one 2*2 output (Figure 3.4d).



(a) (b) (c)

Figure 3.2: Demonstration of the convolution operation. (a) a 4∗4 array, (b) a 3∗3 filter and its weights, and (c) the resulting output array.

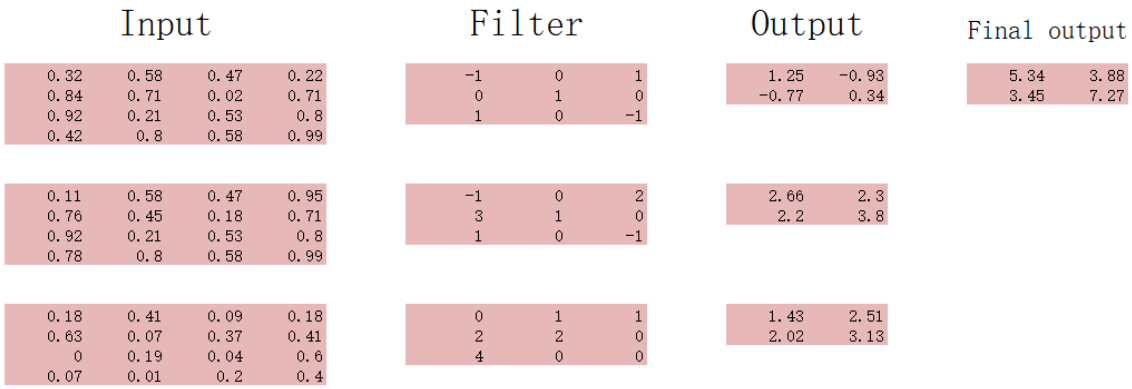


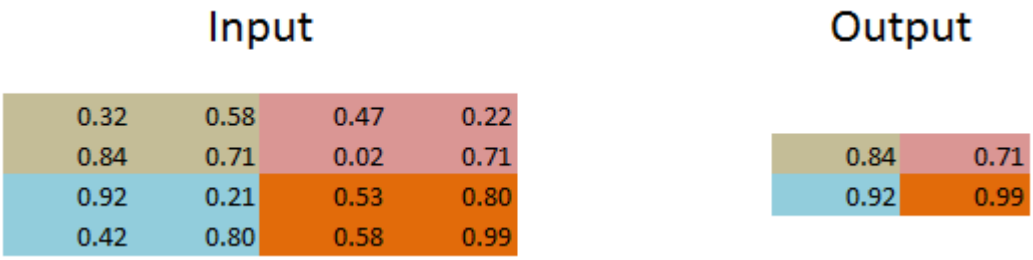
Figure 3.3: Convolution operation for input with multiple channels. (a) 3∗4∗4 arrays, (b) a three-dimension filter, (c) the output arrays after filtering the 3 channels one by one, and (d) the final output array with values being the sum of values on the depth dimension.

When the input is 3D arrays, the 3D filter and its convolve operation are the same to that of 2D except that an additional dimension is added. To be specific, a 3D $p∗q∗r$ filter is used to extract 3D information from the $m∗n∗o$ input, and result in an $(m-p+1)∗ (n-q+1)*(o-r+1)$ output array assuming stride of 1 for all dimensions. Multiple filters may be applied to the same data to extract different levels of information.

The above described convolution procedure only extracts linear information, and for obtaining nonlinear information, an activation layer is introduced after each convolutional layer. Rectified Linear Units (ReLU) is the most commonly used activation function that maps negative values to 0, and keeps the positive values, respectively. This function will not affect the size of the data arrays.

A pooling layer is usually applied after the convolution and activation transformation to reduce the input's dimension in order to avoid overfitting, and unlike convolution,

there's no overlap in pooling operations for each pooling layer. Max pooling is the most widely used pooling method. Figure 3.5 displays the 2D maximum pooling; the maximum value is taken from each block (2*2) of the original image (4*4) and generate a 2*2 pooled image. Similarly, 3D max-pooling layer uses pooling operation in 3D space, where all 3 dimensions are reduced simultaneously.



(a) (b)

Figure 3.4: Max pooling example. A 4*4 image is sampled by a 2*2 max pooling. (a) the original image, and (b) the pooled image.

There are various types of deep learning models, and the most appropriate model for converting the gridded data into features for mining purposes is the auto-encoder network. Each auto-encoder network is composed of multiple deep learning layers, which is divided into two parts: an encoder represents a feature extraction process from the input and a decoder that reconstructs the input.

With the 14*4*37*33*33 dimensional ERA-Interim data, a more efficient auto-encoder network is a 3D Conv-auto-encoder. That is an auto-encoder with a group of 3D convolutional, activation (ReLu), and pooling layers. In each 3D convolutional layer,

there are multiple 3D convolutional filters with learnable weights with an additional channel dimension on the input channels. Moreover, the 14 variables in ERA-Interim data are treated differently than usual spatial or temporal dimension, and therefore, 14 different 3D Conv-auto-encoders are adopted to handle the ERA-Interim data.

To be specific, the input of the encoder are observations with dimension of *37*4*33*33*, with pressure level (37) as its channel. There are 14 such auto-encoder networks.

The network working on a single variable is elaborated below in detail, and the dimension changes of the data are displayed in Figure 3.6.

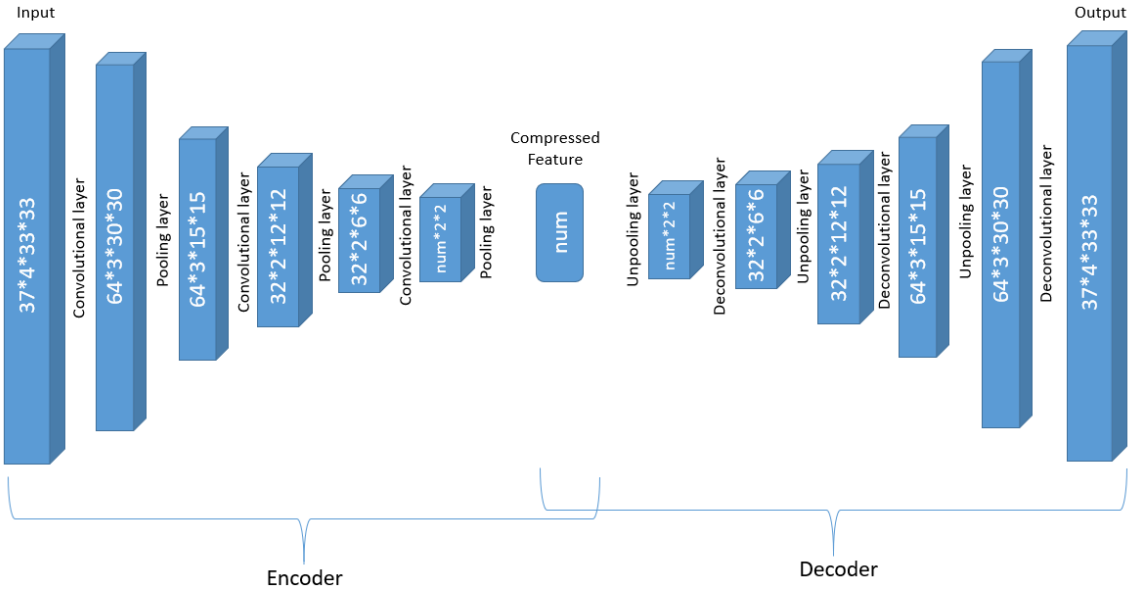


Figure 3.5: Dimension changes of the ERA data through the 3D CNN auto-encoder layers.

- The first convolution layer is with 64 different 37(channel)*2*4*4 filters and converts the 37*4*33*33 array for one variable to 64 3*30*30 arrays. In other words, a 37*2*4*4 filter is applied and the results are summed up in the channel dimension (37), and therefore the vertical pressure layer dimension number is reduced to 1. This procedure is repeated 64 times with different convolution weights. Therefore, after the first convolution layer, the original 37*4*33*33 array becomes 64 3*30*30 arrays. The activation applications after each filter in the convolution layer do not change the array size and therefore are not shown in Figure 3.6.

- A 1*2*2 pooling layer converts the 64 arrays with dimension 3*30*30 to 64 arrays with dimension 3*15*15.

- The second convolution layer has 32 different filters with dimensions 64*2*4*4, and in this layer, the new dimension due to 64 different filters in the previous layer is considered as “channels” and the filtered arrays will be summed over the channel dimension. As a result, each of the 32 filters converts the 64*3*15*15 array to 1 array with reduced dimensions 2*12*12 with the same operation as that of the first convolution layer, and finally there are 32 such arrays.

- The same 1*2*2 pooling layer is applied to the 32 2*12*12 arrays and that results in 32 2*6*6 arrays.

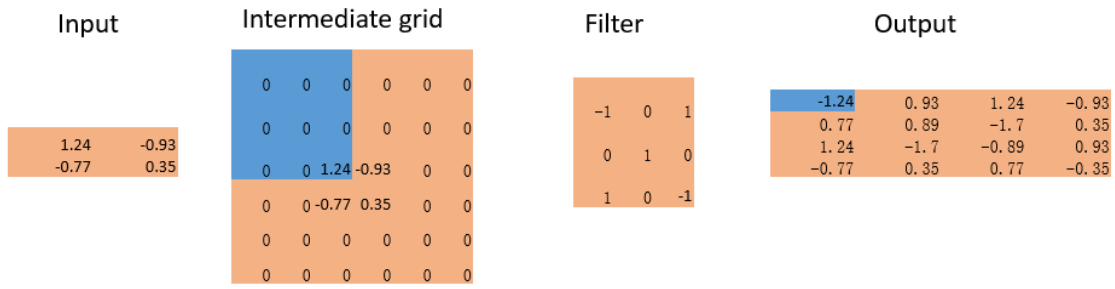
- Similar to the previous two convolutional layers, the third convolution layer has num different convolution filters 32*2*5*5, and the dimension of 32 is treated as

channel again. The result after this filtering process is num arrays of dimension 1*2*2, where the num is a to be determined hyperparameter.

- The 1*2*2 pooling layers will finally compress the arrays into num scalar features.

The decoder is the reverse of the encoder by using the deconvolutional and unpooling layers in DeConvNet network (Zeiler et al. 2010, 2011, 2013) to reconstruct the convolutional networks, i.e., reverse the convolution and the pooling operations. In deconvolutional, 0s are padded to the neighbor of the input (output of the corresponding convolution operation) to generate an intermediate grid, and a learnable filter is used to convolve through the intermediate grid to generate the output, which is the reconstruction of the convolutional input. The learnable filter in deconvolution is updated in the same way as the learnable filter in convolution.

An example is displayed in Figure 3.7, which is the deconvolution operation to reverse the convolution operation described in Figure 3.3. To reverse the 2*2 image to the 4*4 image, the original 2*2 image (a) is padded with 2 rows and 2 columns 0 around each pixel to generate the intermediate grid (b), which is then convolved through the filter (c), and result in the output (d), which is the upsampled result with regard to (a).



(a) (b) (c) (d)

Figure 3.6: The deconvolution operation, reverse of operations shown in Figure 3.3. (a) a 2*2 array, (b) padded 0 to (a), (c) the filter, and (d) resulting output array after filtering.

To reconstruct the Max pooling operation, the location of the feature map that has the maximum value (location of the passed value through Max pooling) is recorded in a switch during the corresponding Max pooling operation. Then the input of the Unpooling is upsampled where the maximum value is put to the saved position in the switch, and 0 is put into everywhere else. An example is displayed in Figure 3.8, which is used to reconstruct the output of Figure 3.5. ReLu function in the convolutional network is the same as in the DeConvNet network.

(a) (b)

Figure 3.7: An unpooling example. A 2*2 image is upsampled by a unpooling process to a 4*4 image. The position of the valid value (nonzero) in each 2*2 sub-image is based on the location of maximum value during pooling (see Figure 3.5 for details). (a) the original image, and (b) the unpooled image.

The structure of DeConvNet network is displayed on the left in Figure 3.9, which is used to reconstruct the corresponding CNN.

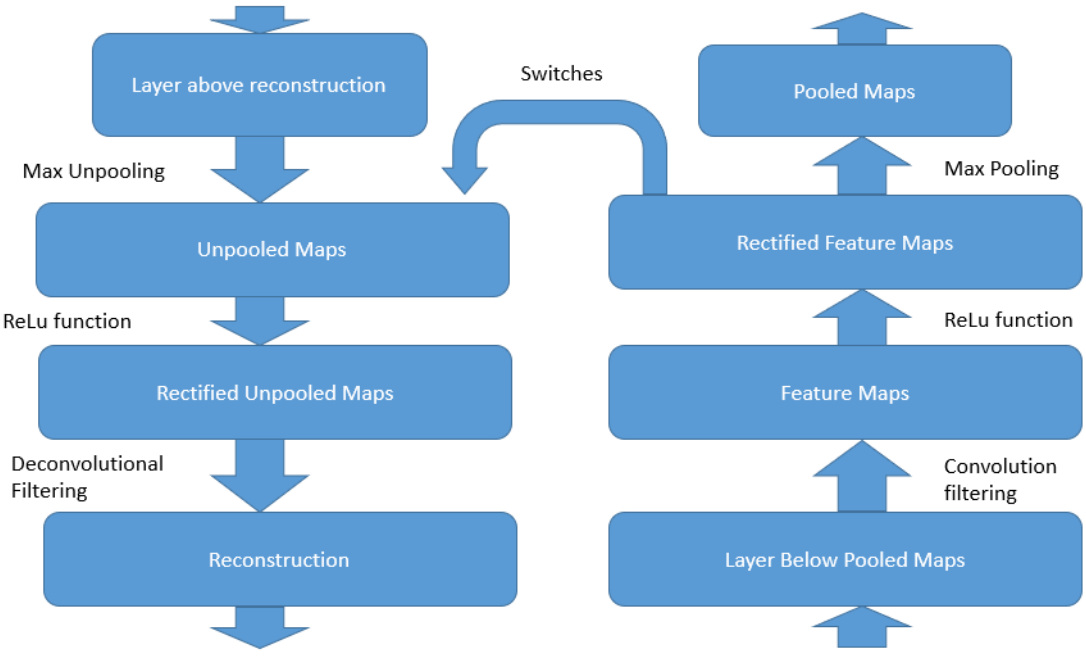


Figure 3.8: The CNN on the right that first runs through the input to the output (from bottom to top), and the position of the Max Pooling pixel is saved as a switch that will be used later for the unpooling operation on the left. On the left, the structure of the corresponding DeConvNet network consisting of one unpooling layer, one ReLu function (the same as in CNN), and one deconvolution layer to reconstruct the CNN on the right based on its output (top to bottom) (Zeiler et al. 2011).

The network is trained through the backpropagation, where the mean square error (Trevor et al. 2009) is used as the loss function, and Adam optimizer (Ruder 2016) is used as the optimizer to update the filter weights through backpropagation.

14 separate networks with the same structure displayed in Figure 3.6 are trained separately for 14 different ERA-Interim variables as shown in Figure 3.10. The compressed features from each of the networks are merged with filtered SHIPS variables and together used as the input of the GMM-SMOTE.

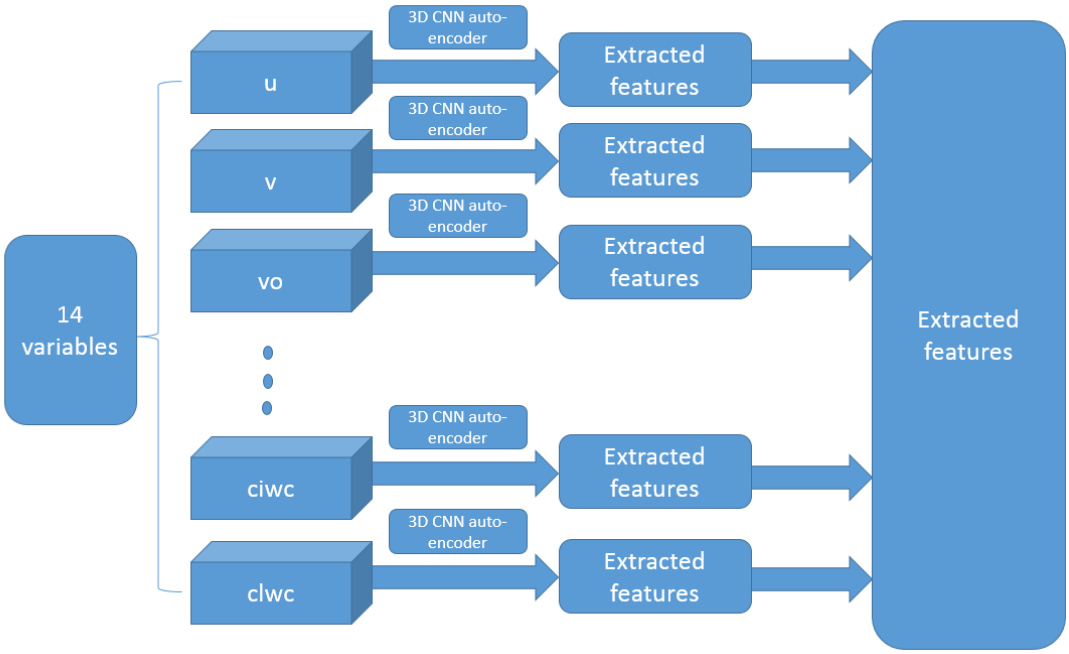


Figure 3.9: Combined deep learning filters for the 14 variables in ERA-Interim data.

CHAPTER 4 GMM-SMOTE SAMPLER

In a binary classification (prediction) problem such as the RI vs. non-RI prediction, we use the machine learning (ML) model to look for a decision boundary in the feature space (Friedl and Brodley 1997) to separate RI and non-RI instances, and the prediction is made for any new instance based on its location in the feature space.

The decision boundary in the feature space will be highly skewed if the data are with highly imbalanced class samples. Unfortunately, the RI and non-RI instances are highly imbalanced because only about 5% of TC instances undergo RI process. Generally, three types of techniques are used to handle the imbalanced data problem, algorithm level approach, cost-sensitive approach, and resampling data approach (Last et al. 2017).

The algorithm level approach aims at modifying ML algorithm that applies to regular balanced data to cope with imbalanced data to correct the skewed decision boundary. Such techniques, including changing the decision threshold and training a separate model (Anyfantis et al. 2006; Chawla et al. 2004; Galar et al. 2012). Cost-sensitive approach assigns different costs for incorrectly classifying different classes in the ML model to correct the skewed decision boundary. In other words, cost-sensitive approach assigns a lower cost for misclassifying majority class and a higher cost for misclassifying minority class (Galar et al. 2012; Castro et al. 2013; López et al. 2015). Instead of making modifications of the ML algorithm structure, the resampling data

approach resamples the imbalanced dataset to create a balanced one to decrease the effect of the skewed distribution in the ML model's learning process (Krawczyk et al. 2014).

At present, resampling is the most widely used approach to overcome the imbalanced data problem. The resampling approach falls into three categories: upsampling, downsampling, and the hybrid method (Last et al. 2017). The upsampling approach upsamples the data by duplicating observations in minority class until the number of observations in minority class matches that of the majority class, and the downsampling removes additional observations in the majority class (Japkowicz 2000). However, simply upsampling or downsampling does not significantly improve the minority class prediction accuracy because they do not fortify the decision boundary. The hybrid method combines both of them by generating new instances different from existing ones for the minority class and removing the majority class instances simultaneously. As an approach that combines upsampling and downsampling approaches to improve their drawbacks, the hybrid method should be used. Among all the hybrid resampling approaches, Synthetic Minority Over-sampling Technique (SMOTE) has been widely employed by researchers and scientists to solve the real-world problem and academy problem due to its simplicity and its advantages to random sampling (Shaiba and Hahsler 2016). SMOTE was proposed by Chawla et al. (2002) to handle the imbalanced dataset, which upsamples minority classes by constructing "synthetic" examples rather than upsampling with replications and outperforms upsampling and downsampling alone (Akbani et al. 2004; Batista et al. 2004; Liu et al. 2006).

The decision boundary that separates RI and non-RI instances in SMOTE may not be enforced since instances in minority class far from the decision boundary have the same probability of being selected as those closed ones. In addition, SMOTE may further amplify the noise present in the data (Nguyen et al. 2011). When all the instances in the minority class have an equal probability of being selected, those noise observations may be amplified and hence decrease the accuracy of the model (Bunkhumpornpat et al. 2009).

To decrease the influence of the noise observations, Han et al. (2005) proposed two approaches, SMOTE1 and SMOTE2, to improve SMOTE by splitting minority class instances into three groups, i.e., noise, safe, and danger using a k-nearest neighbor approach. An instance is regarded as noise if all its neighbors are belonging to the majority class and as safe if more than half of the neighbors are belonging to the minority class. Otherwise, that instance is regarded as danger. Noise instances are useless because they do not provide information about the minority class, and so do safe instances, in that no matter what classification model is used, they are less likely to be misclassified. Therefore, augmenting danger instances could be the most efficient way to increase classification accuracy. Therefore, approaches proposed by Han et al. (2005) only augment danger minority instances.

To avoid the influence of the extreme values in the sampling process, and to make the resample process more efficient (Jo and Japkowicz 2004), Song et al. (2016) proposed a bi-directional sampling approach, where minority and majority classes are separately clustered using K-Means. The majority class is downsampled by only selecting

instances near the cluster center, and the minority instances are upsampled by SMOTE using instances in the same cluster. The downsampling and upsampling processes are replicated multiple times until instances in majority class and minority class are balanced. Last et al. (2017) proposed K-means SMOTE by first clustering the entire population into different clusters. Then only clusters with more than a certain ratio of minority class instances are selected, and each cluster is assigned with a weight equals to the number of minority class elements divided by the sum of their distance to the center of that cluster. Then minority instances are augmented the number of times proportional to their weight.

However, K-means is not working efficiently on complex geometrical shaped data, especially in a high dimensional space. Furthermore, K-means SMOTE does not handle missing values in attributes. Finally, clustering is an unsupervised approach that the selection of the number of clusters is very subjective. If too few clusters are specified, underfitting may occur - to cluster apparently different instances into one cluster hence unable to identify the difference. If too many clusters are specified, overfitting may occur - instances that have a similar property could be clustered into different clusters.

To better fit SMOTE based approach to the high-dimensional data, here, the Gaussian Mixture Model (GMM) with a weighted Euclidean distance is used for clustering (Friedl and Brodley 1997). As a type of model-based clustering approach, GMM has been used in a large number of areas, such as speech recognition (Reynolds et al. 2000), and feature extraction (Torres-Carrasquillo et al. 2002).

However, the number of clusters (M) in GMM should be defined before fitting the model to the data. Bayesian Information Criterion (BIC) (Volinsky and Raftery 2000) is a

statistic calculated for each clustered dataset based on the likelihood to identify enough clustering information while avoiding overfitting as much as possible. BIC can be used to help select the best number of clusters, which is defined as n_cluster, a to be tuned hyperparameter in GMM. The search space for n_cluster is defined as 2 to 10, which implies BIC will be calculated for each n_cluster starting from 2 iteratively, and the process stops when BIC stops decreasing for two continuous iterations or n_cluster equals 10 is reached.

GMM is an unsupervised approach that assumes each observation in the population can be represented as a weighted sum of several (the number of pre-determined clusters) Gaussian distributions, and the weights are summed to 1. Each cluster corresponds to one Gaussian distribution, and observation will be assigned to the cluster with the highest weight (Fraley and Raftery 1998), which is unlike clustering approaches such as K-means, assigning each observation to a different cluster directly.

To be specific, assume x is an observation in population X with D feature dimensions, M is the number of Gaussian distribution (clusters) within X, w_i is the mixed weight for each component and sum to *1*, and $g(x|u_i, \Sigma_i)$ is the *i-th* Gaussian distribution with mean u_i and standard deviation σ_i. u_i and σ_i is calculated by expectation-maximization (E-M) algorithm, which is a likelihood based approach that starts from some initial estimates and stops until convergence arrives (Dempster et al. 1977).

$$p(x|\lambda) = \sum_{i=1}^{M} w_i g(x|u_i, \Sigma_i) \tag{4.1}$$

$$g(x|u_i, \sigma_i) = \frac{1}{2\pi^{1/2}|\sigma_i|^{1/2}} \exp\{-\frac{1}{2}(x - u_i)^2 \sigma_i^{-1}\} \tag{4.2}$$

The GMM clustering and the SMOTE sampling processes is combined in this study as the data sampler to resample the unbalanced RI dataset, and the data sampler employs the BIC to determine the number of clusters for GMM from a predefined range (searching space) between 2 and 10; then GMM is used to cluster all the instances. Every cluster is defined as safe, noise, or dangerous based on the instance imbalance rate (IIR), i.e., the ratio of the minority instances in the cluster divided by the ratio for the entire population. For example, if there are 3% minority instances in a cluster and 5% minority instance in the population, IIR = 3%/5% = 0.6 for this cluster. When a cluster is composed of mainly majority instances, the classification of all instances would be majority class no matter the actual instance is the majority or not. Those clusters cannot make any contribution to improve the classification accuracy and are termed as noise clusters. Similarly, a cluster is defined as safe when its minority instances are dominant in the cluster and are less possible to be misclassified. In this study, 0.2 (5) of IIR value is set as the threshold, and any clusters with IIR <=0.2 (>=5) are termed noise (safe). Otherwise (0.2<IIR<5), the classification of the minority is difficult, and the cluster is termed dangerous. Similar to Last et al. (2017), instances can also be identified as safe, noise, or danger based on the number of minority instances in their m_neighbors neighbors, but slightly different here, an instance is termed as noise (safe) if none (more than half) of its neighbors is in the minority class; otherwise, as danger. Only danger instances in dangerous clusters are upsampled with SMOTE following

$$u_{new} = u_c + w * (u_n - u_c), \quad w \sim U(0,1)$$
$$if\ u_n\ is\ minority\ class;\ otherwise\ w \sim U(0,0.5) \tag{4.3}$$

where u_c presents the selected minority class instance that needs to be augmented; u_n presents the randomly selected neighbor of u_c using k nearest neighbor in the same cluster with k= k_neighbors, another to-be-determined hyperparameter; $U(0,1)$ and $U(0,0.5)$ present random numbers with a uniform distribution between 0 and 1, and 0 and 0.5, respectively. Because the number of majority instances is much larger than that of the minority instance, the increase of m_neighbors will result in the increase of the number of majority neighbors for an instance, leading to the increased possibility of the instance is classified as noise; hence the instance is less likely to be upsampled. Therefore, the number of instances that are upsampled are fewer, and the variety of the upsampled instances is decreased. Similarly, smaller k_neighbors will lead to a smaller variety because fewer neighbors will be selected for the upsampling. Smaller (larger) variety represents less (more) coverage in the feature space and would more likely result in an underfitted (overfitted) model, or a conservative (aggressive) model. Therefore, large (small) m_neighbors and small (large) k_neighbors will lead to conservative (aggressive) models.

Each danger instance in dangerous clusters is augmented N_a times using equation (4.3) where N_a is defined as the integer part of

$$0.75 * \frac{\text{The number of majority instance in the population}}{\text{The total number of minority instances need to be augmented}} - 1,$$

which makes the final number of minority instances are approximately 75% of the majority instance number, assuming most of the minority instances will be augmented. Then, as the final step of the resampling, 25% instances in the majority class are

randomly removed (downsampled) to make the majority class and minority class have similar numbers of instances (Song et al. 2016; Last et al. 2017). Clusters with the number of instances less than the maximum of m_neighbors and k_neighbors will be removed.

The GMM-SMOTE approach described above will be used to augment the output data from the data filter of the COR-SHIPS model, LLE-SHIPS model, and DL-SHIPS model. There are 3 hyperparameters that need to be tuned, as listed in Table 4.1. To avoid overfitting and underfitting, i.e., not to select too large or too small k_neighbors, and m_neighbors, the search space is defined as 3 to 14, and 3 to 10, respectively. Furthermore, n_cluster should guarantee that each cluster has at least k_neighbors+1 instances. Therefore, the search space for n_cluster is defined as 2 to 10. A cluster will be removed if the number of its total instances is less than k_neighbors or m_neighbors.

Table 4.1: Hyperparameters and their searching space in GMM-SMOTE sampling process.

Hyperparameter	Component	Explanation	Minimum	Maximum	Initial value
n_cluster	GMM-SMOTE	The number of clusters in the Gaussian Mixture Model function	2	10	1
m_neighbors	GMM-SMOTE	The number of nearest neighbors used to determine if a minority sample is in danger	3	10	10
k_neighbors	GMM-SMOTE	The number of nearest neighbors used to construct synthetic samples	3	14	5

CHAPTER 5 XGBOOST CLASSIFIER AND HYPERPARAMETER TUNING PROCESS

5.1 XGBoost classifier

The classification and regression tree model (CART) (Friedl and Brodley 1997) is one of the most popular classifications and prediction models in the machine learning community that is capable of capturing the linear and nonlinear relationship between predictors. However, a single CART usually is a weak classifier, only slightly better than random guessing. Therefore, a boosting approach is commonly used, which trains a set of weak classifiers to enhance the classification performance, with new classifiers being trained to correct the mistake made by the previous classifiers. The performance of the boosting model with a large set of weak classifiers usually outperforms the single strong classifier.

State of the art boosting tree model is the Gradient Boosting Tree (GBT) (Trevor et al. 2009), which trains a series of weak CART classifiers iteratively. Each classifier is constructed on the remaining errors of previous classifiers, and new classifiers are trained to correct the mistake made by the previous classifiers. The output of each classifier is a leaf level score, as shown in Figure 5.1, as an example for RI prediction. An instance with attributes A and B lands at leaf A1 of Tree 1 based on the A value, and leaf B1 of Tree 2 based on the B value. The raw classification score for being a RI instance is a weighted sum, w_1*S1+w_2*S3, with w_1 and w_2 being the weights for Tree 1 and Tree 2, respectively. The raw score is rescaled to a value between 0 and 1 using a sigmoid

function (Trevor et al. 2009). In the binary classification in this study, where 0 and 1 are used to encode non-RI and RI classes, if the rescaled score is less than a decision threshold, which is a to-be-tuned hyperparameter, and preselected as 0.5, the instance is predicted as non-RI. Otherwise, the instance is predicted as RI. A greedy algorithm like GBT may generate too many weak classifiers fitting in the residuals that the total model can be easily overfitted, and a regularized distributed GBT, i.e., XGBoost, is designed to control the overfitting (Chen and Guestrin 2016).

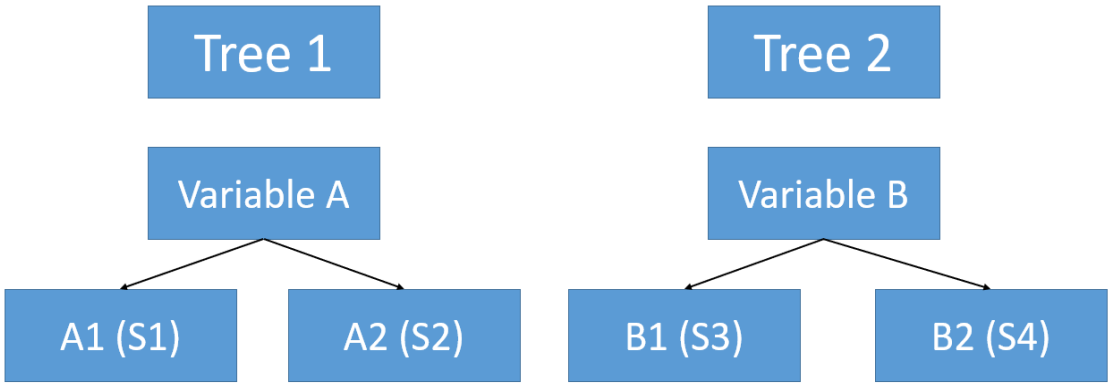


Figure 5.1. CART sample. Two regression trees 1 and 2, split based on values of variable A and variable B, respectively. A1, A2, B1 and B2 are the leave names and S1, S2, S3, and S4 are the corresponding classification scores.

XGBoost generates weaker classifiers iteratively by minimizing an objective function, consisting a loss function and a regularization function. The loss function is based on the errors between the predicted classes and the ground truth classes for all instances. The regularization function is a function of classifiers, and its purpose is to control the overfitting of the final classification. In short, the strategy in XGBoost is to

have the best prediction (minimum loss function) with the regularized complexity of the tree structure (to avoid overfitting).

The regularization constraints can be roughly divided into two categories. The first category is on the overall structure outside individual classifier, which includes shrinkage ratio (shrinkage), the number of classifiers (n_estimator), subsample ratio (subsample), and features ratio (colsample). The second category is on the individual CART (classifier) level, which includes L1 regularization (reg_alpha) and L2 regularization (reg_lambda), minimum loss reduction required to make a split (Split criteria, aka, gamma) and the minimum sum of instance weight in a split (min_child_weight), and the maximum depth of the CART (max_depth).

In the first category, since the subsequent classifiers are iteratively fitted into the remained error from the previous classifiers, the subsequent classifiers contribute less and less as boosted trees go deeper. Therefore, similar to gradient descent algorithm with decreasing steps for better approximation (Trevor et al. 2009), we decrease the contribution of even weaker classifiers with a rate of shrinkage, a hyperparameter within (0,1]. Empirically XGBoost was found to perform best with the shrinkage around 0.1, and the search space is defined as 0 to 0.3 here. While more classifiers would result in better accuracy, too many classifiers will result in overfitting. Therefore, the number of classifiers, n_estimator, is limited in [100, 2000] and will be searched in that range to avoid underfitting and overfitting simultaneously. In addition to the classifier number, overfitting can also be reduced by using only a reduced set of datasets and variables (Trevor et al. 2009). Subsample ratio (subsample) and features ratio (colsample) are used

to control the sizes of randomly selected reduced datasets and feature sets, and the search spaces are defined as 0.5 (50% instances) to 1 and 0.4 (40% features) to 1, respectively. In total, four constrains: shrinkage, n_estimator, subsample, and colsample are adopted as hyperparameters for overall constrains in the classification process.

In the second category, the concern is the same, to avoid overfitting by similar strategies but on individual trees. Although the number of features is reduced by colsample, that process is random. The number of the features are further controlled by L1 regularization and L2 regularization based on their importance, where L1 regularization (reg_alpha) and L2 regularization (reg_lambda) are similar to Ridge and Lasso (Friedl and Brodley 1997) in linear regression but apply on CART to reduce the impact of less-predictive features. The search spaces of L1 and L2 regularization are specified as 0 to 20, and 0.1 to 20, respectively. Another tree-level constraint is to limit tree growth. This can be achieved by setting the minimum loss reduction required to make a split (gamma) and a minimum sum of instance weight in a split (min_child_weight), and 0 to 10, and 0.5 to 5 are defined as their search space, respectively. Finally, the maximum depth (max_depth) is used to control the depth of a CART and is searched in the space from 3 to 10.

The last to-be-tuned hyperparameter is the decision threshold. If the summed score output from the XGboost is above the threshold, an instance is classified as RI. Tentatively, the decision threshold is preselected as 0.5, and will be tuned.

Details of all hyperparameters are specified in Table 5.1. For a short note, lower m_neighbors, k_neighbors, shrinkage, n_estimators, subsample, colsample, max_depth,

and higher reg_alpha, reg_lambda, gamma, min_child_weight lead to a more conservative model. n_cluster and decision threshold will not influence the conservativeness of the model. More details of XGBoost will be elaborated in Chapter 6.

Table 5.1: Hyperparameters, their searching space defined by the minimum and maximum, and the initial values in GMM-SMOTE sampling process and XGBoost classifier.

Hyperparameter	Component	Explanation	Min	Max	Initial value
shrinkage	XGBoost	Shrinkage ratio for each feature	0	0.3	0.1
n_estimator	XGBoost	The number of CART to grow	100	2000	100
subsample	XGBoost	Subsample ratio of the training instances	0.5	1	1
colsample	XGBoost	Subsample ratio of columns for creating each classifier	0.4	1	1
reg_alpha	XGBoost	L1 regularization term on weights	0	20	0
reg_lambda	XGBoost	L2 regularization term on weights	0.1	20	1
gamma	XGBoost	Minimum loss reduction required to make a further partition on a leaf node of the CART	0	10	0
min_child_weight	XGBoost	Minimum sum of instance weight in a split	0.5	5	1
max_depth	XGBoost	Max depth of each CART model in XGBoost	3	10	3
decision threshold	XGBoost	Decision threshold on the XGBoost classifier output	0	1	0.5

5.2 Hyperparameter tuning process

Hyperparameter tuning is based on pre-defined measures of classification performance, and all performance measures are derived from the elements of the so-called confusion matrix, as shown in Table 5.2. The commonly used accuracy, (TP+TN)/(TP+FP+FN+TN), is not a good measure for the unbalanced RI cases. Instead, Probability Of Detection (POD), False Alarm Ratio (FAR), Peirce's Skill Score (PSS), and kappa scores are often used in RI prediction evaluations (Wilks 2011; Yang 2016; Kaplan et al. 2015).

Table 5.2: Confusion matrix.

	Predicted positive	Predicted negative
Actual positive	Truth positive (TP)	False negative (FN)
Actual negative	False positive (FP)	Truth negative (TN)

POD (aka recall) is defined as TP/ (TP + FN), or the ratio of correct positive prediction cases to all the positive cases. FAR is defined as FP/ (TP + FP) measuring the false positive prediction ratio compared to all the positive predictions. PSS, defined as TP/(TP + FN) - FP/(FP+TN), can be interpreted as the sum of the class level accuracy. And the kappa score, defined as 2*(TP*TN-FN*FP)/[(TP+FN)(FN+TN)-(TP+FP)(FP+TN)], conceptually the same as Brier Skill Score (BSS), measuring the relative improvement of the prediction against the prediction based on samples without

any models (Wilks 2011). In this study, a single metric, the kappa score, is mainly used for the hyperparameters tuning. Other measures are used mostly to report the performance of the prediction and to compare with previous studies.

To find the optimal values of the hyperparameter set (maximizing the kappa score in this study), grid search is performed in most previous works but the grid search is too time-consuming, especially with a large number of hyperparameters in the model. Bayesian optimization (BO) is then designed to reduce the time consumption, which uses an iteration procedure to search the global optimum. In reality, it is difficult for BO to find the global optimum, and instead, the BO will converge to local optima and diverge from them during the global optimum searching process. Therefore, the iteration should stop at a pre-defined maximum iteration number to avoid almost never-end global optimum search (Snoek et al. 2012; Shahriari et al. 2015). In this study, BO is used with pre-defined ranges for most of the searched hyperparameters, and the ranges are independent of each other.

CHAPTER 6 RESULT

In data mining and machine learning, a dataset is usually divided into training dataset, validation dataset, and test dataset. The training dataset is used to train the model, the validation dataset is adopted for hyperparameter tuning, and the test dataset is employed to provide the final assessment of the model (James et al. 2013).

In this study, the whole dataset is divided into 90% for training-validation and 10% for test. To further divide the training-validation dataset into the training dataset and validation dataset, the entire training-validation dataset is split into ten mutually exclusive equal-sized subsets. One of the subsets is reserved as the validation dataset, and the rest nine subsets are used for training. Each of the subsets is used in turns for validation dataset once. This is defined as 10-fold cross-validation (Kohavi 1995). The model's performance (kappa score) is evaluated by the mean performance (mean kappa score) on the 10 validation datasets.

When the study is first conducted with the SHIPS data, there were 11,317 instances (cases) from 1982 to 2016, and 571 (approximately 5%) were under rapid intensification (RI). A random stratified sampling on RI and non-RI cases was drawn with a similar proportion, and that resulted in 10,185 instances (including 523 RI cases, 5.1%) in the training-validation set, and 1,132 instances (including 48 RI cases, 4.2%) in the test dataset.

After the training was done, however, 465 instances in 2017 and the last tropical cyclone in 2016 are added to the SHIPS developmental database, and all of these

instances are added to the test dataset. The test dataset proportion is ended up with 1,597 (14.1%) instances in total with 95 RI instances (5.9%).

Therefore, in this study, COR-SHIPS, LLE-SHIPS, and DL-SHIPS models are trained using the training dataset initially. Then their hyperparameter tuning process will be derived on the validation dataset following steps introduced in Chapter 6, and their performance will be evaluated in the test dataset, which will be compared with previous works in Y16 and KRD15. Finally, their variable importance will be evaluated and discussed.

All algorithms, including those for data processing, data visualization, and data mining and machine learning in this study are performed with R (version 3.5.1), python base (version 3.7.0), python multiprocessing package (version 2.5), scikit-learn package (version 1.9.2), XGBoost package (version 0.83), and pyspark package (Spark API) (version 2.21). The entire process is implemented on Amazon Web service (72 cpus (3.0 GHz Intel Xeon Platinum processors), 144 G memory), and a local machine (8 i7 cores, 32G RAM, and GTX 1080).

6.1 COR-SHIPS model

6.1.1 Hyperparameter tuning for model selection

6.1.1.1 Hyperparameters tuning for SHIPS data filter

The structure of the COR-SHIPS model is displayed in Figure 1.1, and the correlation threshold in the SHIPS data filter is tuned based on the trial-and-error with 4 values, *0.7, 0.8, 0.9,* and *0.95.* For each of the correlation threshold, variables are first filtered as discussed before. Then Bayesian Optimization with 40 iterations is used to

tune hyperparameters in Table 4.1 and Table 5.1 with no clustering and the preset 0.5 classification decision threshold. With each iteration, a 10 cross-validation kappa score is recorded with a corresponding set of hyperparameter values, and the top 5 out of 40 kappa scores for each threshold are listed in Table 6.1. Threshold 0.95, 0.9, 0.8, 0.7 reach mean kappa scores of 0.401, 0.409, 0.411, and 0.343, respectively. This indicates the model performs the best at 0.8 and 0.9 among the four given numbers.

Table 6.1: Kappa scores of the 5 best 10-fold cross-validation results and their means for different correlation thresholds. The name of the 1st to 5th indicates the 1st to 5th best kappa scores. The "number variables selected" is the number of variables kept after highly correlated variables removal.

Correlation Threshold	5th	4th	3rd	2nd	1st	Mean	Number variables selected
0.7	0.314	0.321	0.334	0.352	0.392	0.343	56
0.8	0.404	0.407	0.411	0.417	0.418	0.411	72
0.9	0.402	0.403	0.405	0.415	0.419	0.409	99
0.95	0.387	0.397	0.401	0.409	0.411	0.401	136

Table 6.1 also shows there are 136, 99, 72, and 56 variables left for threshold 0.95, 0.9, 0.8, and 0.7, respectively after filtering. Approximately 30 variables are reduced when the threshold is changed from 0.95 to 0.9, 0.9 to 0.8, and 0.8 to 0.7. Having less number of variables results in a lower possibility of overfitting, and the score of threshold 0.8 is higher than that of 0.9. Therefore, 0.8 is selected as the correlation threshold for removing highly correlated attributes in the SHIPS data. After the removal,

there are 72 groups of highly correlated variables, as listed in Table A1, and the first variable in each group will be selected to form the 72 selected variables.

6.1.1.2 The number of clusters selected in GMM-SMOTE

After the correlation threshold is determined, the hyperparameters for GMM-SMOTE and XGBoost still need to be tuned for the best results. In GMM tunning for the "optimal" cluster numbers, BIC is chosen as the selection criterion, and the BIC values with the different number of clusters are displayed in Figure 6.1. The BIC values decrease with the cluster number first and then increase. The BIC values decrease with the cluster number for small cluster numbers, but stop decreasing at n_cluster=6, and increases when the cluster numbers increase to 7 and 8. Therefore, n_cluster is selected as 6, associated with the lowest BIC value.

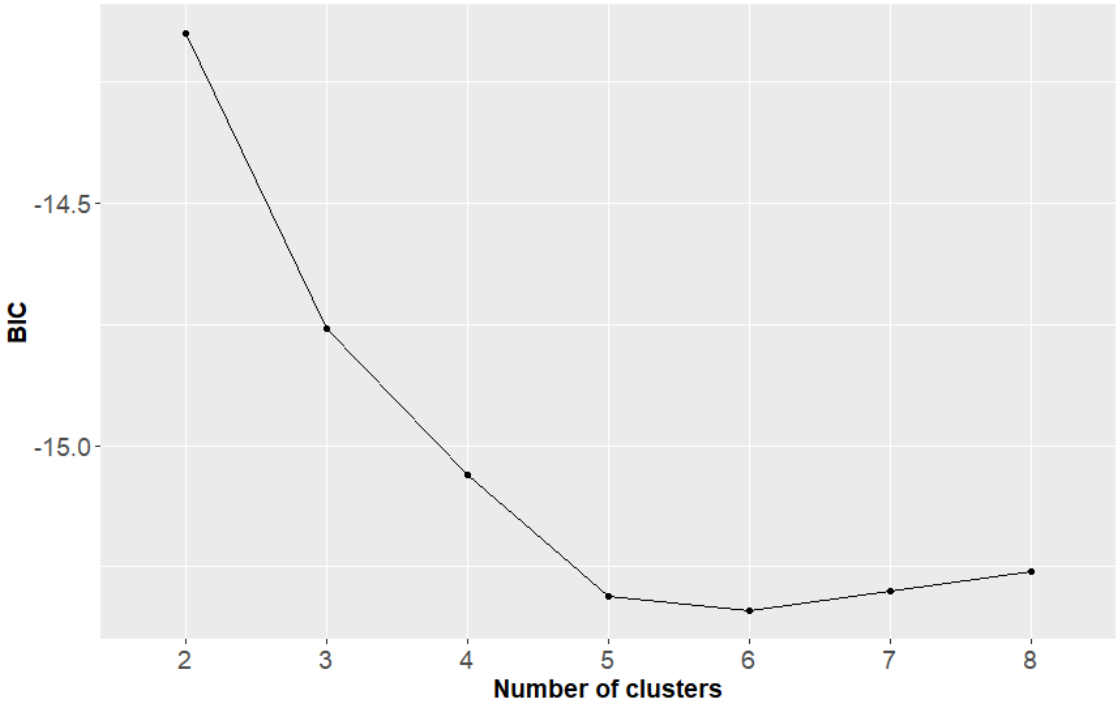


Figure 6.1: BIC (10^5) for GMM with different number of clusters.

The six clusters with the numbers of minority (RI) and total instances, and the IIR in each are displayed in Table 6.2. As we defined danger clusters with 0.2-5 IIR range, Clusters 3 and 5 could be excluded due to too few minority cases in the following augmentation, which removed a total of 2,401 instances with 17 RI cases (0.71%).

Table 6.2: The number of minority and total instances, and the Imbalance Ratio (with population RI ratio at 5.1%) for the 6 clusters.

Cluster	1	2	3	4	5	6	Total
Number of the minority instance	84	69	12	235	5	118	523
Number of the total instance	2275	1481	1255	2390	1146	1638	10185
Imbalance Rate	0.724	0.914	0.187	1.928	0.086	1.413	1

6.1.1.3 Hyperparameters tuning for GMM-SMOTE and XGBoost

While the correlation threshold in the SHIPS data filter and the cluster number are tuned separately, all other eleven hyperparameters except for the decision threshold are tuned together by Bayesian Optimization (BO). Those parameters are listed in Table 4.1 and Table 5.1 with the pre-defined searching spaces and initial values.

Figure 6.2 shows the change of the 10-fold cross-validation kappa scores, the tuning criterion, over a total 40 BO iterations. The kappa score is fluctuating over the iterations. For example, BO process helps the model reach a local optimum at iteration 8, and diverges from the local optimum to look for the global optimum and reach another local optimum after iteration 11. This process continues until the global optimum is found, which is barely possible, or the maximum iteration, 40 preset in this case, is reached. Since the trend with the iteration is unpredictable, hyperparameter sets with the best 5 kappa scores are selected, and the scores and hyperparameter values are displayed in Table 6.3. The different hyperparameter set is named as M*X*, where *X* is defined by the iteration number. For example, M11 implies the parameter set is selected after the 11^{th} iteration.

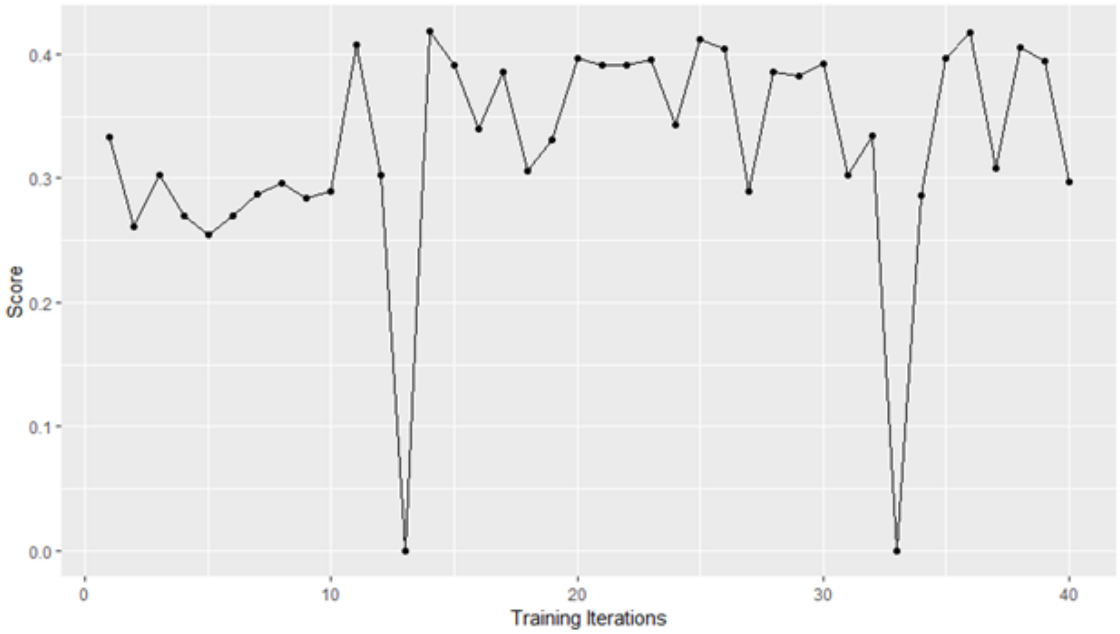


Figure 6.2: Variation of Cross-validation kappa scores over Bayesian Optimization iteration numbers.

Based on Table 6.3, the performance of M14 (0.418) and M36 (kappa=0.417) are better than M25 (0.411), M38 (0.405), and M11(0.407). As indicated in previous sections, lower k_neighbors, shrinkage, n_estimators, subsample, colsample_bytree, max_depth, and higher m_neighbors, reg_alpha, reg_lambda, gamma, min_child_weight will lead to a more conservative prediction model. Therefore, we will analyze the values of hyperparameters to find a balanced (not too conservative and not to aggressive) model based on the ranking of those values.

Table 6.3: Top performed hyperparameter sets, the corresponding cross-validation kappa scores, and specific values of the tuned hyperparameters. The numbers after "M" denoting the iteration numbers.

Name	M38	M11	M25	M36	M14
Kappa score	0.405	0.407	0.411	0.417	0.418
m_neighbors	4	3	3	3	5
k_neighbors	6	11	9	10	10
shrinkage	0.29	0.23	0.3	0.3	0.21
n_estimators	2000	572	2000	376	1510
subsample	0.75	0.5	0.5	0.5	0.67
colsample	0.99	0.78	0.99	0.9	0.99
reg_alpha	0.5	1.34	0.5	0.5	0.5
reg_lambda	20	20	20	18.91	20
gamma	0	0	0	0	0
min_child_weight	0.5	0.5	2.12	1.26	0.91
max_depth	7	8	7	7	10

To look for the balanced model, a system is designed to score the conservativeness of the hyperparameter set. The scores are based on a 1 (the least conservative) to 5 (the most conservative) scale associated with the hyperparameter value ranks, as listed in Table 6.4. For hyperparameters favoring smaller values for conservativeness (k_neighbors, shrinkage, n_estimators, subsample, colsample, max_depth), the scores are the same as the descending parameter value ranks. When a tie appears, the tied values will have the same rank (score), and the next rank value depends on how many values tie together. For example, in k_neighbors, M11 has the largest value, 11, hence M11 is scored 1. M36 and M14 have the second largest value, 10, hence they are scored 2. The next largest value, 9, is ranked the 4^{th} (instead of the 3^{rd}) in M25, and is scored 4. For other hyperparameters (m_neighbors, reg_alpha, reg_lambda, gamma, and min_child_weight), the conservativeness ranking scores are opposite to the

descending value ranks. After the ranking scores are assigned to all of the 11 hyperparameters in the five local optimal cases, the scores are summed up for the five cases (Table 6.4). Since our goal is to choose a model neither conservative nor aggressive, the parameter set M36 with the middle conservativeness ranking score is chosen for following implementation and discussion.

Table 6.4: The descending value ranking of individual hyperparameter among the top 5 performed cases, and the corresponding conservativeness ranking scores in parentheses. The variables with normal font are those favoring smaller values for conservativeness, and those italicized favoring larger values.

Name	M38	M11	M25	M36	M14
m_neighbors	*2 (4)*	*3 (1)*	*3 (1)*	*3 (1)*	*1 (5)*
k_neighbors	5 (5)	1 (1)	4 (4)	2 (2)	2 (2)
shrinkage	3 (3)	4 (4)	1 (1)	1 (1)	5 (5)
n_estimators	1 (1)	4 (4)	1 (1)	5 (5)	3 (3)
subsample	1 (1)	3 (3)	3 (3)	3 (3)	2 (2)
colsample	1 (1)	3 (3)	1 (1)	2 (2)	1 (1)
reg_alpha	*2 (1)*	*1 (5)*	*2 (1)*	*2 (1)*	*2 (1)*
reg_lambda	*1 (2)*	*1 (2)*	*1 (2)*	*5 (1)*	*1 (2)*
gamma	*1 (1)*	*1 (1)*	*1 (1)*	*1 (1)*	*1 (1)*
min_child_weight	*5 (1)*	*5 (1)*	*1 (5)*	*2 (4)*	*3 (3)*
max_depth	3 (3)	2 (2)	3 (3)	3 (3)	1 (1)
Total score	**23**	**27**	**23**	**24**	**26**

6.1.1.4 Hyperparameters tuning for XGBoost decision threshold

The last tuned hyperparameter is the decision threshold on the XGBoost classifier output, which was set as 0.5 before tuning. To tune this hyperparameter, we use a graphic method based on the relative values of POD and FAR as well as the kappa score. Since

POD and FAR are monotonically decreasing function with the decision threshold, we instead use precision (1-FAR) for identifying a threshold that balances the POD and FAR.

Figure 6.3a displays the variations of precision and POD as functions of the decision threshold. The precision and POD curves cross each other around 0.2 of the threshold value, a relatively balanced point for POD and FAR. At the same point, the kappa scores shown in Figure 6.3b close to the highest value, 0.35. As a result, 0.2 is selected as the decision threshold, which is expected to balance the POD and FAR, and therefore could minimize the overfitting effect in the final classification results.

(a) (b)

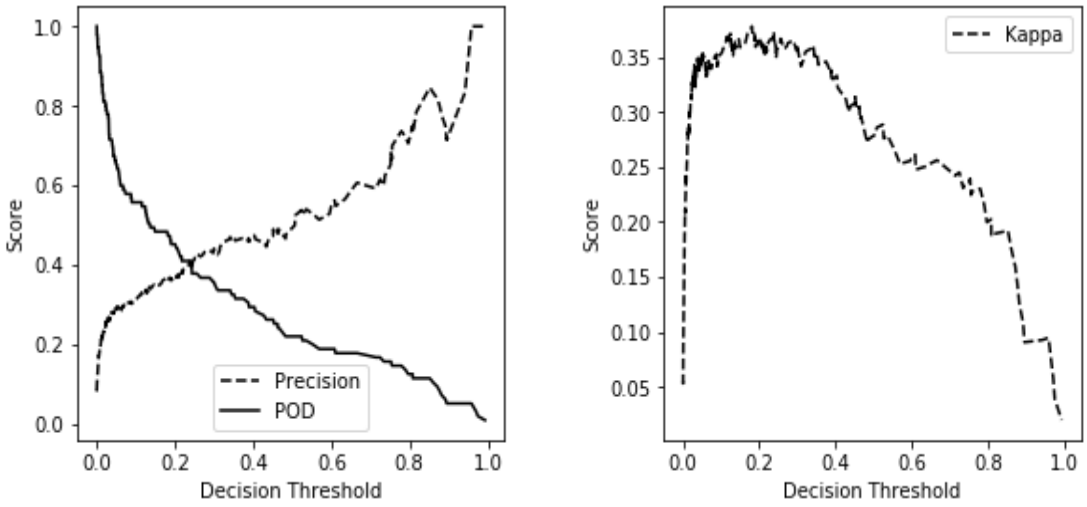


Figure 6.3: (a) Precision and POD score vs. decision threshold, and (b) Kappa score vs. decision threshold.

6.1.2 COR-SHIPS result on test data

Unlike the traditional classifier such as decision tree (Yang et al. 2016), modern classifier such as the XGBoost gives almost perfect classification on the training-validation data, i.e., POD≈1 and FAR≈0. Due to this fact, the performance measures on the training results are of little value, and therefore, the evaluation of the prediction is on the test data only.

The confusion matrix for the testing data, before hyperparameter tuning (MB), and after hyperparameter tuning (MA) is displayed in Table 6.5. The hyperparameter set of MB is displayed in the last column of Table 4.1, and Table 5.1 (initial values set by the software) with the decision threshold as 0.5, and the MA is with the hyperparameter set of M36 (Table 6.3) with the 0.2 decision threshold and 6 clusters. MA's TN (1,438) and FN (56) are slightly smaller than 1,447 and 62 of MB, while MA's TP (39) and FP (64) are larger than 33 and 55 of MB. Smaller TP and larger FN in MB implies that MB is worse at correctly predicting RI instances, and vice versa. It seems that there is a trade-off between correctly predicting RI and non-RI, i.e., if we want to better predict the RI instance, we should sacrifice the power we predict for non-RI instances, and vice versa.

Table 6.5: Confusion matrix values of our model after (before) hyperparameter tuning

	Predicted RI	Predicted non-RI	Actual
Actual RI	39 (33)	56 (62)	95
Actual non-RI	64 (55)	1438 (1447)	1502

Predicted	103 (88)	1494 (1509)	

Kappa, PSS, POD, and FAR are used for the model evaluation, and their values for MB and MA are elaborated in Table 6.6. The POD and FAR values for MB and MA cases demonstrated the importance of hyperparameter tuning. After tuning, POD increases 26.1% from 0.326 to 0.411, while FAR increase only from 0.617 to 0.621, almost nill (0.6%). That is, the benefit of higher correct RI prediction is much higher than the cost of false alarm with the hyperparameter tuning. The overall statistics PSS and kappa score also increased from 0.293 to 0.368 (25.6%) and from 0.315 to 0.354 (12.4%), respectively, confirming the significant improvement on RI prediction with the hyperparameter tuning procedure.

Table 6.6: Performance comparisons. MB and MA denote the models before and after the hyperparameter tuning.

Model	Kappa	PSS	POD	FAR
MB	0.315	0.293	0.326	0.617
MA	0.354	0.368	0.411	0.621
Improvement MB	12.4%	25.6%	26.1%	+0.6%

6.1.3 Feature importance

Generally, the variable (feature) importance is used to leverage the variable contribution and is defined as a quantitative score. The higher the score is, the more the variable contributes and the more useful that variable is for classifying RI. The classifier

used in this study, XGBoost, provides the scaled importance scores with the sum of all scores being one.

Table 6.7 displays the variables with the top 10 importance scores and their definition (SHIPS 2018c). The scores of the full 72 variables are given in Table A4. The past 12-hour intensity change, BD12, has the largest importance score, 0.0362, which almost doubles the importance score of the second important variable. Because BD18 and BD06 are highly correlated with BD12 (see Table A1), we can safely assume that they are as important as BD12. The second most important variable is DTL, the distance from a TC to the nearest major land. The importance of DTL is slightly higher than the third to seventh most important variables, CFLX, SHRD, G150, jd, and VAMX, which are related to dry air, vertical wind shear magnitude at 850-200 hPa, the temperature perturbation at 150 hPa, Julian day, and the current TC intensity. The eighth to ninth variables are IRM1_5 and PW08. The tenth most important variable is VMPI, the Maximum potential intensity, which ranked higher in other RI studies.

Table 6.7: Features of top ten importance, their importance scores, and feature description from SHIPS (2018c) in the COR-SHIPS model.

Variable	Importance	Description
BD12	0.0362	The past 12 hour intensity change
DTL	0.0217	The distance to nearest major land
CFLX	0.0207	Dry air predictor based on the difference in surface moisture flux between air with the observed (GFS) RH value, and with RH of air mixed from 500 hPa to the surface

SHDC	0.0206	850-200 hPa shear magnitude (kt *10) vs time (200-800 km) but with vortex removed and averaged from 0-500 km relative to 850 hPa vortex center
G150	0.0205	Temperature perturbation at 150 hPa due to the symmetric vortex calculated from the gradient thermal wind. Averaged from r=200 to 800 km centered on input lat/lon (not always the model/analysis vortex position) (deg C*10)
jd	0.0204	Julian date
VMAX	0.0201	Maximum Surface Wind
IRM1_5	0.0199	Predictors from GOES data (not time dependent) for r=100-300 km but at 1.5 hours before initial time
PW08	0.0191	Time dependent 600-800 km TPW standard deviation (mm * 10)
VMPI	0.0190	Maximum potential intensity from Kerry Emanuel equation (kt)

It is interesting to notice that, IRM1_5, the standard deviation (STD) of GOES (Knaff et al. 2008) BT (brightness temperature) in 100-300 km radius 1.5 hours before the initial time, is more important than the average BT value itself (IRM1_2). The phenomenon plausibly says that the non-uniform BT distribution around TC center plays a more important role than the uniform BT level for the RI. The same thing takes place with PW08, the 600-800 km TPW (Total Precipitable Water) standard deviation from the GFS analysis (Berger 2016), which is more important than the corresponding TPW value, PW07 represented by the highly correlated RHMD (Table A1). This finding is consistent with the relationship between TC intensity and the symmetricity of the TC structure. Asif

et al. (2020) used the STD and other statistics of brightness temperature in centric bands to establish a relationship with TC intensity, and those statistics play a similar role of the variance of the deviation angle described by Piñeros et al. (2011) and Ritchie et al. (2012).

6.2 LLE-SHIPS model

6.2.1 Hyperparameter tuning for model selection

6.2.1.1 Hyperparameters tuning for data filters

LLE-SHIPS model is trained with both the SHIPS data and the near core ERA gridded data. The SHIPS data will inherit the filtered data for the COR-SHIPS model, and the ERA data will be filtered with LLE. Two new hyperparameters will present with the LLE filter, the number of the nearest neighbors for each observation (no_neighbors), and the number of dimensions in the reduced space (no_dimension). As we did in the tuning process for the COR-SHIPS model, BO with 40 iterations is used to tune the two new hyperparameters with no clustering and the preset 0.5 classification decision threshold. In addition, a 10-fold cross-validation kappa score is recorded with a corresponding set of hyperparameter values in each iteration, and the 5 hyperparameter sets with the best 10-fold cross-validation kappa scores are listed in Table 6.8. The search range for no_neighbors and no_dimension are pre-defined to 5 to 15 and 10 to 90, respectively.

Based on Table 6.8, we can find that the best kappa score is 0.297 achieved at no_dimension being 90, and no_neighbors equaling 15. Therefore, 90 and 15 are selected as final parameter values for no_dimension and no_neighbors. With the above filter

setting, the original 4*37*14 = 2072 features are filtered into 90 new variables, named as lle1 to lle90. Based on the property of LLE, lle1 to lle90 are independent of each other, i.e., the correlation between any of them is 0. In addition, after calculation, the absolute correlation between new lle variables and the SHIPS variables are less than 0.8; hence no additional variables will be removed in this phase.

Table 6.8: The performance for models with different sets of values of the hyperparameters, no_dimension and no_neighbors.

Kappa Score	Rank	no_dimension	no_neighbors
0.297	1	90	15
0.295	2	89	15
0.290	3	87	13
0.287	4	90	14
0.282	5	80	15

6.2.1.2 The number of clusters selected in GMM-SMOTE

After the hyperparameters in data filters are tuned, the hyperparameters for GMM-SMOTE and XGBoost still need to be tuned for the best results. Similar to the COR-SHIPS model, the BIC values with the different number of clusters are displayed in Figure 6.4, and the BIC values decrease with the cluster number first and then increase. BIC stops decreasing for the next two iterations first at cluster 7, and the minimum BIC is reached at 5 before reaching at 7. Therefore, n_cluster is selected as 5.

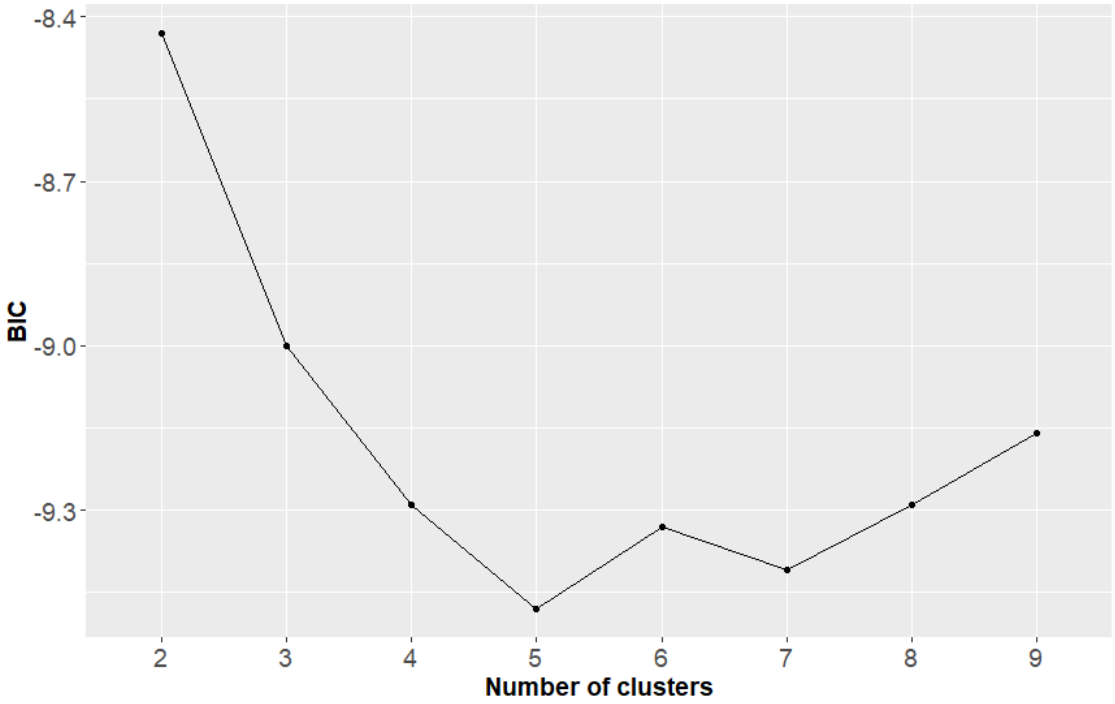


Figure 6.4: BIC (10^6) for GMM with a different number of clusters in LLE-SHIPS model.

The clustering result is displayed in Table 6.9 with the numbers of minority (RI) and total instances, and the IIR in each cluster. As we defined danger clusters with a 0.2-5 IIR range, all the clusters are included in the following augmentation. Although there is no instance removed, the synthetic instances are created only using instances in the same cluster, which decreases the possibility of outlier creations.

Table 6.9: The number of minority and total cases, and the Imbalance Rate (with population RI ratio at 5.1%) for the 5 clusters generated by GMM.

Cluster	1	2	3	4	5	Total
Number of the minority instance	196	26	112	39	150	523
Number of the total instance	3048	1858	1078	1608	2593	10185
Imbalance Rate	1.286	0.280	2.078	0.485	1.157	1

6.2.1.3 Hyperparameters tuning for GMM-SMOTE and XGBoost

While the hyperparameter tuning is conducted separately for the two data filters, the hyperparameters in the GMM-SMOTE and XGBoost in Table 4.1 and Table 5.1 with their pre-defined searching space are tuned together by BO as for the COR-SHIPS model. Figure 6.5 shows the change of the 10-fold cross-validation kappa scores, the tuning criterion, over a total 40 BO iterations.

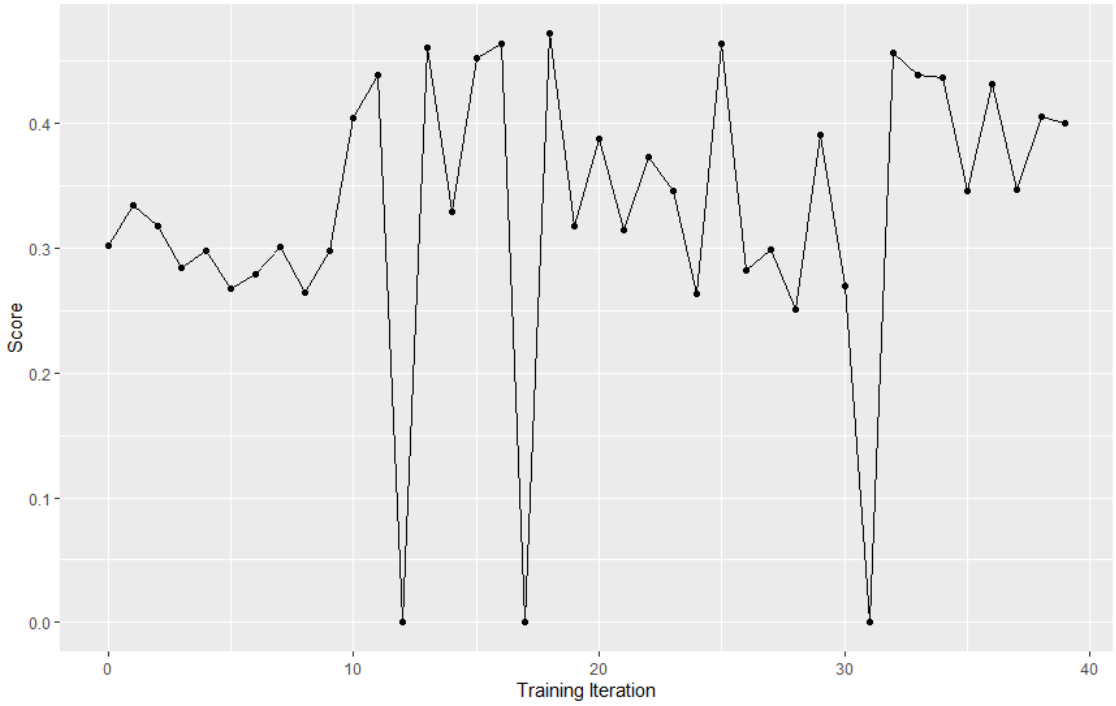


Figure 6.5: Variation of Cross-validation kappa scores over Bayesian Optimization iteration numbers for LLE-SHIPS model.

As shown in Table 6.10, the top 5 performed hyperparameter sets are M18 (kappa=0.472), M25 (0.464), M16 (0.464), M13 (0.461), and M32 (0.456). To find a balanced model, the same score system described for the COR-SHIPS model is used, and the value ranks and their corresponding conservativeness ranking scores are listed in Table 6.11. The total conservativeness scores are also calculated, which are 25, 20, 30, 31, and 23, respectively for M18, M25, M16, M13, and M32, and the parameter set M18 with the middle conservativeness ranking score is chosen for the following implementation and discussion. Coincidently, M18 is also the set associated with the highest kappa score.

Table 6.10: Top 5 performed hyperparameter sets, the corresponding cross-validation kappa scores, and specific values of the tuned hyperparameters. The numbers after "M" denoting the iteration numbers.

Name	M18	M25	M16	M13	M32
Kappa score	0.472	0.464	0.464	0.461	0.456
n_cluster	5	5	5	5	5
m_neighbors	8	8	3	7	7
k_neighbors	12	14	3	14	14
shrinkage	0.21	0.16	0.23	0.16	0.16
n_estimators	731	2000	2000	1286	1819
subsample	0.85	0.9	0.64	0.77	0.72
colsample_bytree	0.8	0.99	0.4	0.4	0.99
reg_alpha	0.1	0.1	0.1	0.1	0.1
reg_lambda	0.5	0.5	20	17.93	12.2
gamma	0	0	0	0	0
min_child_weight	1.4	0.5	0.5	3.48	0.5
max_depth	10	10	3	7	8

Table 6.11: The descending value ranking of individual hyperparameter among the top 5 performed cases, and the corresponding conservativeness ranking scores in parentheses. The variables with normal font are those favoring smaller values for conservativeness, and those *italicized* favoring larger values.

Name	M18	M25	M16	M13	M32
m_neighbors	*1 (4)*	*1 (4)*	*5 (1)*	*3 (2)*	*3 (2)*
k_neighbors	4 (4)	1 (1)	5 (5)	1 (1)	1 (1)
shrinkage	2 (2)	3 (3)	1 (1)	3 (3)	3 (3)
n_estimators	5 (5)	1 (1)	1 (1)	4 (4)	3 (3)
subsample	2 (2)	1 (1)	5 (5)	3 (3)	4 (4)
colsample_bytree	3 (3)	1 (1)	4 (4)	4 (4)	1 (1)
reg_alpha	*1 (1)*	*1 (1)*	*1 (1)*	*1 (1)*	*1 (1)*
reg_lambda	*4 (1)*	*4 (1)*	*1 (5)*	*2 (4)*	*3 (3)*
gamma	*1 (1)*	*1 (1)*	*1 (1)*	*1 (1)*	*1 (1)*
min_child_weight	*2 (4)*	*3 (1)*	*3 (1)*	*1 (5)*	*3 (1)*
max_depth	1 (1)	1 (1)	5 (5)	4 (4)	3 (3)
Total score	28	16	30	32	23

6.2.1.4 Hyperparameter tuning for the decision threshold

The last tuned hyperparameter is the decision threshold on the XGBoost classifier output. Initially, the decision threshold is set as 0.5. Figure 6.6 (b) displays that the kappa score approximates the highest value, 0.52 when the decision threshold reaches 0.15. Figure 6.6 (a) displays variations of precision (1-FAR) and POD variations as functions of the decision threshold from 10-fold cross-validation in the training/validation data. Decision threshold of 0.15 also close to the intersection of the POD and precision score, a relatively balanced point for POD and FAR. As a result, 0.15 is selected as the decision threshold.

(a) (b)

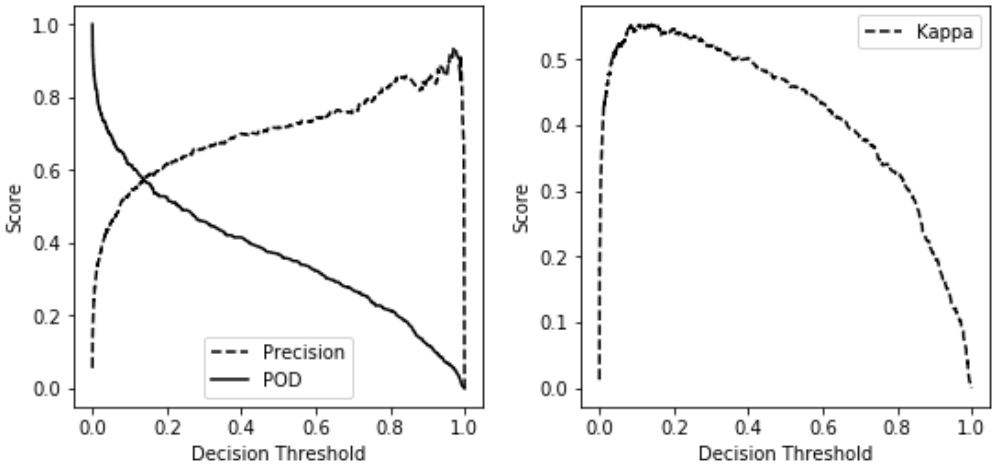


Figure 6.6: (a) Precision and POD score vs. decision threshold. (b) Kappa score vs. decision threshold in LLE-SHIPS model.

6.2.2 LLE-SHIPS result on test data

As discussed in the COR-SHIPS model, the evaluation of the prediction performance is on test data only. The test confusion matrix for the model, before hyperparameter tuning (MB), and after hyperparameter tuning (MA) is displayed in Table 6.12. Learning from the Table, MA's TN (1,469) and FN (55) is slightly smaller than 1,491 and 75 of MB, while MA's TP (40) and FP (33) is significant larger than 20 and 11 of MB.

Table 6.12: Confusion matrix values after (before) hyperparameter tuning with the test data.

	Predicted RI	Predicted non-RI	Actual
Actual RI	40 (20)	55 (75)	95
Actual non-RI	33 (11)	1469 (1491)	1502
Predicted	73 (88)	1524 (1509)	

Kappa, PSS, POD, and FAR are used for the model evaluation, and their values for MB and MA are elaborated in Table 6.13. The POD and FAR values for MB and MA cases demonstrated the importance of hyperparameter tuning. After tuning, POD increases 99.5% from 0.211 to 0.421, while FAR decreases from 0.645 to 0.563, 12.7%. The overall statistics PSS and kappa score also increased from 0.203 to 0.399 (96.6%) and from 0.297 to 0.454 (52.9%), respectively, confirming the significant improvement on RI prediction with the hyperparameter tuning procedure, and apparently, the model

was overfitted before tuning process with so many features. Furthermore, the improvement is almost 4 times than that of the COR-SHIPS model (25.6% and 12.4% in PSS and kappa improvement), indicating the hyperparameter tuning is more efficient in the more complicated LLE-SHIPS model.

Table 6.13: Performance comparisons. MB and MA denote the models before and after the hyperparameters in GMM-SMOTE and XGBoost are tuned.

Model	Kappa	PSS	POD	FAR
MB	0.297	0.203	0.211	0.645
MA	0.454	0.399	0.421	0.563
Improvement MB	52.9%	96.6%	99.5%	-12.7%

6.2.3 Feature importance

We learned from section 6.1.3 that how XGBoost evaluates the variable (feature) importance score for COR-SHIPS model. In LLE-SHIPS model, the same approach can be used to calculate the importance score for the SHIPS variables and lle1, …, lle90, but we cannot directly calculate the importance score for the original ERA-Interim variables in their feature space; hence they cannot be linked to the original ERA variables. So instead, we try to relate the IS of all the lle1, …, lle90 to individual ERA parameter groups (based on correlation). Therefore, the feature importance evaluation in the LLE-SHIPS model is divided into 2-steps. First, the XGBoost is used to evaluate the importance score for SHIPS variables and lle1 to lle90 in the same way as in COR-SHIPS model, and then a feature permutation approach is used to evaluate the importance

score for the original ERA-Interim feature space separately based on the importance score generated from the first step.

6.2.3.1 Variable importance in XGBoost

Table 6.14 displays the 10 most important variables among the 162 selected not highly correlated variables (90 LLE variables and 72 SHIPS variables) and their definition. We need to notice that none of the LLE variables are among the top 10 variables. The reason might be there are too many variables generated from the LLE, and having so many variables reduces the importance score for each of them. The assumption is confirmed by summing the importance score for lle1 to lle90, which is 0.4288, only a bit smaller than that of the SHIPS variables (0.5712).

Table 6.14: Variables of top ten importance, their importance scores, and feature description from SHIPS (2018c) in LLE-SHIPS model.

Variable	Importance	Description
BD12	0.018769	The past 12 hour intensity change
VMAX	0.016706	Maximum Surface Wind
DTL	0.013821	The distance to nearest major land
SHRD	0.012952	850-200 hPa shear magnitude
TWXC	0.01151	Maximum 850 hPa symmetric tangential wind at 850 hPa from NCEP analysis
G150	0.011339	Temperature perturbation at 150 hPa due to the symmetric vortex calculated from the gradient thermal wind. Averaged from r=200 to 800 km centered on input lat/lon (not always the model/analysis vortex position) (deg C*10)
VMPI	0.011262	Maximum potential intensity from Kerry Emanuel equation (kt)
REFC	0.011262	Relative eddy momentum flux convergence

TGRD	0.01106	The magnitude of the temperature gradient between 850 and 700 hPa averaged from 0 to 500 km estimated from the geostrophic thermal wind
IRM1_5	0.010672	Predictors from GOES data (not time dependent) for r=100-300 km but at 1.5 hours before initial time

1793

1794 The past 12 hour intensity change, BD12, has the largest importance score,
1795 0.018769, which slightly better than the importance score of the second important
1796 variable, VMAX, the Maximum Surface Wind, and its highly correlated variable, the
1797 Minimum Sea Level Pressure. The importance score between BD12 and VMAX is very
1798 similar. The rest top 10 variables are DTL, SHRD, TWXC, G150, VMPI, REFC, TGRD,
1799 and IRM1_5. The highly correlated variable groups with the important variables and the
1800 importance scores of all of the 162 variables can also be found in Table A2.

1801 ***6.2.3.1 Group importance in LLE***

1802 Molnar (2019) described a feature permutation approach to evaluate the
1803 importance of features on training dataset for nonlinear models where the importance
1804 score cannot be derived easily. We assume that for the feature space in any given dataset
1805 is *X*, *f(X)* is the predicted value by the classifier *f*, and *y* is the ground truth. We further
1806 assume the loss of the classifier is *L(y, f(X))*. Then for each feature in the feature space *X*,
1807 permute its value to 0 for all the observations while keeping other features unchanged
1808 (represented as $X^{permute}$). Finally, the difference between the loss of the permuted
1809 feature space ($X^{permute}$), and the original loss is calculated for each feature, and the
1810 difference is used as its importance.

Although feature permutation is an efficient approach to evaluate the feature importance for different models, especially for a black-box model such as LLE, Molnar (2019) also indicates that the permutated feature importance could be biased by the highly correlated features. For example, if we evaluate the importance score for each of the 2,072 variables, the result, i.e., the importance score is not accurate due to the existence of the highly correlated variables because they could influence each other. Similar to the removal of highly correlated variables in the SHIPS data filter, pairwise correlations of all the features are calculated and compared with the correlation threshold 0.8. This process results in 135 groups, and the details of the Group are elaborated in Table A2. Then an importance score is calculated for each Group, and details will be elaborated later in the section.

The group-level importance score is calculated specifically as:

f: trained model; *X*: original feature space; *y*: ground truth; *L(y, f(X))*: loss between the ground truth and the predicted value by the classifier.

1. Calculate $Importance_{LLE\_Total}$ as the sum of the importance score of lle1 to lle90 derived from XGBoost, here is 0.4288.
2. Calculate the original model error *L(y, f)*.
3. for each group:
    a) Generate feature matrix $X^{permute}$ by setting features in that group to 0, which breaks the corresponding correlation between all the features
    b) Calculated error *L(y, f($X^{permute}$))*
    c) Estimate the importance for the group *imp = L(y, f($X^{permute}$)) - L(y, f)*

d) Associate the score to the group

e) Negative importance is set to 0

4. Group importance score is rescaled as attributing the total important scores by LLE variables based on the ratio of loss of a particular group to the total loss (sum of a group losses), the specific calculation is:

$$Imp_{group} = Importance_{LLE\_Total} * imp_{group} / \sum_{i\ for\ all\ the\ groups} imp_i \quad (6.1)$$

5. Sort groups by their number of features.

Based on the Algorithm, the importance score for each Group is calculated, and groups with the top five important scores are list in Table 6.15. Intuitively, turning much more variables to 0 could reduce the model's performance more than turning much fewer variables to 0 because changing more variables are likely to alternate the model's performance more. However, based on Table 6.15, we find that the top 5 does not contain too many groups with a large number of variables, which indicates that these groups with few variables play a more important role in RI prediction than other groups, especially the groups with significant more variables.

Table 6.15: ERA-Interim variable group with top 5 importance scores, calculated from the second step. The group number (Group), the number of variables in the group (Group size), and the importance score (Importance).

Group	Group size	Importance
49	5	0.023614
88	1	0.021988
1	309	0.019687
29	11	0.019662
3	148	0.017280

Group 49 (G49) has the highest importance score (IS), 0.024, and it has 5 variables, NT12_v_l18, NT06_v_l18, NT00_v_l17, NT00_v_l18, NT06_v_l17, which indicates that the northward wind speed on level 17 (450 hPa) at 6 hours before, and at present, together with level 18 (400 hPa) at 12 hours before, 6 hours before, and at present are important in RI prediction. We can find that the middle level's (400 hPa and 450 hPa) northward wind plays a significant role than that in the lower level, i.e., 1000 hPa, and higher level (1 hPa). The reason could be when the RI starts to occur, the northward wind speed in 400 and 450 hPa change faster than that of other levels. We can also find that both 6 hours before and the present northward wind speed are important at 400 and 450 hPa, which indicates that the northward wind speed in 400 and 450 hPa start to change immediately before the occurrence of RI, and 18 hours before are too long to influence the occurrence of RI.

Wang et al. (2015) found that "In the active (inactive) season, the low-level (deep layer) shear is more negatively correlated with the TC intensity change than the deep-layer (low level) shear." Our study identifies that importance of the northward wind

speed in the 400 and 450 hPa for the RI prediction, that could contribute to the 400 and 450 hPa VWS (vertical wind shear), which recognizes the importance of the mid-layer shear with regard to the intensity change in addition to the finding of Wang et al. (2015).

The second most important group, the G88 with an IS of 0.0220, only has one variable, NT18_pv_l1, the potential vorticity at 18 hours before on the first level (1000 hPa). The importance score for NT18_pv_l1 is even 17% higher than that for the most important variable in Table 6.13, BD12 with a 0.0188 score, which is also the highest importance for a single variable. This result demonstrated that the machine learning method could identify important features, which may not be in the commonly used data set, such as the SHIPS database. However, the role of pv in RI was identified by others already (e.g., Martinez et al. 2019; Tsujino and Kuo 2020). Tsujino and Kuo (2020) detailed the changed of pv during the RI of Supertyphoon Haiyan (2013) with numerical simulation. They emphasized the pv increasing around 3-5km height at the beginning stage of the RI. Carefully checking their results (Fig. 2b&c), one can find the pv actually increases simultaneously around the sea level in 20-40 km range from the center, which is the same as what we identified here by the NT18_pv_l1.

All other level 1 pv (3 of them) are grouped in G63 with importance scores (IS) (0.010746). All level 2 in G55 with 4 members and IS 0.004744. All other pv are in G4 with 140 members but IS being only 0.006264. Those numbers demonstrated that only lower layer pv affects the RI process.

The third most important Group is G1, which has 309 features in the Group, and with IS 0.0197. Since all types of ERA-Interim variables are included in the Group, it is difficult to trace back which variable is more important. This implies the SHIPS dataset is very useful because it removes a lot of highly correlated variables and only extract important information from these variables.

The fourth most important group is G29, which has the IS, 0.020, and consists of 11 variables, i.e., NT18_u_l1, NT12_u_l1, NT06_u_l1, NT00_t_l10, NT00_u_l16, NT00_u_l17, NT06_u_l17, NT12_u_l17, NT18_u_l18, NT12_u_l18. We can find that most of the variables in the group is u, the eastward wind speed. Similar with G49 but a slightly different, the eastward wind speed at level 17 (450 hPa), and 18 (400 hPa) play an important role in RI prediction. Other than 400 and 450 hPa eastward wind speed, the eastward wind at 1000 hPa at 6, 12, and 18 hours before RI also plays an important role. As discussed above, Wang et al. (2015) found that "low-level shear between 850 (or 700) and 1000 hPa is more negatively correlated with TC intensity change than any deep-layer shear during the active typhoon season," which matches our findings that eastward wind speed, related to the VWS, at 1000 hPa are significant in RI prediction. Additionally, we also recognize that the mid-level (400 and 450 hPa) eastward wind speed (VWS) are important to TC intensity change. One exception variable in the Group is the temperature (t) at 775 hPa, NT00_t_l10, although highly correlated with u in terms of value, possibly misplaced in the group because there's only one t variable in the Group.

The fifth most important Group, G3, has IS 0.017 and 148 variables, consists of w (the pressure vertical velocity) at all levels over 18 hours before, 12 hours before, 6 hours

before, and at present. This indicates the vertical pressure speed plays an important role in RI prediction, which includes SHIPS variable O500 (highly correlated with O700 in Table A1), ranked 72 in Table A5. This indicates that other than 500 and 700 hPa, the pressure vertical velocity (w) at other pressure levels is also important, and more researches should be done to figure out more details.

In sum, two out of the top five important groups, G45, and G29, contain eastward and northward wind speed variables, especially at 400, 450, and 1000 hPa, which indicates that wind velocity, hence the VWS at 400, 450, and 1000 hPa pressure level plays a significant role in RI prediction, not only matches what have been found in Wang et al. (2015) but also identifies the importance of the mid-level vertical shear to intensity change. Another group, G3, only contains the pressure vertical velocity, indicates that vertical pressure speed is critical in RI prediction. Other than the O500, and O700 included in the SHIPS database, it is necessary to dig out other significant pressure levels for the pressure vertical velocity. One variable, 18 hour before potential vorticity at 1000 hPa, is more significant than BD12, and more researches need to be done for this specific variable.

Here we derive the group level importance score for ERA-Interim variables. Although because the AI system is consisted of too many components that the score is not 100% accurate, the system is still able to identify useful features in addition to SHIPS database.

More details of the XGBoost scores can be found in Table A5, group importance scores can be found in Table A3.

6.3 DL-SHIPS model

6.3.1 Hyperparameters tuning and result

6.3.1.1 Hyperparameters tuning for data filter

The structure of the DL-SHIPS model is almost the same as that of the LLE-SHIPS model except that the ERA-Interim data is filtered with a CNN-based autoencoder network, as introduced in section 3.2.3. Similar to the LLE-SHIPS model, in the DL-SHIPS model, the correlation threshold (0.8) in the SHIPS data filter is inherited directly from that of the COR-SHIPS model. The difference is that in the DL-SHIPS model, unlike in the LLE-SHIPS model where all variables in ERA-Interim reanalysis data are treated together with only one dimension reducing model for the feature extraction, one autoencoder network is trained to extract information from each individual of 14 ERA-Interim variables. In other words, there are 14 different autoencoder networks in total for the 14 variables. Other components in the LLE-SHIPS model and the DL-SHIPS model are the same.

In the tuning process of the ERA-Interim data filter in the DL-SHIPS model, all 14 ERA-Interim original variables are initialed with the same 3D CNN auto-encoder structure as described in Figure 3.6. The to-be-determined hyperparameter, the dimension of the compressed feature (num), is pre-determined as 8, therefore, 8 new variables are generated from each network, labeled as variable+order in the compressed feature layer (’1’ to ‘8’). For example, v1 to v8 are new variables derived from the trained 3D CNN auto-encoder for variable v. A three-step tuning process for “num,” which is similar with the tuning process of SHIPS data filter. In the first step, training a separated 3D CNN

auto-encoder for each of the 14 variables for 200 epochs. In the second step, the output from SHIPS data filter (72 variables) and the output of the ERA data filter (the trained DL models), which is all the 14 variables from variable1 to variable8 (14*8 = 112) are concatenated to form the input to the sampler, where zero (zero values for all instances) and highly correlated variables are removed. Then the BO with 40 iterations is used to tune hyperparameters in for GMM-SMOTE in Table 4.1 and XGBoost related hyperparameters listed in Table 5.1 with no clustering, and the preset 0.5 classification decision threshold. Finally, the hyperparameter set with the highest 10-fold cross-validation is selected; instead of kappa score, here, we use the importance score for hyperparameter tuning instead of the kappa score used in the SHIPS data filter tuning. The importance score for each input variable is derived. In the third step, the importance for each of the 14 variables is calculated as the sum of the variable1 to variable8, and the num is determined by the summed importance score that will be discussed later.

Figure 6.7 shows the training losses (mean square error) of 14 auto-encoder networks change over iterations that are trained for 200 epochs, respectively, which indicates that all the networks are converged after 100 epochs. 8 new variables are engineered from each ERA-interim variable first, because of the characteristic of the auto-encoder, features with no information, i.e., zero feature, could be created. The zero features are listed in the last column of Table 6.16, and they are removed. A correlation check is conducted among the 72 SHIPS variables and the newly derived non-zero variables, and highly correlated (>0.8) variables should be removed. There are two correlations, i.e., the correlation between ERA variables vs. SHIPS variables, and the

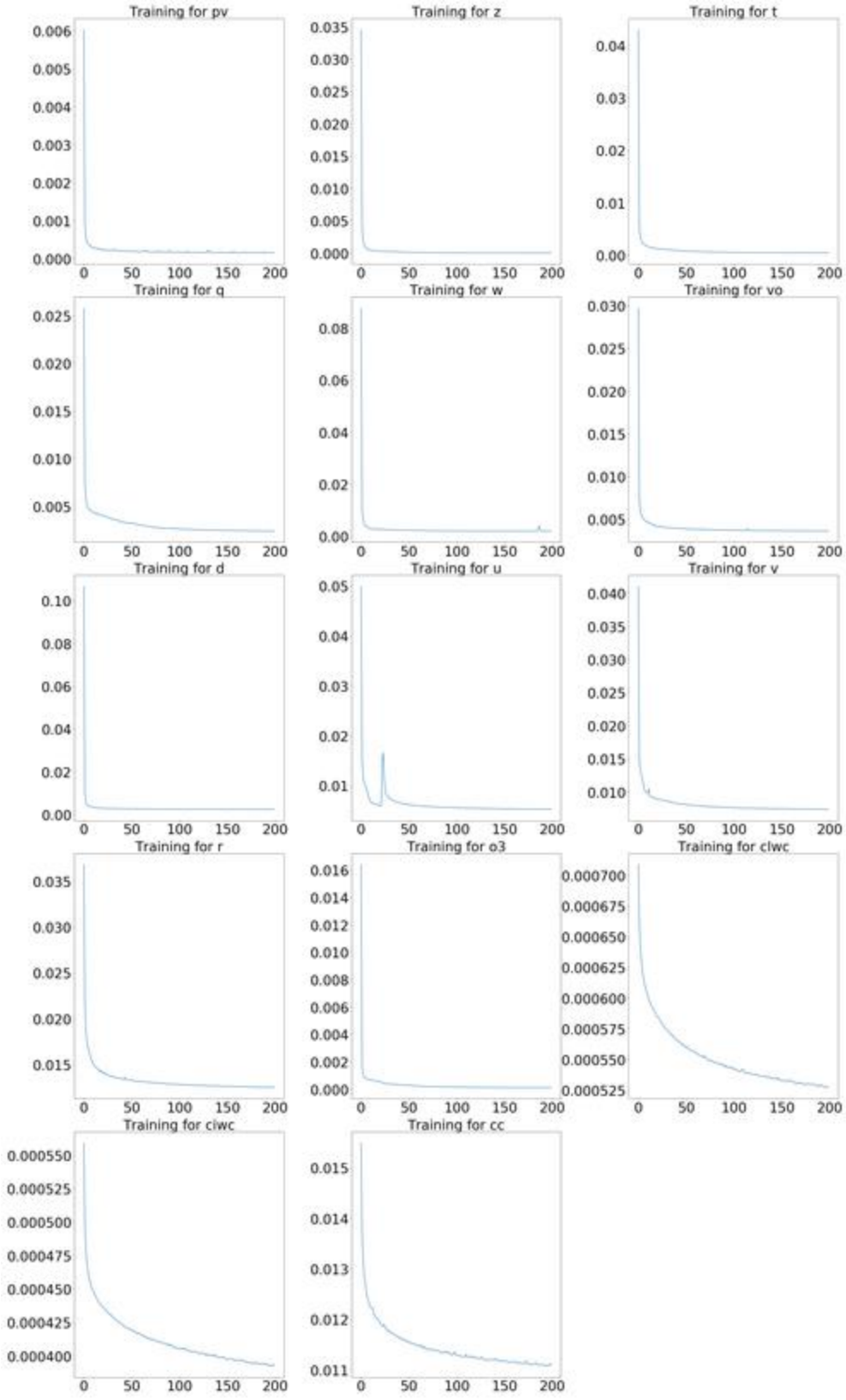
correlation between only ERA variables, as highly correlated SHIPS variables are already removed in section 3.1.3. Correlation between ERA variables vs. SHIPS variables is all less than 0.8 so no variable will be removed in this phase. However, when we check the correlation between ERA variables, we find highly correlated (>0.8) variables are existed, and should be removed. The same procedure in section 3.1.3 in SHIPS data filter is used again, and highly correlated variables are sorted in Table 6.17 (only variables with highly correlated ones are displayed). Therefore, d5, r4, clwc4, w6 are kept as well as those not highly correlated features, while d2, d3, cc8, q7, cc3, and w1 are removed. Then all remained features are concatenated together and fit into the GMM-SMOTE sampler and XGBoost classier. The hyperparameters of the model is tuned by BO based on no clustering, decision threshold set at 0.5, and other hyperparameters as listed in the final column of Table 4.1 and Table 5.1. The summed importance score for all 14 variables are displayed in Table 6.16, where variables are sorted based on their summed importance scores. Based on Table 6.16, we can find that with the decreasing of the summed importance score, the number of kept (removed) features are roughly decreasing (increasing). Therefore, we are tuning the hyperparameter, dimension of the compressed feature (num), based on the summed importance score for each variable, and the details are displayed in Table 6.16.

Table 6.16: 14 variables, their summed importance score, non-zero features extracted from each variable network, and the corresponding missing variables due to all zeros.

Variable	Summed Importance Score	The number of kept features	Corresponding missing features
q	0.062	6	q3, q8
r	0.055	7	r4
u	0.055	8	
v	0.056	7	v7
pv	0.050	6	pv7, pv8
vo	0.049	5	vo1, vo6, vo7
w	0.042	6	w3, w4
d	0.039	5	d1, d4, d6
t	0.020	4	t1, t2, t5, t6
z	0.017	3	z2, z3, z5, z7, z8
o3	0.019	3	o31, o32, o33, o34, o36
clwc	0.014	2	clwc2, clwc4, clwc5, clwc6, clwc7, clwc8
cc	0.011	2	cc1, cc2, cc4, cc5, cc6, cc7
ciwc	0.008	1	ciwc2, ciwc3, ciwc4, ciwc5, ciwc6, ciwc7, ciwc8

Table 6.17: Highly correlated variable groups. Only groups with more than one variable is displayed. "Variable" column indicates the selected variable, and its highly correlated (>0.8) variables are displayed in "Highly correlated variables."

Variables	Highly correlated variables
d5	d2, d3
r4	cc8, q7
clwc4	cc3
w6	w1


Training for pv
Training for z
Training for t
Training for q
Training for w
Training for vo
Training for d
Training for u
Training for v
Training for r
Training for o3
Training for clwc
Training for ciwc
Training for cc

2007

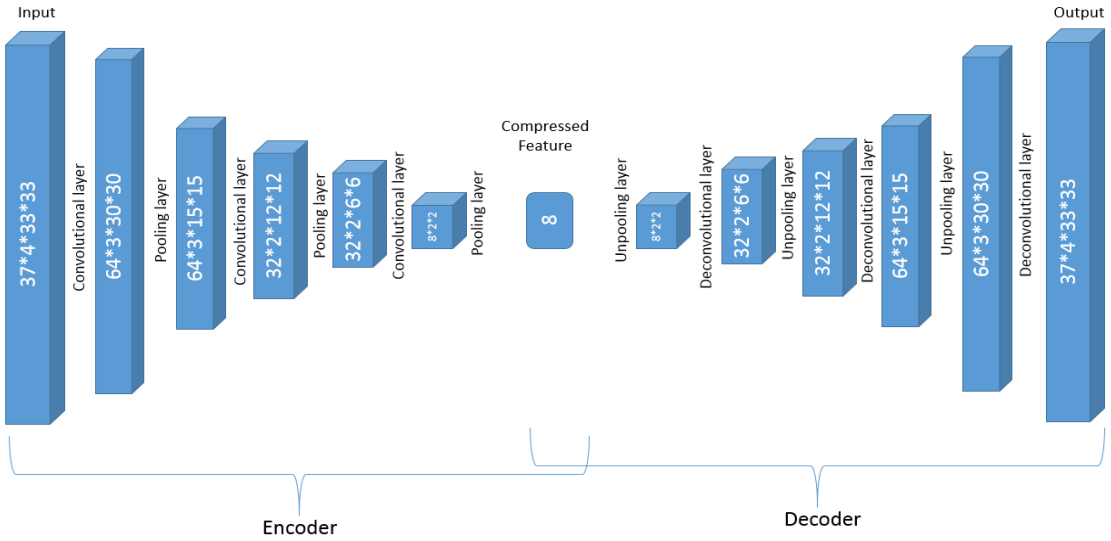
Figure 6.7: Network training loss over iterations for pv, z, t, q, w, vo, d, u, v, r, o3, clwc, ciwc, cc from left to right and from top to bottom. The 14 graphs represent the training loss change for each variable respectively. The y-axis represents the training loss, and decreasing from top (the maximum loss value of the variable) to bottom (0). The x axis represents the iterations, and increasing from left (0) to right (200).

These variables are categorized into 3 classes in Table 6.18 based on their summed importance score described in Table 6.16, and their new structures are displayed in Figure 6.8. The dimension of the compressed feature (num) equal to 8, 4, and 2 in Table 6.18 corresponding to (a), (b), and (c), respectively in Figure 6.8. Therefore, the auto-encoder network for each variable is retrained with the new structure, i.e., Figure 6.8 (a) for variables pv, q, r, u, v, and vo, Figure 6.8 (b) for variables, w, d, t, Figure 6.8 (c) for z, o3, cc, ciwc, and clwc.

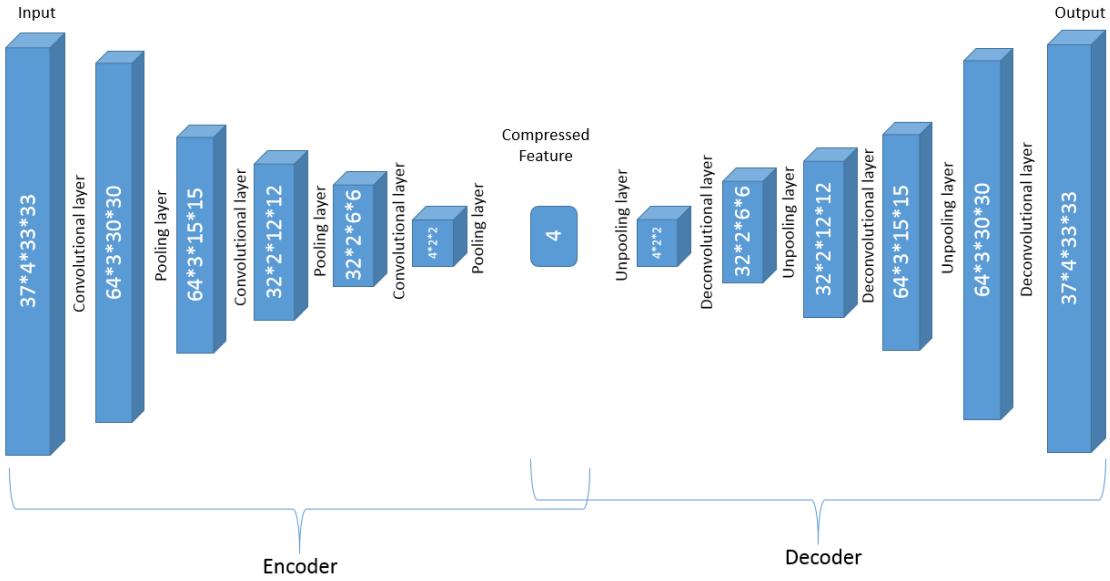
Table 6.18: Dimensions of the compressed features of auto-encoder after tuning based on the summed importance score described in Table 8 for each of the 14 variables.

Importance sum	Dimension of the compressed feature (num)
Less than 0.02	2
0.02 to 0.045	4
Above 0.045	8

(a)


Input
37*4*33*33
Convolutional layer
64*3*30*30
Pooling layer
64*3*15*15
Convolutional layer
32*2*12*12
Pooling layer
32*2*6*6
Convolutional layer
8*2*2
Pooling layer
Compressed Feature
8
Unpooling layer
8*2*2
Deconvolutional layer
32*2*6*6
Unpooling layer
32*2*12*12
Deconvolutional layer
64*3*15*15
Unpooling layer
64*3*30*30
Deconvolutional layer
Output
37*4*33*33
Encoder
Decoder

(b)


Input
37*4*33*33
Convolutional layer
64*3*30*30
Pooling layer
64*3*15*15
Convolutional layer
32*2*12*12
Pooling layer
32*2*6*6
Convolutional layer
4*2*2
Pooling layer
Compressed Feature
4
Unpooling layer
4*2*2
Deconvolutional layer
32*2*6*6
Unpooling layer
32*2*12*12
Deconvolutional layer
64*3*15*15
Unpooling layer
64*3*30*30
Deconvolutional layer
Output
37*4*33*33
Encoder
Decoder

(c)

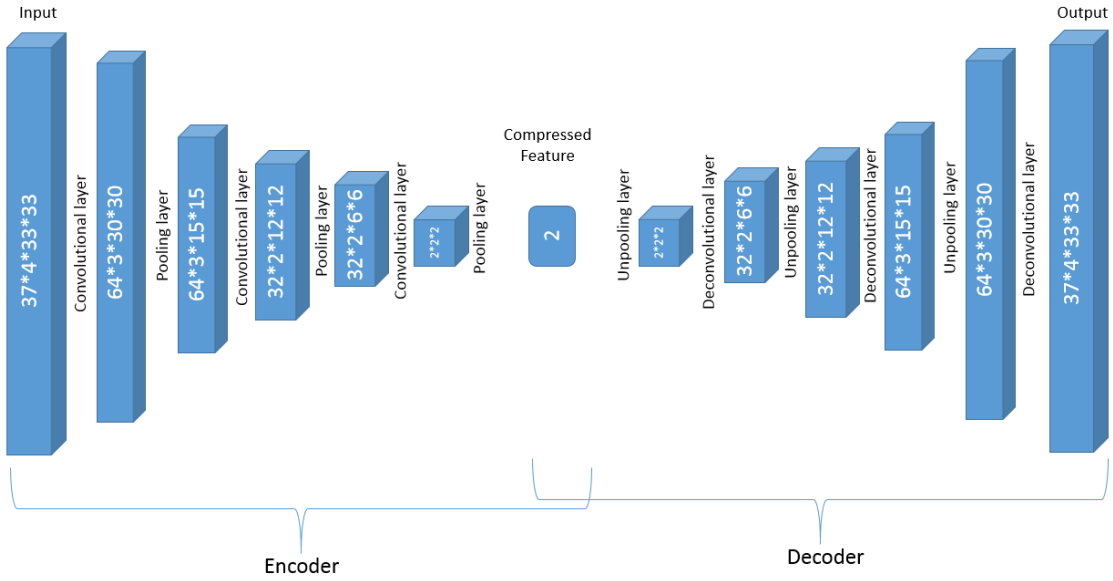


Figure 6.8: Structure for adjusted auto-encoder network.

After the new structure for DL Interim filter is derived, we retrain each network for 200 times, and the training loss is very similar with those shown in Figure 6.7 so their training loss graph is not displayed. We also find that all the networks are converged in 100 iterations. As same as being processed above, pv2, pv4, pv5, q1, u4, u7, v1, v2, vo2, vo7, w4, d1, d2, d4, t2, t3, t4, z1, and o32 are zero features, i.e., contain all zeros, are removed. cc2 is highly correlated (>0.8) with cc1, and w2 and w3 are highly correlated with w1. Hence clwc1, w2, and w3 are removed. The remained 48 (6*8+3*4+5*2-22) features are concatenated with the filtered SHIPS variables, and are used as the input to the GMM-SMOTE.

6.3.1.2 The number of clusters selected in GMM-SMOTE

After the hyperparameters in data filters are tuned, the hyperparameters for GMM-SMOTE and XGBoost still need to be tuned for the best results. Similar to the LLE-SHIPS model, the BIC values with the different number of clusters are displayed in Figure 6.9, n_cluster is selected as 3 with the smallest BIC value.

The clustering result is displayed in Table 6.19 with the numbers of minority (RI) and total instances, and the IIR in each cluster. As we defined danger clusters with 0.2-5 IR range, Clusters 1, 2, and 3 are all included in the following augmentation.

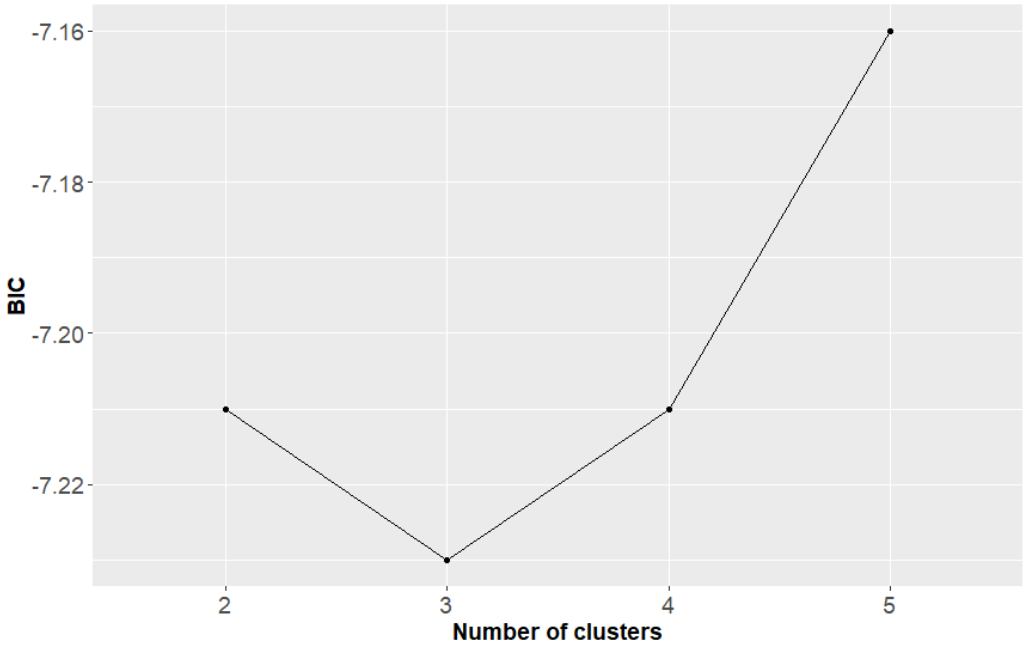


Figure 6.9: BIC (10^6) for GMM with different number of clusters.

Table 6.19: Number of minority, total cases, and the IIR (with population RI ratio at 5.1%) for the 3 clusters generated by GMM.

Cluster	1	2	3	Total
Number of the minority instance	209	36	258	523
Number of the total instance	3222	2645	4318	10185
Imbalance Rate	1.297	0.272	1.195	1

6.3.1.3 Hyperparameters tuning for GMM-SMOTE and XGBoost

Similar to LLE-SHIPS model, Figure 6.10 shows the 10-fold cross-validation kappa scores on the training-validation dataset change over a total 40 BO iterations. Similarly, since the trend with the iteration is unpredictable, hyperparameter sets with the best 5 kappa scores are selected, and their performance and hyperparameter values are displayed in Table 6.20 with the same M*X* notation.

The top 5 performed hyperparameter sets are M23 (0.516), M25 (0.506), M27 (0.502), M31 (0.501), and M21 (0.498). Similarly, the ranking for conservativeness among the five groups for individual hyperparameters are listed in Table 6.21. The total conservativeness scores are also calculated, which are 17, 25, 23, 32, and 32 respectively for M21, M31, M27, M25, and M23. Since our goal is to choose a model neither conservative nor aggressive, the parameter set M31 with the middle conservativeness ranking score is chosen for following implementation and discussion.

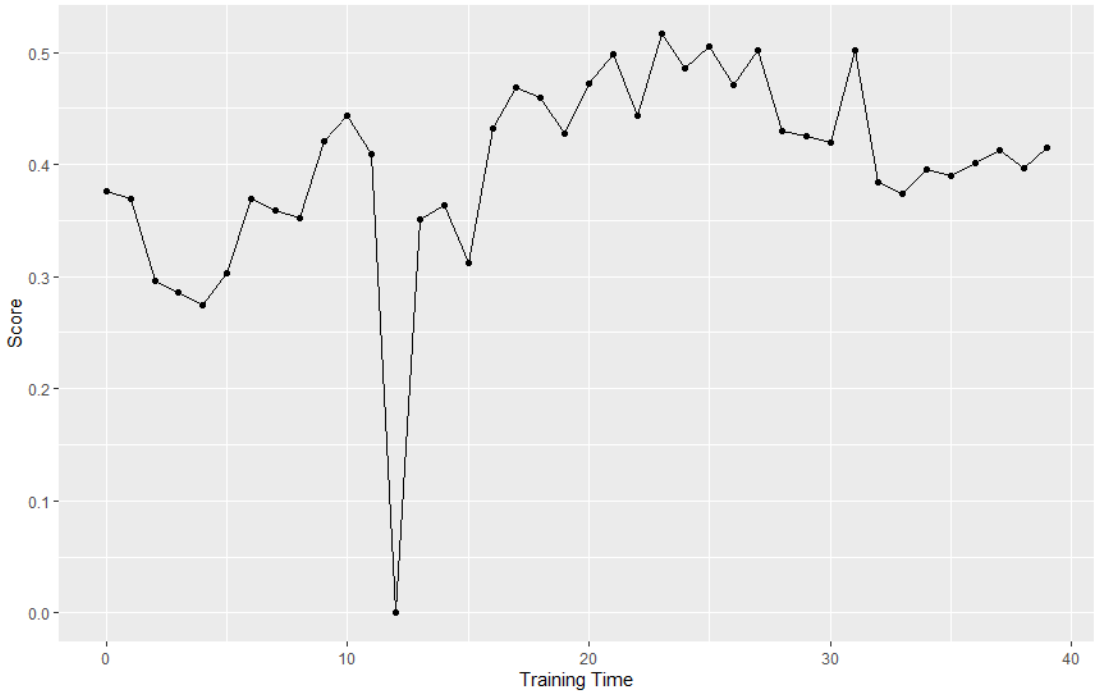


Figure 6.10: Variation of Cross-validation kappa scores over Bayesian Optimization iteration numbers.

Table 6.20: Top performed hyperparameter sets, the corresponding cross-validation kappa scores, and specific values of the tuned hyperparameters. The numbers after "M" denoting the iteration numbers.

Name	M21	M31	M27	M25	M23
Kappa score	0.498	0.501	0.502	0.506	0.516
m_neighbors	10	10	10	10	9
k_neighbors	14	9	14	7	8
shrinkage	0.23	0.19	0.15	0.12	0.15
n_estimators	2000	2000	2000	1088	1603
subsample	0.50	0.50	0.50	1.00	0.88
colsample_bytree	1.00	1.00	1.00	0.50	0.82
reg_alpha	0.50	0.50	0.50	0.50	0.82
reg_lambda	20.00	20.00	0.50	20.00	18.38
gamma	0	0	0	0	0
min_child_weight	0.50	0.50	0.50	0.50	0.76
max_depth	6	3	3	5	3

Table 6.21: The descending value ranking of individual hyperparameter among the top 5 performed cases, and the corresponding conservativeness ranking scores in parentheses. The parameters with normal font are those favoring smaller values for conservativeness, and those *Italicized* favoring larger values.

Name	M21	M31	M27	M25	M23
m_neighbors	*1 (2)*	*1 (2)*	*1 (2)*	*1 (2)*	*5 (1)*
k_neighbors	1 (1)	3 (3)	1 (1)	5 (5)	4 (4)
shrinkage	1 (1)	2 (2)	3 (3)	5 (5)	3 (3)
n_estimators	1 (1)	1 (1)	1 (1)	5 (5)	4 (4)
subsample	3 (3)	3 (3)	3 (3)	1 (1)	2 (2)
colsample_bytree	1 (1)	1 (1)	1 (1)	5 (5)	4 (4)
reg_alpha	*2 (1)*	*2 (1)*	*2 (1)*	*2 (1)*	*1 (5)*
reg_lambda	*1 (3)*	*1 (3)*	*5 (1)*	*1 (3)*	*4 (2)*
gamma	*1 (1)*	*1 (1)*	*1 (1)*	*1 (1)*	*1 (1)*
min_child_weight	*2 (1)*	*2 (1)*	*2 (1)*	*2 (1)*	*1 (5)*
max_depth	1 (1)	5 (5)	5 (5)	2 (2)	5 (5)
Total score	16	23	20	31	36

6.3.1.4 Hyperparameters tuning for XGBoost

Similarly, to tune the decision threshold, Figure 6.11 (a) displays variations of precision and POD variations as functions of the decision threshold from 10-fold cross-validation in the training/validation data. The precision and POD curves cross each other around 0.2 of the threshold value, a relatively balanced point for POD and FAR. At the same point, the kappa scores shown in Figure 6.11 (b) is closer to the highest value, 0.61. As a result, 0.2 is selected as the decision threshold as before.

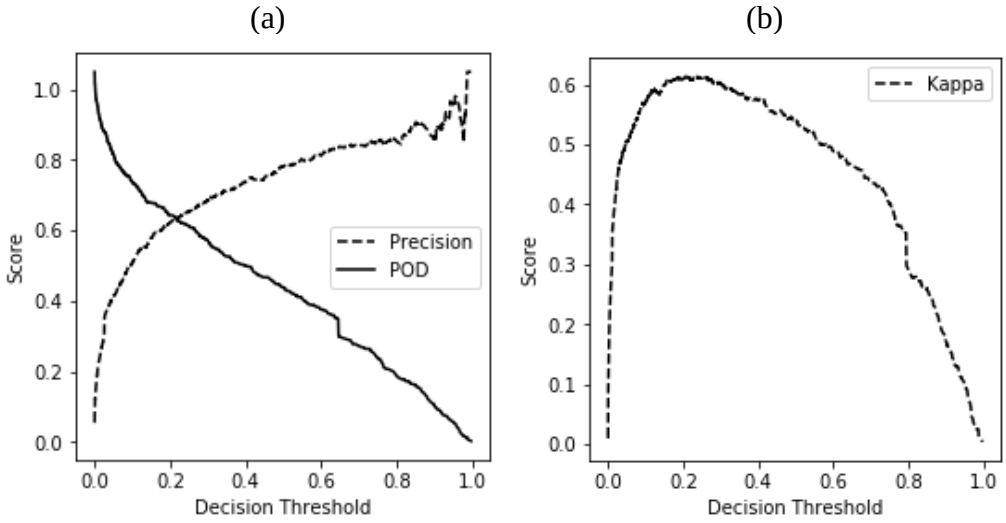


Figure 6.11: (a) Precision and POD score vs. decision threshold. (b) Kappa score vs. decision threshold

6.3.2 Model result on test data

Similar to LLE-SHIPS model, the evaluation of the prediction for DL-SHIPS model is on the test data only. The test confusion matrix for the model, before hyperparameter tuning (MB), and after hyperparameter tuning (MA) is displayed in Table 6.22. The result indicates that the hyperparameter tuning procedure does help the model performance.

Table 6.22: Confusion matrix values after (before) hyperparameter tuning with the test data.

	Predicted RI	Predicted non-RI	Actual
Actual RI	48 (29)	47 (66)	95
Actual non-RI	37 (31)	1465 (1471)	1502
Total Predicted	85 (60)	1512 (1537)	

Kappa, PSS, POD, and FAR are used for the model evaluation, and their values for MB and MA are elaborated in Table 6.23. The POD and FAR values for MB and MA cases demonstrated the importance of hyperparameter tuning. After tuning, POD increases 65.6% from 0.305 to 0.505, while FAR decreases from 0.517 to 0.435, 15.9%. The overall statistics PSS and kappa score also increased from 0.285 to 0.481 (68.8%) and from 0.344 to 0.506 (47.1%), respectively, confirming the significant improvement on RI prediction with the hyperparameter tuning procedure, and apparently, the model was overfitted before tuning process with so many variables.

Table 6.23: Performance comparisons. MB and MA denote the models before and after the hyperparameters in GMM-SMOTE and XGBoost are tuned.

Model	Kappa	PSS	POD	FAR
MB	0.344	0.285	0.305	0.517
MA	0.506	0.481	0.505	0.435
Improvement MB	47.1%	68.8%	65.6%	-15.9%

6.3.3 Feature importance

Similar to LLE-SHIPS model, the importance score could be derived from XGBoost for the output of the data filters. However, in the DL-SHIPS model, since there are even significantly more variables (each grid in each variable could be regarded as a feature) than that of the input for the LLE data filter, it is even more computationally expensive and impossible to implement the same feature importance evaluation approach

(permutation) as in LLE-SHIPS model, i.e., tracing back the importance of each ERA-Interim feature is almost impossible for DL-SHIPS model. Therefore, although we can evaluate the importance of the output of data filters, how to evaluate the contribution from each of the original ERA-Interim variables is a notoriously difficult task for deep learning networks, a.k.a, autoencoder network. Here we are roughly evaluating the importance of the ERA-Interim variables by calculating their summed importance score derived from the XGBoost classifier for each of the 14 ERA-Interim variables, as well as the averaged individual score for parameters associated with each variable. And for the variables with higher importance score, the feature level information from the individual 3D auto-encoder are traced back based on the feature map (Zeiler 2014), where the extracted information, for example, the geometric location, is visualized.

Table 6.24 displays the 10 most important variables among the 120 selected variables, including not highly correlated 72 SHIPS variables, and 48 variables extracted from the DL ERA-interim data filter. We can find that among the top 10, there are six SHIPS variables and four DL variables, and the total importance score for DL variables is 0.4119, while that of SHIPS variables is 0.5881. Therefore, the average score per SHIPS variable/ERA variable is 0.0082/0.0086. The fact that the average score for the SHIPS variable is less than that of ERA variables indicates that the ERA-Interim data filter has a similar importance score comparing to that of SHIPS variables; hence DL ERA-interim data filter is working efficiently.

Table 6.24: Variable importance in DL-SHIPS model.

Variable	Importance	Description
BD12	0.019747	The past 12 hour intensity change
VMAX	0.017600	Maximum Surface Wind
SHRD	0.014810	850-200 hPa shear magnitude
DTL	0.014381	The distance to nearest major land
IRM1_5	0.013737	Predictors from GOES data (not time dependent) for r=100-300 km but at 1.5 hours before initial time
o31	0.013308	3rd variable in o3
G150	0.013093	Temperature perturbation at 150 hPa due to the symmetric vortex calculated from the gradient thermal wind. Averaged from r=200 to 800 km centered on input lat/lon (not always the model/analysis vortex position) (deg C*10)
q7	0.013093	7th variable in q
u3	0.012878	3rd variable in u
q4	0.012878	4th variable in q

Table 6.24 also indicates that BD12 has the largest importance score, 0.0197, and the second most important variable is VMAX. The third and fourth most important variables, SHRD and DTL. The fifth to tenth variables are IRM1_5, o31, G150, q7, u3, and q4, and four of them are derived from the DL-interim data filter. o31 is the first variable extracted from o3's network, while q7, u3, and q4 are the seventh, third, and fourth variables of q, u, and q. The importance scores of all of the 120 variables can also be found in Table A6.

Since the 3D auto-encoder model structure is different over 14 ERA-Interim variables, the summed importance score, which is the sum over all the output from the same network, for example, the summed importance score for r is the sum of the importance score over r1, r2, …, r8, for each of the 14 variables, as well as the averaged

importance score on the non-zero, non-highly-correlated features. For example, the averaged importance score for r is the summed importance score for r divided by 8, are described in Table 6.25. Based on Table 6.25, we can find that q, vo, and u are scored in the top 5 in terms of both summed score and the average score. Therefore, below we are looking at feature maps from the first layer of the networks for q, vo, and u between an example RI and non-RI instances to roughly estimate what plays a more significant role to distinguish between RI and non-RI instances.

Table 6.25: Summed variable importance score, the number of non-zero, non-correlated features, the feature-wise averaged importance score, and its ranking for each ERA-Interim variable.

Variable	The number of features	Summed Importance Score	Importance score rank	Average importance score	Average importance score rank
q	7	0.062	1	0.010843	3
vo	6	0.055	2	0.010583	4
u	6	0.055	3	0.00975	5
v	6	0.056	4	0.008483	7
pv	5	0.050	5	0.00882	6
r	8	0.049	6	0.004838	14
ciwc	2	0.042	7	0.0072	10
o3	1	0.039	8	0.0133	1
cc	2	0.020	9	0.006	12
d	1	0.017	10	0.0118	2
t	1	0.019	11	0.0082	8
z	1	0.014	12	0.0077	9

w	1	0.011	13	0.0071	11
clwc	1	0.008	14	0.0058	13

Figure 6.12 and Figure 6.13 represent examples of non-RI and RI instances featutres extracted by the 3D autoencoder in all 3 channels in terms of variable relative humidity (q), and based on the figures we can find that the extracted feature maps are sparse, with only 22/64, and 24/64 non empty feature maps, where the deep blue feature maps represents all the pixels have value 0, for non-RI and RI instances respectively. With the limited available information, we can find the non-RI instances are extracting features from the northeast (upper left) of the center or the whole domain, while RI instances are extracting features mainly on the southeast (bottom right) of the center. So we can conclude that the relative humidity (q) in northeast of the center is more important in RI occurance.

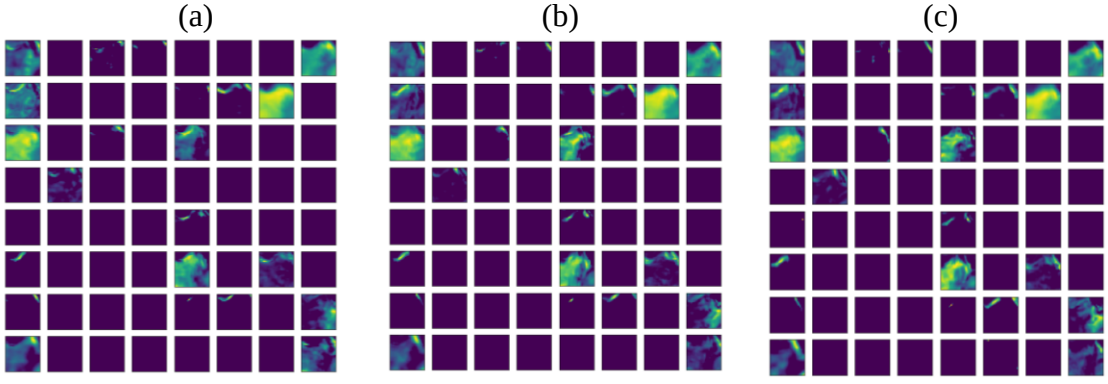


Figure 6.12： 3 channels, 64 feature maps for the first layer (dimension: 3 (channel) * 64 (feature map) * 30 (feature map dimension) * 30 (feature map dimension)) of the network that is immediate after the input layer (dimension: 37 (pressure level) *

4 (-18h, -12h, -6h, and 0h) * 33 (vertical grid) * 33 (horizontal grid)) for variable q with its network structure displayed in Figure 6.8a. This is an example for a non-RI instance, and (a) Non-RI in channel 1. (b) Non-RI in channel 2. (c) Non-RI in channel 3, and the sequence of the channel does not matter. In each channel, there are 64 (8 in the row and 8 in the column) feature maps, and each feature map has 30 (pixels) * 30 (pixels) dimension. Deep blue implies the value in that pixel is 0, and the brighter the color is, the high the value in that pixel.

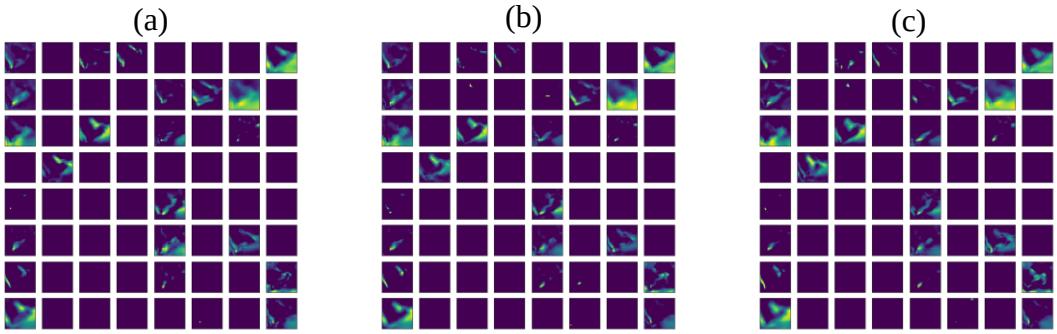


Figure 6.13：Same as Figure 6.12 but for a RI instance (a) RI in channel 1. (b) RI in channel 2. (c) RI in channel 3.

Figure 6.14 describes the examples of non-RI instances and Figure 6.15 describes the RI features extracted by the 3D autoencoder in all 3 channels in terms of variable relative vorticity (vo), and we can find that the feature map is even more sparse comparing to variable relative humidity (q), with only approximately 7/64, and 7/64 non empty feature maps for both situations. Among them, 5/7 feature maps are towarding north (top; of the center) and the rest 2 feature maps are towarding south (bottom; of the center). In comparison, RI instance indicates that 5 are towarding south (lower) and 2 are towarding north (of the center). So we can conclude that relative vorticity (vo) in the south of the center are more important in RI occurance.

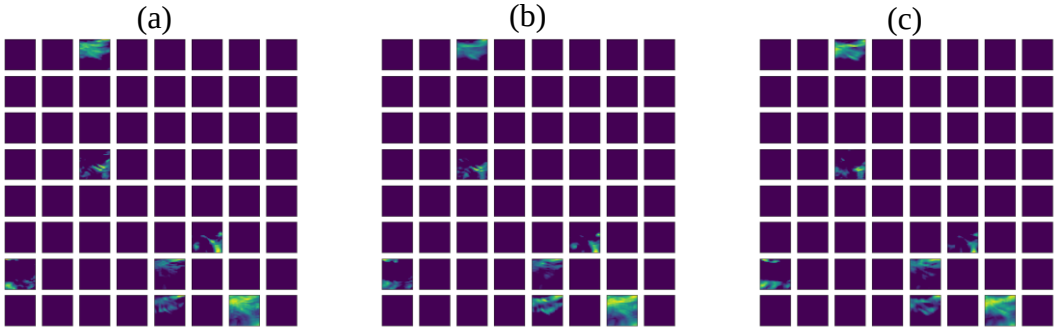


Figure 6.14：Same as Figure 6.12 but for variable vo with its network structure in Figure 6.8a in a non-RI instance: (a) non-RI in channel 1. (b) non-RI in channel 2. (c) non-RI in channel 3.

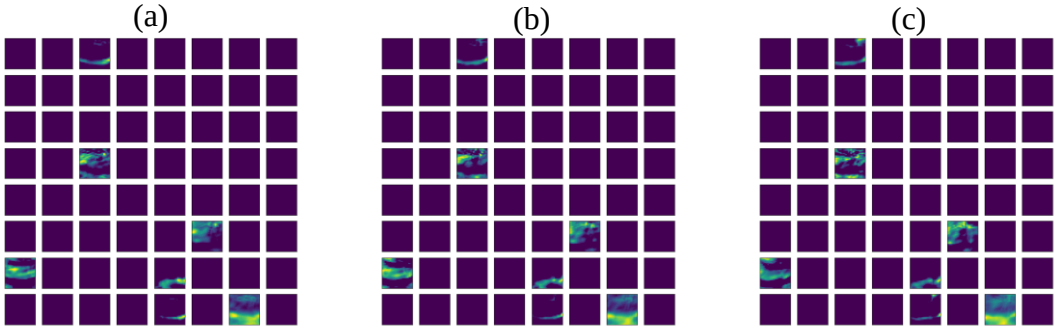


Figure 6.15：Same as Figure 6.12 but for variable vo with its network structure in Figure 6.8a in 3 channels in a RI instance: (a) RI in channel 1. (b) RI in channel 2. (c) RI in channel 3.

Similarly, based on Figure 6.16 and 6.17, 11/64 and 11/64 are non-empty feature maps for non-RI and RI instances in variable eastward wind (u), non-RI instance has 6, 1, and 4 feature maps toward south, north, and east of the center. In comparison, RI instance has 6, 4, and 1 feature maps concentrating north, south, and east of the center, which indicated that eastward wind (u) in the north of the center is more possible to result in RI.

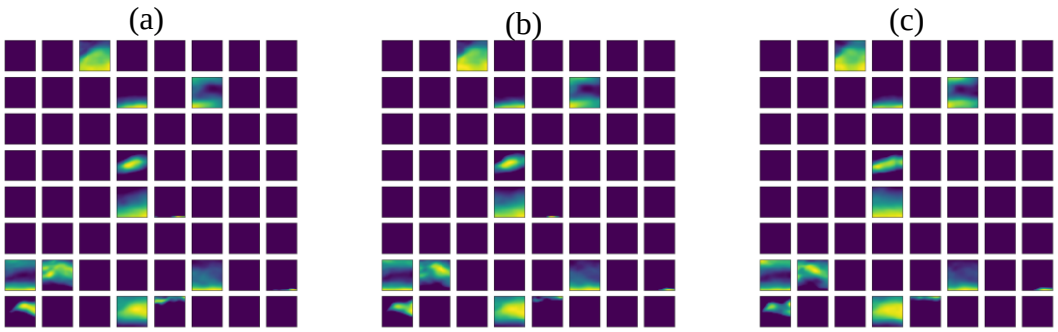


Figure 6.16： Same as Figure 6.12 but for variable u with its network structure in Figure 6.8a in a non-RI instance: (a) non-RI in channel 1. (b) non-RI in channel 2. (c) non-RI in channel 3.

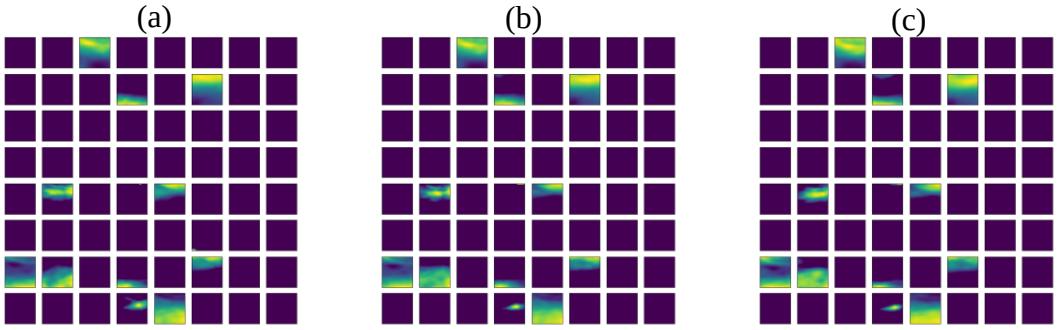


Figure 6.17： Same as Figure 6.12 but for variable u with its network structure in Figure 6.8a in 3 channels in a RI instance: (a) RI in channel 1. (b) RI in channel 2. (c) RI in channel 3.

6.4 Model performance comparison

Two works, i.e., the best model in Y16 and KRD15, which outperforms almost all of the other works in the RI prediction, are used to compare with the performance of the COR-SHIPS model, LLE-SHIPS model, and DL-SHIPS model.

6.4.1 Model performance in Yang (2016) and Kaplan et al. (2015)

In Y16, experiments are conducted using C4.5 decision tree (Quinlan, 1993), alternating decision tree (ADTree; Freund and Mason, 1999), random forest (Breiman, 2001), classification and regression tree (CART; Breiman et al. 2017), logistic model tree (LMT; Landwehr et al. 2005), the repeated incremental pruning to produce error reduction (RIPPER; Cohen 1995), function-based classification such as support vector machines with sequential minimal optimization (SMO; Platt 1999), naïve Bayes scheme (Tan et al. 2015), and the decision tree with naïve Bayes classifiers at the leaves (NBTree; Kohavi 1996) with cost ratio 4.6 to predict RI. Figure 6.18 shows the best performed two classifiers C4.5 decision tree and ADTree, where measures in Figure 6.18 is defined as the different groups of variables selected by different variable selection criteria. The performances of ADTree over all measures are more robust since its kappa scores over different dataset is more stable than that of C4.5 decision tree, although the best test kappa (27.5%) is achieved by C4.5 decision tree.

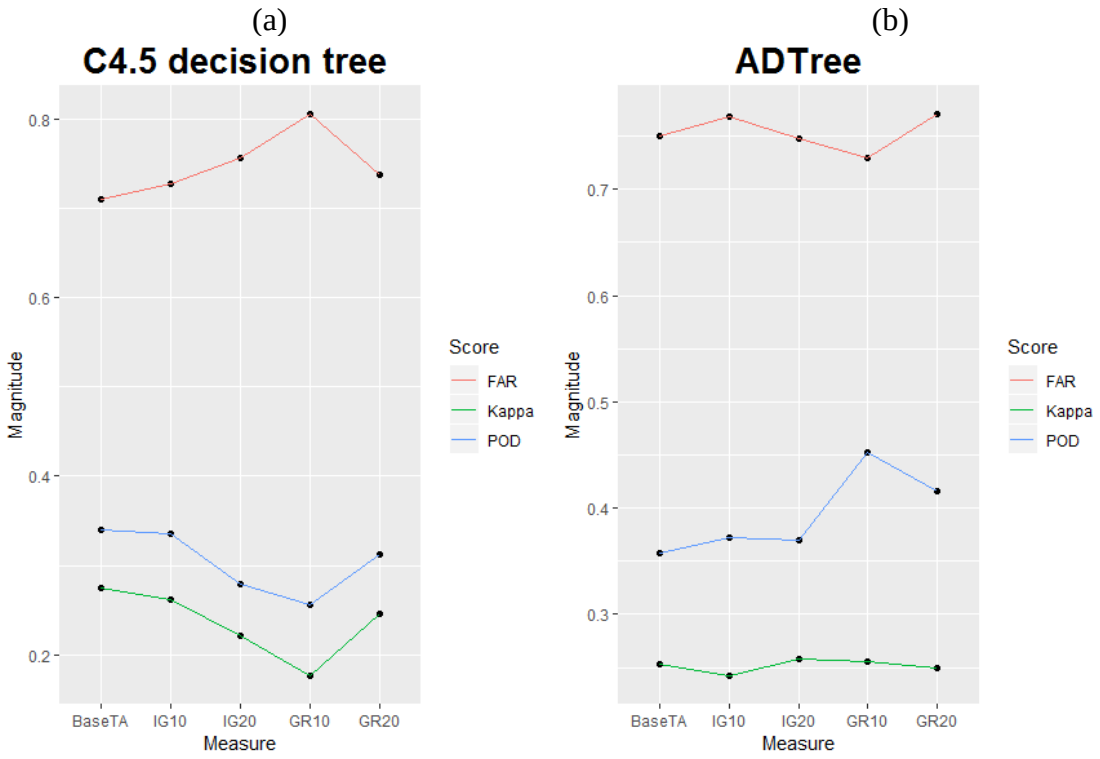


Figure 6.18: Kappa, POD, and FAR for (a) C4.5 decision tree. (b) ADTree. Data are from Y16.

In KRD15, CON-RII outperforms all other operational models, i.e., the 5-day SHIFOR model (SHF5; Knaff et al. 2003), the decay version of the SHIPS model (DSHP; DeMaria et al. 2005), the logistic growth equation model (LGEM; DeMaria 2009), the Geophysical Fluid Dynamical Laboratory (GFDL) hurricane prediction model early (GFDI) and late (GFDL) versions (Kurihara et al. 1998) and the Hurricane Weather Research and Forecasting Model early (HWFI) and late (HWRF) versions (Tallapragada et al. 2014), and the NHC official forecast (OFCL) in RI prediction at the threshold 30 knots in the 24-hour lead-time in terms of Peirce's skill score (PSS) with approximately 0.225 for TC cases in 2008–13. The result details of all the models are presented in

Figure 6.19, the CON-RII with PSS 0.225 performs little over two times than the second best-performed model (OFCL).

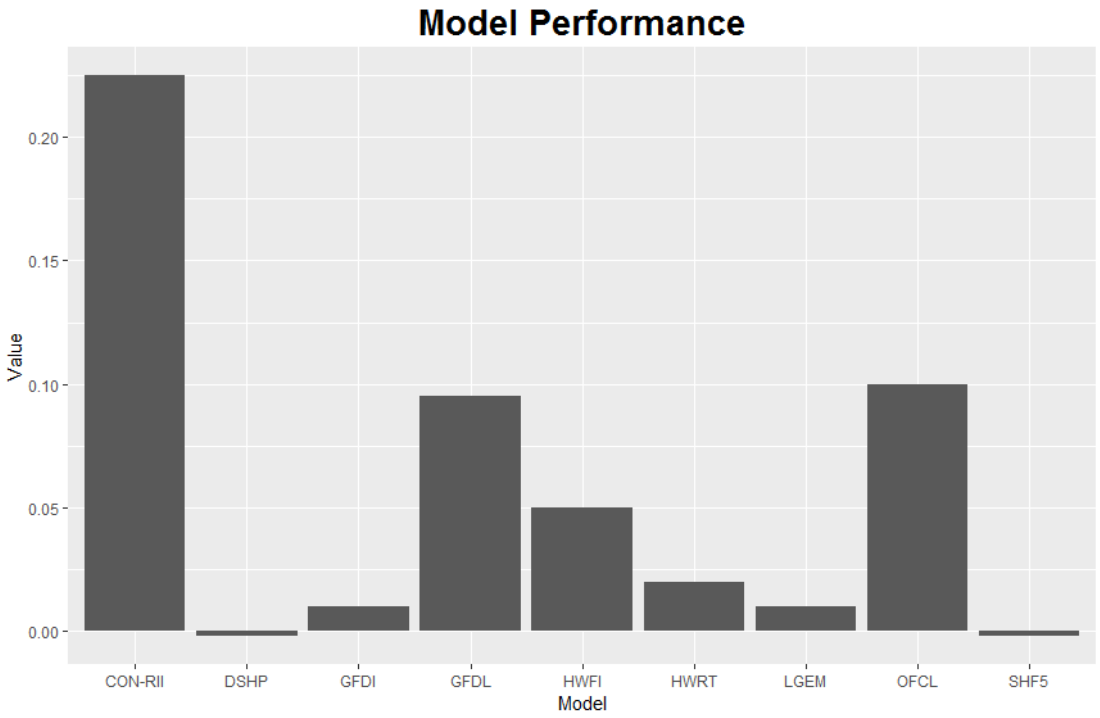


Figure 6.19: Different model's performance regarding Peirce's skill score (PSS) based on data from KRD15.

6.4.2 Model comparison

Figure 6.20 displays the model performance comparison between the best model in Yang (2016) (a.k.a. Y16) and the three newly developed models in this study for kappa score, POD, and FAR. The performance of the COR-SHIPS model, LLE-SHIPS model, and DL-SHIPS model is significantly better than that of the best model in Yang (2016).

Details are listed in Table 6.26, and we can find that the performance (kappa score, POD, and FAR) improvement by using the entire SHIPS database with regard to Y16 is medium with 28.7%, 20.9%, and -12.7%, respectively. If we use ERA-Interim data in addition to the SHIPS database, we achieve significant improvement by at least 65.1%, 23.8%, and -20.8% in terms of kappa score, POD, and FAR (the smaller, the better).

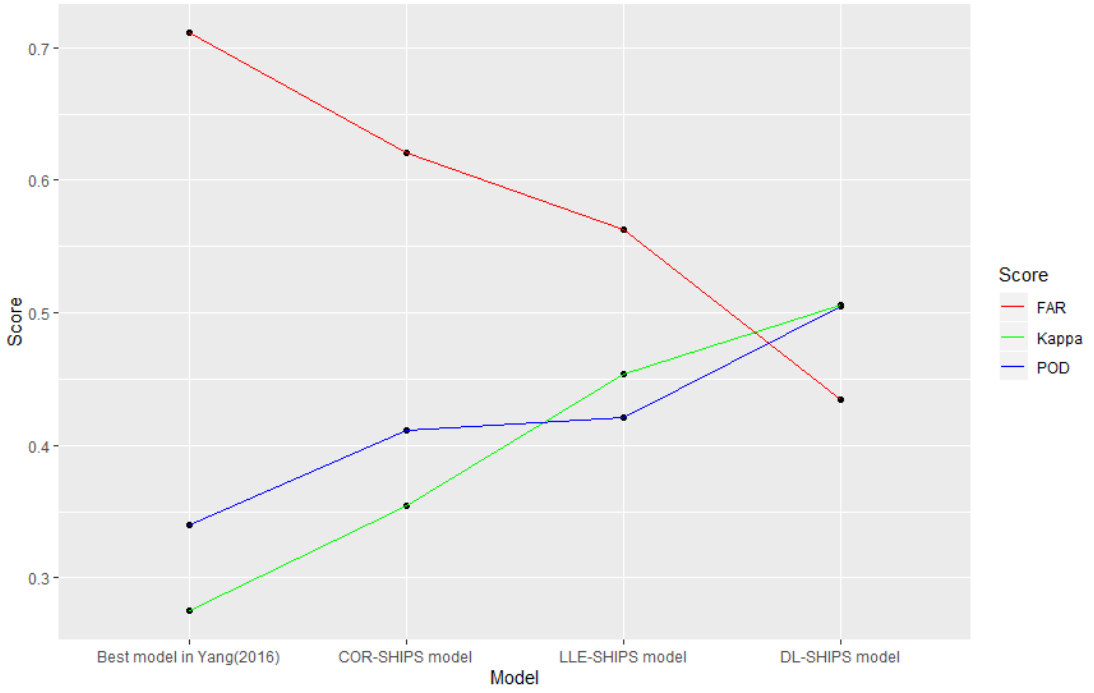


Figure 6.20: Model performance comparison: Model's test kappa, FAR, and POD score in the best model in Yang (2016), SHIPS model, LLE-SHIPS model, and DL-SHIPS model.

Table 6.26: Performance comparison between our models, and Y16 and KRD15.

Model	Kappa	PSS	POD	FAR	Improvement Yang			Improvement KRD15		
					Kappa	POD	FAR	PSS	POD	FAR
COR-SHIPS	0.354	0.368	0.411	0.621	28.7%	20.9%	-12.7%	63.6%	49.5%	-24.7%

LLE-SHIPS	0.454	0.399	0.421	0.563	65.1%	23.8%	-20.8%	77.3%	53.1%	-31.8%
DL-SHIPS	0.506	0.481	0.505	0.435	84.0%	48.5%	-38.8%	114.0%	83.6%	-47.3%
Y16	0.275	NA	0.340	0.711						
KRD15	NA	0.225	0.275	0.825						

Figure 6.21 displays the model performance comparison among CON-RII in Kaplan et al. (2015) (a.k.a. KRD15), COR-SHIPS model, LLE-SHIPS model, and DL-SHIPS model in terms of PSS, POD, and FAR. The performance of the SHIPS model, LLE-SHIPS model, and DL-SHIPS model is significantly better than that of KRD15 as at least 63.6%,49.5%, 24.7% improvement in PSS score, POD, and FAR, and more details are elaborated in Table 6.26. The performance improvement in KRD15 for our 3 models are more significant than that in Y16, especially we are considering that the performance of the model in KRD15 is evaluated in the training dataset, and the performance in Y16 and our models are evaluated based on the test dataset.

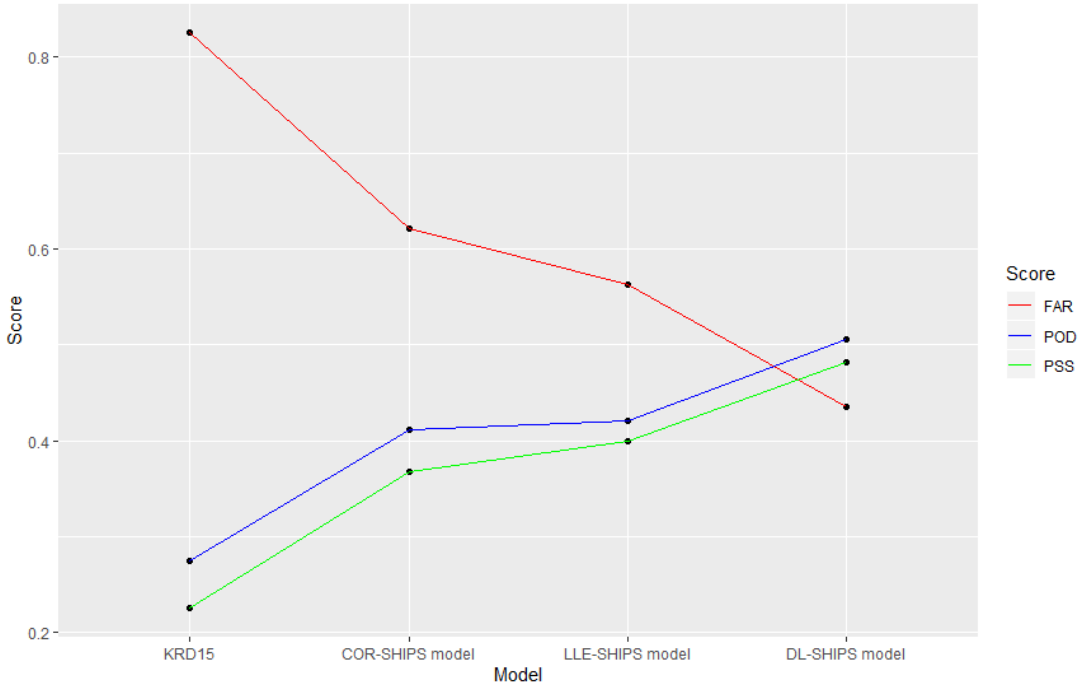


Figure 6.21: Model's test PSS, FAR, and POD score in KRD15 (Kaplan et al., 2015), COR-SHIPS model, LLE-SHIPS model, and DL-SHIPS model.

The performance comparison between our 3 models – COR-SHIPS, LLE-SHIPS, and DL-SHIPS are elaborated in Table 6.27. We can find a moderate improvement was made from the COR-SHIPS model to the LLE-SHIPS model in terms of kappa, PSS, POD, and FAR of 28.2%, 8.4%, 2.4%, and 9.3%, which indicates that most of the near center information has already been explored/extracted by variables in SHIPS database. In comparison, a significant change is made from the COR-SHIPS model to the DL-SHIPS model in terms of kappa, PSS, POD, and FAR of 42.9%, 30.7%, 22.9%, and 30.0%, which indicates that the DL-SHIPS model catches large-scale information not only as averages (from SHIPS database) but also the variations (from DL model). With

almost half (48 vs. 90) variables generated from ERA-Interim data filter, the DL-SHIPS model still outperforms the LLE-SHIPS model significantly, which not only indicates that DL ERA-interim data filter is extracting more efficient information, but also shows that using large-scale ERA variables provides much more information than incorporating near core ERA variables only in RI prediction.

Table 6.27: Performance comparison between 3 models developed in this study, 'X' in the table indicates that not available value.

Model	Kappa	PSS	POD	FAR	Improvement COR-SHIPS				Improvement LLE-SHIPS			
					Kappa	PSS	POD	FAR	Kappa	PSS	POD	FAR
COR-SHIPS	0.354	0.368	0.411	0.621	X	X	X	X	X	X	X	X
LLE-SHIPS	0.454	0.399	0.421	0.563	28.2%	8.4%	2.4%	-9.3%	X	X	X	X
DL-SHIPS	0.506	0.481	0.505	0.435	42.9%	30.7%	22.9%	-30.0%	11.5%	20.6%	20.0%	-22.7%

6.5 Feature importance

6.5.1 Feature importance comparison between COR-SHIPS model, LLE-SHIPS model, and DL-SHIPS model

Table 6.28 lists the 36 most important variables for each of the three models. Learning from Table 6.28, 7 out of top 10 in the LLE-SHIPS model is in the top 10 of the COR-SHIPS model, and in all the SHIPS variables identified in top 10 in the DL-SHIPS model are included in that of the COR-SHIPS model and LLE-SHIPS model. This indicates that all three models identify similar important variables.

Table 6.28: Top 36 most important variables in COR-SHIPS model, LLE-SHIPS model, and DL-SHIPS model.

Rank	COR-SHIPS model variable	COR-SHIPS model variable Importance Score	LLE-SHIPS model variable	LLE-SHIPS model variable Importance Score	DL-SHIPS model variable	DL-SHIPS model variable Importance Score
1	BD12	0.0362	BD12	0.0188	BD12	0.0197
2	DTL	0.0217	VMAX	0.0167	VMAX	0.0176
3	CFLX	0.0207	DTL	0.0138	SHRD	0.0148
4	SHDC	0.0206	SHRD	0.0130	DTL	0.0144
5	G150	0.0205	TWXC	0.0115	IRM1_5	0.0137
6	jd	0.0204	G150	0.0113	o31	0.0133
7	VMAX	0.0199	VMPI	0.0113	G150	0.0131
8	IRM1_5	0.0199	REFC	0.0113	q7	0.0131
9	PW08	0.0191	TGRD	0.0111	u3	0.0129
10	VMPI	0.019	IRM1_5	0.0107	q4	0.0129
11	SHTD	0.0187	IR00_12	0.0107	G200	0.0129
12	IR00_12	0.0183	V300	0.0105	vo3	0.0127
13	HE07	0.018	VVAC	0.0105	REFC	0.0124
14	MTPW_2	0.0177	G200	0.0103	vo5	0.0122
15	XD18	0.0177	PEFC	0.0096	vo8	0.0120
16	SHTS	0.0175	MTPW_2	0.0096	PEFC	0.0120
17	PW14	0.0173	XDTX	0.0095	d3	0.0118
18	TWXC	0.0172	PSLV_1	0.0095	CFLX	0.0116
19	R000	0.0168	T150	0.0095	PSLV_3	0.0116
20	V300	0.0167	CFLX	0.0094	T150	0.0114
21	OAGE	0.0165	HIST_2	0.0094	jd	0.0114
22	PSLV_1	0.0162	HE07	0.0092	R000	0.0114
23	Z850	0.0161	SHTS	0.0091	TWXC	0.0112
24	SHRS	0.0161	PSLV_3	0.0089	u8	0.0112
25	SDDC	0.0157	SHTD	0.0085	PW08	0.0112
26	VVAC	0.0156	G250	0.0085	q3	0.0112
27	PSLV_5	0.0156	CD26	0.0085	XDTX	0.0112
28	TGRD	0.0154	lle84	0.0082	CD26	0.0109
29	T150	0.0153	EPSS	0.0082	q8	0.0109
30	CD26	0.0153	R000	0.0077	pv3	0.0107
31	TADV	0.0152	SDDC	0.0076	v4	0.0107
32	V850	0.0151	IRM3_19	0.0075	r1	0.0105
33	PSLV_4	0.0148	RD26	0.0075	u1	0.0101
34	Z000	0.0145	PW08	0.0074	q5	0.0099
35	REFC	0.0145	SHRS	0.0074	IR00_12	0.0099
36	RD26	0.0142	NDTX	0.0074	vo4	0.0099

The 6 common variables are BD12, VMAX, SHRD, DTL, IRM1_5, and G150. Although there is no LLE engineered variables in the top 10 in the LLE-SHIPS model, the importance score for the most important variable, BD12, is only approximately 11% higher than the second important variable VMAX. In comparison, the importance score for BD12 in the COR-SHIPS model has 50% higher importance score than the second most important variable, DTL, which indicates that the performance of the LLE-SHIPS model does not heavily rely on one individual variable. In addition, the performance of the LLE-SHIPS model has approximately 28% higher kappa score than that of the COR-SHIPS model, which indicates that the ERA-interim data filter efficiently extracts important near center features that help the RI prediction.

DL-SHIPS model has 120 variables, which is 42 (33%) variables less than that of the LLE-SHIPS model, and the performance of the DL-SHIPS model (0.506 kappa value) is approximately 11% better than that of the LLE-SHIPS model (0.454). The fact that with much fewer variables, the DL-SHIPS model is performing much better than the LLE-SHIPS model, indicates that DL ERA-Interim data filter extracts large-scale features that is more representative of RI than that of LLE ERA-Interim data filter, i.e., near center feature. However, unlike the LLE-SHIPS model, the DL-SHIPS model extracts features from each variable separately, and interaction between different variables are ignored. If we also extract the interaction between terms in the large-scale dataset, we can get better performance.

Since we have 72 variables in the COR-SHIPS model, we further compare the top 36, i.e., 50% number of variables in the COR-SHIPS model, for the COR-SHIPS model,

LLE-SHIPS model, and DL-SHIPS model that described in Table 6.28. LLE-SHIPS model has 34 SHIPS variables, and 2 ERA-Interim variables, and 25 of 34 variables (73.5%) are overlapped with that of the COR-SHIPS model, which also support the fact that LLE ERA-interim data filter extracts important near center features that help the RI prediction, although the new features seem not as efficient since there are only 2 in the top 36. However, with 162 variables in total, the LLE-SHIPS model has the total importance score 0.259 for the overlap variables, and only with 72 variables, COR-SHIPS model has 0.456 for the same overlap variables, which almost double that of the LLE-SHIPS model. The reason is there are significantly more variables in the LLE-SHIPS model, and it is not surprising that the importance score for the overlapped variables is significantly different in the COR-SHIPS model, and LLE-SHIPS model.

In contrast, the DL-SHIPS model has 19 SHIPS variables, and 17 ERA-Interim variables. Among all these variables, 14 of 19 SHIPS variables (73.7%) are overlapped with that of the COR-SHIPS model, and the summed importance scores are 0.276, and 0.180 respectively for the COR-SHIPS model and DL-SHIPS model, which indicates DL-SHIPS model relies less on SHIPS variables. 18 of 19 SHIPS variables (94.7%) are overlapped with that of the LLE-SHIPS model, and the summed importance scores are 0.199, and 0.231 respectively for the LLE-SHIPS model and DL-SHIPS model, almost same.

With significantly less ERA-Interim variables and almost the same number of SHIPS variables, we can conclude that the DL ERA-interim data filter is efficient at either improving the prediction accuracy, or extracting new variables at the large-scale. There

might be two reasons for this. The first reason might be large-scale features are more efficient than the near center features in RI prediction, which matches domain scientist experience, because most of the SHIPS variables are large scale variables. The second reason might be compared with the LLE data filter, the DL data filter has multiple convolutional layers, which distilled the all levels of large-scale information comprehensively that may be ignored by domain scientists for a long time.

6.5.2 Feature importance comparison between previous studies and this study

A two-side t-test is used for variable selection in KD03, and the RII model was built based on the five variables, DVMX (Intensity change during the previous 12 h), SST, POT (Maximum potential intensity (MPI) – maximum sustained surface wind speed), SHR (850-200-hPa vertical shear averaged from r = 200-800 km), and RHLO (850-700-hPa relative humidity averaged from r=200-800 km), which are found significant in a 99.9% level and with the highest individual RI prediction power. In the first 10 importance variables identified by the COR-SHIPS model, BD12 (ranked 1st), SHRD (4th), VMPI (10th), and VMAX (7th) (POT = VMPI - VMAX) are consistent with the selected variables in KD03. The missed variables in the top ten compared with the top five in KD03 are SST and RHLO. SST is highly correlated with the selected E000, which is listed 52th in the importance ranks. RHLO is highly correlated with RHMD, which is listed 57th in the importance ranks (Table A1 and Table A4).

Compared with variables selected by KD03, in KDK10, SST is removed and 4 additional variables, D200 (time averaged 200-hPa divergence within a 1000-km radius),

OHC (time averaged oceanic heat content), SDBT (STD of GOES-IR BT (t = 0h) within a 50–200-km radius), and PX30 (the percentage area from 50- to 200-km radius covered by IR cloud-top BT of -30ºC or colder), are added. Among the four new variables, D200 is ranked 44th with a 0.0131 importance score (Table A4) in the COR-SHIPS model. The OHC related parameters include COHC, NOHC, and RHCN, and among them, the highest importance score is achieved by COHC, which is highly correlated with CD26 ranked 30th with a score of 0.0153. The PX30 is corresponding to IR00_8, which is highly correlated with IRM1_16 ranked 50th with a 0.0119 score value. The only caught new KRD10 variable in our top ten in the COR-SHIPS model is the SDBT by IRM1_5 (ranked 8th), representing GOES BT STD within the 100-300 km around the TC centers but 1.5 hours before the current time.

KRD15 replaced RHLO with TPW (Percentage of an area with TPW<45 mm within a 500-km radius and ±45º of the upshear SHIPS wind direction (t = 0h)), and PX30 with PC2 (second principal component of GOES-IR imagery within a 440-km radius (t = 0h)), and added 2 new variables, ICDA (Inner-core dry-air predictor (time avg)), and VMX0 (Max sustained wind (t = 0h)), comparing with variable used in KDK10. Among the four new variables, VMX0 is consistent with VMAX, ranked 7th in the COR-SHIPS model importance list. ICDA is not directly included in SHIPS data, but the related parameter found is CFLX, the dry air predictor except for a factor of VMX0 in KRD15, and CFLX is ranked the 3rd in the top 10 parameter list. The definition of TPW is the same as MTPW_19 in the SHIPS, which ranked only 37th with an importance score of 0.014. The PC2 equivalent parameter in SHIPS is PC00, which ranked only the 70th.

In summary, variables used by KD03, KDK10, and KRD15 for RI prediction are mostly caught up with our top 10 variables in the COR-SHIPS model. The missed variable in KD03 is RHLO and SST, and RHLO was actually replaced by TPW later (KRD15), and TPW is ranked 37th in our list, much more important than the RHLO via the highly correlated RHMD at the 57th place. Among the 4 newly added parameters in KDK10, three, OHC, D200, and PX30, are outside the top 10 list. There are several variables in SHIPS representing the OHC. The most important one is found to be climatological OHC via the highly correlated parameter CD26 at the 30th rank. KRD15 mentioned that OHC works well only when the other two variables, POT and ICDA are included in a model. D200 was introduced to SHIPS in 1998 (DeMaria and Kaplan 1999), but it was eliminated in 2001 and added back in 2002 (DeMaria et al. 2005). DeMaria et al. (2005) also found that the role of this divergence in TC intensity forecasting is sensitive to the data sources. Therefore, it is not very unusual if this model did not rank this predictor high. The last parameter not in the top 10 list, PX30, was replaced by PC2 in KRD15. Actually, PC2 is ranked 70th in this study, and it is hard to interpret the result. It is very unfortunate that the GOES-IR principal components were mistreated initially in this work, and we missed the opportunity to rank the importance of other PCs among the first nine PCs. The other missed parameter in KRD15 is the TPW, ranked only 37th. It is plausible that the humidity effects are also reflected in the 3rd ranked parameter CFLX.

Comparing to the COR-SHIPS model, the LLE-SHIPS model not only employs SHIPS parameters but also ERA-Interim near center parameters for predicting RI, and we

divide the ERA-Interim parameters into different groups with highly correlated parameters to evaluate the group importance instead of evaluating the importance for each parameter. Based on the top 5 important groups, we can find that wind speed, especially at 400, 450, and 1000 hPa play a significant role in RI prediction. Another important information we get, potential vorticity at 1000 hPa, although possibly the importance score is over-estimated, is more important than the most important SHIPS variable, BD12. Finally, vertical pressure speed is also found important in RI prediction.

In addition to the LLE-SHIPS model, the DL-SHIPS model, which adopts the large-scale ERA-Interim information, further improves the performance of COR-SHIPS significantly. Relative humidity (q), relative vorticity (vo), and eastward wind (u) are found to be top 3 important variables, but evaluating the contribution from each of the original ERA-Interim parameters is a notoriously difficult task for deep learning networks. So we roughly evaluating their 3D auto-encoder first layer weights, and find out that RI instances tend to have higher weights in southeast humidity (q), north relative vorticity (vo), and north eastward wind (u), where the direction is with regard to the TC center

CHAPTER 7 CONCLUSION AND DISCUSSION

To improve RI prediction with modern techniques, this study constructs a well-tailored artificial intelligence (AI) system that goes back to the SHIPS database, the most complete dataset with parameters known to be related to TC intensity changes, as well as the ERA-Interim dataset, the best reanalysis product at the moment, to extract information from a more complete set of variables. This system consists of four major components, data filters to remove variables unrelated to RI, reduce variables among highly correlated variables, screen out variables with high missing value rates, and engineer/extract a reduced set of variables from the high dimensional variable space; a customized sampler to upsample the minority (RI) instances and to downsample majority instances simultaneously by a GMM-SMOTE sampler; a very powerful state-of-the-art classifier, the XGBoost, to classify instances into RI and non-RI and to evaluate variable importance based on the information gain; a hyperparameter tuning procedure tweaking hyperparameters appearing in all of the three above components, within pre-defined value ranges.

Based on the AI system shown in Figure 1.1, three models, the COR-SHIPS model, the LLE-SHIPS model, and the DL-SHIPS model, are developed. The COR-SHIPS model only employs SHIPS data and is the continued work of Y16. Comparing with Y16, the COR-SHIPS model adopts a different data filter, oversamples RI instances, employs a more powerful classifier, and tunes their hyper-parameters to improve the performance.

However, the COR-SHIPS model still only employs the SHIPS dataset, which is largely based on expert experiences. Since the mechanism of the tropical cyclone RI is unknown, the knowledge from the domain scientist may not be comprehensive, which indicates some important variables may not be included in the SHIPS dataset. Therefore, ECMWF ERA-Interim reanalysis data are used to improve the performance of the COR-SHIPS model. Two automatic feature extraction approaches, local linear embedding (LLE) and deep learning (DL), are used to extract features from near center data (small scale) and large-scale data respectively to create the LLE-SHIPS model and DL-SHIPS model.

The entire dataset is split into training/validation and test set, where the former is used to fit our model and to tune the hyperparameters, and the latter is used for the performance evaluation and comparison. The performance of our model on the test data shows improvement in the RI prediction with hyperparameter tuning.

It is found that our three model outperforms Y16 by 20.9%, 23.8%, and 48.5%, and KRD15 by 49.5%, 53.1%, and 83.6% on POD, while reducing the FAR by 12.7%, 20.8%, and 38.8% comparing with Y16, and 24.7%, 31.8%, and 47.3% with KRD15 respectively. Our model also improves the kappa score of 28.7%, 65.1%, and 84.0% vs. Y16 and the PSS 63.6%, 77.3%, and 114% against KRD15. With the difficulties in RI prediction and the slow improvement rates in previous studies (KD03, KDK10, KRD15, Y16), we believe the improvement by this work is substantial. The significant improvement made by the three models also challenges the mainstream point of selecting only a few variables fitting in the simple model for the prediction, i.e., involving more

variables in the complicated model with high penalty terms is better than a simple model with few variables.

The variable importance is also evaluated, and BD12, the past 12-hour intensity change, is found to have the largest importance score and contributes more than other variables in all three models, and the common 6 variables in top 10 are BD12, VMAX, SHRD, DTL, IRM1_5, and G150.

Previous important variables for RI prediction are determined by the significance test in KD03, KDK10, and KRD15, and most of them are consistent with top 10 variables in our three models with some exceptions. The variables in the top 10 list but not considered in other RI studies may be helpful for future RI studies, especially for DLT, jd, G150, PW08, TWXC, REFC, and TGRD. The additional significant variables identified by the ERA-Interim data filter are the wind speed, especially at 400 and 450 hPa, potential vorticity at 1000 hPa, vertical pressure in the near center and southeast humidity (q), north relative vorticity (vo), and north eastward wind (u) in the large-scale and those would help understand the mechanism of the TC intensification.

COR-SHIPS, LLE-SHIPS, and DL-SHIPS model designed in this study performed significantly better than most of the previous works such as KRD15, and Y16. Although we are able to evaluate the importance of the output of different data filters, and we can somehow even trace back the importance to the feature level, the feature level importance is not accurate. Accurately tracing back the feature level importance is related to interpretability or explaiability of the complicated machine learning model, which is still a challenging problem in the AI field because with so many non-linear transforms

happening in the machine learning model structure, no one could easily tell what's going on (Molnar 2019). There are some attempts for the model interpretability mentioned above, such as Lime (Local Interpretable Model-Agnostic Explanations) (Ribeiro et al. 2016), that a simple/explainable model is used to approximate the underline model, SHAP (Shaley Additive Explanations) (Lundberg and Lee 2017) that a game theory-based approach is used to evaluate the feature level importance score, but these are all the workaround solutions, and more researches need to be done for more accurately describing the importance of the original variables.

APPENDIX 1 PRINCIPAL COMPONENT ANALYSIS

1.1 Principal component analysis

Principal component analysis (PCA) is a dimension reduction approach to identify new features (principal component, i.e., PC) that contain as much statistical information of the original features as possible, and PCs that are not correlated to each other. Statistical information is represented by the variance of the original features. The correlation between different PCs equal to 0, and PCs are sorted by their variance.

PCA is elaborated in math format as below:

Assume there are *m* observations in the entire dataset w.r.t. each observation $x^{(i)}$ $i = 1, \dots, m$, where each observation $x^{(i)}$ $(i = 1, \dots, m)$ is a multi-dimension vector.

The data $X = [x^{(1)}, \dots, x^{(m)}]$ is centered through $\hat{X} = X - [\frac{1}{m}\sum_{i=1}^{m} x_i, \dots, \frac{1}{m}\sum_{i=1}^{m} x_i]$, where $\hat{X}$ represents that each column of X is subtracted by $\frac{1}{m}\sum_{i=1}^{m} x_i$. Assume $A = [A_1, \dots, A_m]$ is the projection matrix that

$$Y = A\hat{X}$$

Where ${A_i}^T A_i = 1$ for $i = 1, \dots, m$, $Y = [Y_1, \dots, Y_m]$, and $Y_i Y_j = 0$ for $i, j = 1, \dots, m; i \neq j$.

(the largest variance for each Y_i while $Y_iY_j = 0$ for $i, j = 1, \dots, m; i \neq j$). Therefore, since the variance of Y_i that $Y_iY_j = 0$ for $i, j = 1, \dots, m; i \neq j$ should be maximized, hence the trace of the covariance matrix of $A\hat{X}$ should be maximized, which results in (1).

$$A = \underset{A}{argmax}\, trace(A^T S A) \tag{1}$$

$$s.t. \qquad A^T A = I \; and \; S = \frac{1}{m}\hat{X}\hat{X}^T$$

By using Lagrangian multiplier and taking the derivative on (1), we get

$$SA = \lambda A$$

where λ is Lagrangian multiplier.

Based on eigen-decomposition described in Stoer and Bulirsch (2013), λ is the diagonal eigenvalue matrix of S, where eigenvalues of S are located in the diagonal of λ, and sorted decreasingly from left upper corner to the right lower corner. A is the

corresponding eigenvector matrix of S, where A_i for $i = 1, \dots, m$ is the eigenvector that corresponds to *i-th* largest eigenvalue of S. A_i for $i = 1, \dots, m$ represent the first m PCs.

In applications, the first few PCs are chosen with the largest contribution to the total variance (variation).

More details can be found in Friedman et al. (2001).

1.2 Kernel PCA

Kernel PCA maps the original data X to a kernel Hilbert space through a transformation ϕ to perform the PCA, and the kernel space is unknown. Similarly, assume there are *m* observation in the entire dataset w.r.t. each observation $x^{(i)} \quad i = 1, \dots, m$. Therefore, data $X = [x^{(1)}, \dots, x^{(m)}]$, and $A = [A_1, \dots, A_m]$ is the projection matrix that

$$Y = A\phi(X)$$

$Y = [Y_1, \dots, Y_m]$, and $Y_i Y_j = 0$ for $i, j = 1, \dots, m; i \neq j$

Similar to PCA, the problem can be rewritten as (2)

$$A = \underset{A}{argmax}\, trace(A^T C A) \tag{2}$$

$$s.t. \qquad A^T A = I \; and \; C = \frac{1}{m}\widehat{\phi(X)}\widehat{\phi(X)}^T$$

where $\widehat{\phi(X)} = \phi(X) - [\frac{1}{m}\sum_{i=1}^{m}\phi(x_i), \ldots, \frac{1}{m}\sum_{i=1}^{m}\phi(x_i)]$

Equation (2) could be solved as same as Equation (1), hence convert to

$$CA = \lambda A \tag{3}$$

(3) can be transformed to vector form

$$CA_k = \lambda_k A_k \tag{4}$$

since $C = \frac{1}{m}\widehat{\phi(X)}\widehat{\phi(X)}^T$, (4) becomes

$$\frac{1}{m}\sum_{i=1}^{m}\widehat{\phi(x_i)}[\widehat{\phi(x_i)}^T A_k] = \lambda_k A_k \tag{5}$$

2600

2601 If both sides of the (5) is divided by λ_k, A_k can be rewritten as

$$A_k = \sum_{i=1}^{m} t_{ki}\phi(x_i) \tag{6}$$

2602

$$\frac{1}{m}\sum_{i=1}^{m}\widehat{\phi(x_i)}\widehat{\phi(x_i)}^T\sum_{j=1}^{m} t_{kj}\phi(x_j) = \lambda_k\sum_{i=1}^{m} t_{ki}\phi(x_i) \tag{7}$$

2603 Define $k(x_i, x_j) = \widehat{\phi(x_i)}^T\widehat{\phi(x_i)}$ for $i, j = 1, \dots, m$. K is a $m * m$ dimensional matrix

2604 that $K(i, j) = k(x_i, x_j)$ for $i, j = 1, \dots, m$.

$$\frac{1}{m}\sum_{i=1}^{m} k(x_l, x_i)\sum_{j=1}^{m} t_{kj}k(x_i, x_j) = \lambda_k\sum_{i=1}^{m} t_{ki}k(x_l, x_i) \tag{8}$$

2605 Then (8) becomes

$$K^2 t_k = \lambda_k m K t_k \tag{9}$$

2606

2607 where $t_k = [t_{k1}, \ldots, t_{km}]^T$

2608 (9) is divided by K

$$K t_k = \lambda_k m t_k \tag{10}$$

2609 Y_k is then calculated as:

$$Y_k = \widehat{\phi(X)}^T A_k = \sum_{i=1}^{m} t_{ki} k(x, x_i) \tag{11}$$

2610 (Schölkopf et al. 1997)

2611 More details can be found in Schölkopf et al. (1998).

2612

2613

2614

APPENDIX 2 ADDITIONAL TABLES

Table A1: The highly correlated parameter group list.

[IRM1_16, IR00_10, IR00_13, IR00_15, IR00_16, IR00_2, IR00_4, IR00_6, IR00_7, IR00_8, IR00_9, IRM1_10, IRM1_12, IRM1_13, IRM1_15, IRM1_2, IRM1_4, IRM1_6, IRM1_7, IRM1_8, IRM1_9, IRM3_10, IRM3_12, IRM3_13, IRM3_15, IRM3_16, IRM3_2, IRM3_4, IRM3_6, IRM3_7, IRM3_8, IRM3_9]

[E000, CSST, DSST, DSTA, ENEG, ENSS, EPOS, MTPW_10, MTPW_12, MTPW_14, MTPW_16, MTPW_20, MTPW_2, MTPW_4, MTPW_6, MTPW_8, PW03, PW05, PW07, PW09, PW11, PW13, PW15, PW17, PW21, RSST, T000, XDST, XNST, XTMX]

[HIST_8, HIST_9, HIST_10, HIST_11, HIST_12, HIST_13, HIST_14, HIST_2, HIST_3, HIST_4, HIST_5, HIST_6, HIST_7]

[PW14, MTPW_11, MTPW_13, MTPW_15, MTPW_3, MTPW_5, PW04, PW06, PW12, PW16]

[CD26, CD20, COHC, ND26, XD20, XD22, XD24, XD26, XO20, XOHC]

[MTPW_19, MTPW_0, MTPW_18, PW01, PW19, PW20], [HIST_15, HIST_16, HIST_17, HIST_18, HIST_19, HIST_20]

[IRM3_19, IRM1_18, IRM1_19, IRM1_20, IRM3_18, IRM3_20]

[PW08, MTPW_9, MTPW_17, PW10, PW18]

[PSLV_4, PSLV_2, PSLV_6, U200, U20C]

[IRM1_5, IR00_5, IRM1_3, IRM3_3, IRM3_5]

[SHRD, SHDC, SHGC, SHRG]

[V850, TWAC, V000, V500]

[IRM3_11, IR00_11, IRM1_11]

[RHMD, RHHI, RHLO]

[IR00_20, IR00_18, IR00_19]

[DTL, LON, TLON]

[PSLV_3, PSLV_5, PSLV_7]

[PENV, PENC, Z000]

[VVAC, VMFX, VVAV]

[BD12, BD06, BD18]

[T250, T200]

[NSST, NTMX]
[HIST_1, HIST_0]
[NOHC, RHCN]
[RD26, RD20]
[XD18, XD16]
[D200, DIVC]
[PC00, IR00_1]
[OAGE, NAGE]
[NTFR, XTFR]
[PCM1, IRM1_1]
[HE07, HE05]
[PEFC, V20C]
[PCM3, IRM3_1]
[VMAX, MSLP]
[TLAT, LAT]
[MTPW_1, PW02]
[O500, O700]
[NDFR, XDFR]
[IR00_12]
[VMPI]
[IR00_3]
[ND20]
[EPSS]
[TWXC]
[G150]
[SHTD]
[NDTX]
[XDML]
[Z850]
[CFLX]
[XDTX]

[SHTS]

[SDDC]

[jd]

[SHRS]

[IR00_14]

[IR00_17]

[G250]

[G200]

[REFC]

[PSLV_1]

[V300]

[IRM1_17]

[T150]

[TGRD]

[TADV]

[IRM3_14]

[R000]

[IRM3_17]

[IRM1_14]

Table A2: Detail of the Group with the original score from LLE data filter.

Group1, 0.07131729492630356, [NT18_cc_l11, NT18_cc_l12, NT18_cc_l13, NT18_cc_l14, NT18_cc_l15, NT18_ciwc_l12, NT18_ciwc_l13, NT18_ciwc_l14, NT18_ciwc_l15, NT18_ciwc_l16, NT18_o3_l1, NT18_o3_l10, NT18_o3_l11, NT18_o3_l14, NT18_o3_l15, NT18_o3_l16, NT18_o3_l17, NT18_o3_l2, NT18_o3_l9, NT18_q_l1, NT18_q_l2, NT18_q_l3, NT18_q_l37, NT18_q_l4, NT18_q_l5, NT18_q_l6, NT18_r_l11, NT18_r_l12, NT18_r_l14, NT18_r_l15, NT18_r_l16, NT18_t_l10, NT18_t_l14, NT18_t_l15, NT18_t_l16, NT18_t_l17, NT18_t_l18, NT18_t_l19, NT18_t_l20, NT18_t_l21, NT18_t_l22, NT18_t_l5, NT18_t_l6, NT18_t_l7, NT18_t_l8, NT18_t_l9, NT18_u_l11, NT18_u_l12, NT18_u_l13, NT18_u_l14, NT18_u_l15, NT18_u_l16, NT18_u_l25, NT18_u_l26, NT18_u_l27, NT18_u_l28, NT18_u_l29, NT18_u_l30, NT18_u_l31, NT18_u_l32, NT18_u_l33, NT18_u_l34, NT18_u_l35, NT18_u_l36, NT18_u_l37, NT12_cc_l11, NT12_cc_l12, NT12_cc_l13, NT12_cc_l14, NT12_cc_l15, NT12_ciwc_l12, NT12_ciwc_l13, NT12_ciwc_l14, NT12_ciwc_l15, NT12_ciwc_l16, NT12_o3_l1, NT12_o3_l10, NT12_o3_l11,

NT12_o3_l14, NT12_o3_l15, NT12_o3_l16, NT12_o3_l17, NT12_o3_l2, NT12_o3_l5,
NT12_o3_l9, NT12_q_l1, NT12_q_l13, NT12_q_l14, NT12_q_l2, NT12_q_l3,
NT12_q_l37, NT12_q_l4, NT12_q_l5, NT12_q_l6, NT12_r_l11, NT12_r_l12,
NT12_r_l13, NT12_r_l14, NT12_r_l15, NT12_r_l16, NT12_t_l10, NT12_t_l11,
NT12_t_l14, NT12_t_l15, NT12_t_l16, NT12_t_l17, NT12_t_l18, NT12_t_l19,
NT12_t_l20, NT12_t_l21, NT12_t_l22, NT12_t_l5, NT12_t_l6, NT12_t_l7, NT12_t_l8,
NT12_t_l9, NT12_u_l11, NT12_u_l12, NT12_u_l13, NT12_u_l14, NT12_u_l15,
NT12_u_l16, NT12_u_l25, NT12_u_l26, NT12_u_l27, NT12_u_l28, NT12_u_l29,
NT12_u_l30, NT12_u_l31, NT12_u_l32, NT12_u_l33, NT12_u_l34, NT12_u_l35,
NT12_u_l36, NT12_u_l37, NT12_z_l15, NT12_z_l16, NT12_z_l17, NT12_z_l18,
NT06_cc_l11, NT06_cc_l12, NT06_cc_l13, NT06_cc_l14, NT06_cc_l15,
NT06_ciwc_l12, NT06_ciwc_l13, NT06_ciwc_l14, NT06_ciwc_l15, NT06_ciwc_l16,
NT06_o3_l1, NT06_o3_l10, NT06_o3_l11, NT06_o3_l12, NT06_o3_l13, NT06_o3_l14,
NT06_o3_l15, NT06_o3_l16, NT06_o3_l17, NT06_o3_l2, NT06_o3_l5, NT06_o3_l9,
NT06_q_l1, NT06_q_l13, NT06_q_l14, NT06_q_l15, NT06_q_l2, NT06_q_l3,
NT06_q_l4, NT06_q_l5, NT06_q_l6, NT06_r_l11, NT06_r_l12, NT06_r_l13,
NT06_r_l14, NT06_r_l15, NT06_r_l16, NT06_t_l10, NT06_t_l11, NT06_t_l14,
NT06_t_l15, NT06_t_l16, NT06_t_l17, NT06_t_l18, NT06_t_l19, NT06_t_l20,
NT06_t_l21, NT06_t_l22, NT06_t_l28, NT06_t_l29, NT06_t_l30, NT06_t_l31,
NT06_t_l32, NT06_t_l5, NT06_t_l6, NT06_t_l7, NT06_t_l8, NT06_t_l9, NT06_u_l11,
NT06_u_l12, NT06_u_l13, NT06_u_l14, NT06_u_l15, NT06_u_l16, NT06_u_l25,
NT06_u_l26, NT06_u_l27, NT06_u_l28, NT06_u_l29, NT06_u_l30, NT06_u_l31,
NT06_u_l32, NT06_u_l33, NT06_u_l34, NT06_u_l35, NT06_u_l36, NT06_u_l37,
NT06_z_l14, NT06_z_l15, NT06_z_l16, NT06_z_l17, NT06_z_l18, NT06_z_l19,
NT00_cc_l11, NT00_cc_l12, NT00_cc_l13, NT00_cc_l14, NT00_cc_l15,
NT00_ciwc_l11, NT00_ciwc_l12, NT00_ciwc_l13, NT00_ciwc_l14, NT00_ciwc_l15,
NT00_ciwc_l16, NT00_o3_l1, NT00_o3_l10, NT00_o3_l11, NT00_o3_l12,
NT00_o3_l13, NT00_o3_l14, NT00_o3_l15, NT00_o3_l16, NT00_o3_l17, NT00_o3_l2,
NT00_o3_l5, NT00_o3_l9, NT00_q_l1, NT00_q_l13, NT00_q_l14, NT00_q_l15,
NT00_q_l2, NT00_q_l3, NT00_q_l4, NT00_q_l5, NT00_q_l6, NT00_q_l7, NT00_r_l11,
NT00_r_l12, NT00_r_l13, NT00_r_l14, NT00_r_l15, NT00_r_l16, NT00_t_l11,
NT00_t_l14, NT00_t_l15, NT00_t_l16, NT00_t_l17, NT00_t_l18, NT00_t_l19,
NT00_t_l20, NT00_t_l21, NT00_t_l22, NT00_t_l27, NT00_t_l28, NT00_t_l29,
NT00_t_l30, NT00_t_l31, NT00_t_l32, NT00_t_l33, NT00_t_l34, NT00_t_l5,
NT00_t_l6, NT00_t_l7, NT00_t_l8, NT00_t_l9, NT00_u_l11, NT00_u_l12,
NT00_u_l13, NT00_u_l14, NT00_u_l15, NT00_u_l26, NT00_u_l27, NT00_u_l28,
NT00_u_l29, NT00_u_l30, NT00_u_l31, NT00_u_l32, NT00_u_l33, NT00_u_l34,
NT00_u_l35, NT00_u_l36, NT00_u_l37, NT00_z_l13, NT00_z_l14, NT00_z_l15,
NT00_z_l16, NT00_z_l17, NT00_z_l18, NT00_z_l19, NT18_r_l13]
Group2, -0.040873502972860964, [NT18_cc_l16, NT18_cc_l17, NT18_cc_l18,
NT18_cc_l19, NT18_cc_l20, NT18_cc_l21, NT18_cc_l22, NT18_cc_l23, NT18_cc_l27,
NT18_cc_l28, NT18_cc_l29, NT18_cc_l30, NT18_cc_l31, NT18_cc_l32, NT18_cc_l33,
NT18_cc_l34, NT18_ciwc_l18, NT18_ciwc_l19, NT18_ciwc_l20, NT18_ciwc_l21,
NT18_clwc_l21, NT18_clwc_l22, NT18_clwc_l24, NT18_clwc_l25, NT18_clwc_l26,

NT18_clwc_l27, NT18_clwc_l28, NT18_clwc_l29, NT18_clwc_l30, NT18_clwc_l31,
NT18_q_l15, NT18_q_l16, NT18_q_l17, NT18_q_l18, NT18_q_l19, NT18_q_l20,
NT18_q_l21, NT18_q_l22, NT18_q_l23, NT18_q_l28, NT18_q_l29, NT18_q_l30,
NT18_q_l31, NT18_q_l32, NT18_q_l33, NT18_q_l34, NT18_r_l17, NT18_r_l18,
NT18_r_l19, NT18_r_l20, NT18_r_l21, NT18_r_l22, NT18_r_l23, NT18_r_l24,
NT18_r_l25, NT18_r_l26, NT18_r_l27, NT18_r_l28, NT18_r_l29, NT18_r_l30,
NT18_r_l31, NT18_r_l32, NT18_r_l33, NT18_r_l34, NT18_r_l35, NT18_r_l36,
NT12_cc_l16, NT12_cc_l17, NT12_cc_l18, NT12_cc_l19, NT12_cc_l20, NT12_cc_l21,
NT12_cc_l22, NT12_cc_l23, NT12_cc_l27, NT12_cc_l28, NT12_cc_l29, NT12_cc_l30,
NT12_cc_l31, NT12_cc_l32, NT12_cc_l33, NT12_ciwc_l17, NT12_ciwc_l18,
NT12_ciwc_l19, NT12_ciwc_l20, NT12_ciwc_l21, NT12_ciwc_l22, NT12_clwc_l19,
NT12_clwc_l20, NT12_clwc_l21, NT12_clwc_l22, NT12_clwc_l23, NT12_clwc_l24,
NT12_clwc_l25, NT12_clwc_l26, NT12_clwc_l27, NT12_clwc_l28, NT12_clwc_l29,
NT12_clwc_l30, NT12_clwc_l31, NT12_q_l16, NT12_q_l17, NT12_q_l18,
NT12_q_l19, NT12_q_l20, NT12_q_l21, NT12_q_l22, NT12_q_l23, NT12_q_l27,
NT12_q_l28, NT12_q_l29, NT12_q_l30, NT12_q_l31, NT12_q_l32, NT12_q_l33,
NT12_r_l17, NT12_r_l18, NT12_r_l19, NT12_r_l20, NT12_r_l22, NT12_r_l23,
NT12_r_l24, NT12_r_l25, NT12_r_l26, NT12_r_l27, NT12_r_l28, NT12_r_l29,
NT12_r_l30, NT12_r_l31, NT12_r_l32, NT12_r_l33, NT12_r_l34, NT12_r_l35,
NT12_r_l36, NT12_r_l37, NT06_cc_l16, NT06_cc_l17, NT06_cc_l18, NT06_cc_l19,
NT06_cc_l20, NT06_cc_l21, NT06_cc_l22, NT06_cc_l23, NT06_cc_l28, NT06_cc_l29,
NT06_cc_l30, NT06_cc_l31, NT06_cc_l32, NT06_cc_l33, NT06_ciwc_l18,
NT06_ciwc_l19, NT06_ciwc_l20, NT06_ciwc_l21, NT06_ciwc_l22, NT06_clwc_l20,
NT06_clwc_l21, NT06_clwc_l22, NT06_clwc_l23, NT06_clwc_l24, NT06_clwc_l25,
NT06_clwc_l26, NT06_clwc_l27, NT06_clwc_l28, NT06_clwc_l29, NT06_clwc_l30,
NT06_clwc_l31, NT06_q_l16, NT06_q_l17, NT06_q_l18, NT06_q_l19, NT06_q_l20,
NT06_q_l21, NT06_q_l22, NT06_q_l23, NT06_q_l24, NT06_q_l25, NT06_q_l26,
NT06_q_l27, NT06_q_l28, NT06_q_l29, NT06_q_l30, NT06_q_l31, NT06_q_l32,
NT06_r_l17, NT06_r_l18, NT06_r_l19, NT06_r_l20, NT06_r_l21, NT06_r_l22,
NT06_r_l23, NT06_r_l24, NT06_r_l25, NT06_r_l26, NT06_r_l27, NT06_r_l28,
NT06_r_l29, NT06_r_l30, NT06_r_l31, NT06_r_l32, NT06_r_l33, NT06_r_l34,
NT06_r_l35, NT06_r_l36, NT00_cc_l16, NT00_cc_l17, NT00_cc_l18, NT00_cc_l19,
NT00_cc_l20, NT00_cc_l21, NT00_cc_l22, NT00_cc_l31, NT00_cc_l32,
NT00_ciwc_l18, NT00_ciwc_l19, NT00_ciwc_l20, NT00_ciwc_l21, NT00_clwc_l21,
NT00_clwc_l22, NT00_clwc_l23, NT00_clwc_l24, NT00_clwc_l25, NT00_clwc_l26,
NT00_clwc_l27, NT00_clwc_l28, NT00_q_l17, NT00_q_l18, NT00_q_l19,
NT00_q_l20, NT00_q_l21, NT00_q_l22, NT00_q_l23, NT00_q_l24, NT00_q_l25,
NT00_q_l26, NT00_q_l27, NT00_r_l17, NT00_r_l18, NT00_r_l19, NT00_r_l20,
NT00_r_l21, NT00_r_l22, NT00_r_l23, NT00_r_l24, NT00_r_l25, NT00_r_l26,
NT00_r_l27, NT00_r_l28, NT00_r_l29, NT00_r_l30, NT00_r_l31, NT00_r_l32,
NT00_r_l33, NT00_r_l34, NT00_r_l35, NT12_r_l21]
Group3, 0.06259891129094408, [NT18_w_l1, NT18_w_l10, NT18_w_l11,
NT18_w_l12, NT18_w_l13, NT18_w_l14, NT18_w_l15, NT18_w_l16, NT18_w_l17,
NT18_w_l18, NT18_w_l19, NT18_w_l2, NT18_w_l20, NT18_w_l21, NT18_w_l22,

NT18_w_l23, NT18_w_l24, NT18_w_l25, NT18_w_l26, NT18_w_l27, NT18_w_l28,
NT18_w_l29, NT18_w_l3, NT18_w_l30, NT18_w_l31, NT18_w_l32, NT18_w_l33,
NT18_w_l34, NT18_w_l35, NT18_w_l36, NT18_w_l37, NT18_w_l4, NT18_w_l5,
NT18_w_l6, NT18_w_l7, NT18_w_l8, NT18_w_l9, NT12_w_l1, NT12_w_l10,
NT12_w_l11, NT12_w_l12, NT12_w_l13, NT12_w_l14, NT12_w_l15, NT12_w_l16,
NT12_w_l17, NT12_w_l18, NT12_w_l19, NT12_w_l2, NT12_w_l20, NT12_w_l21,
NT12_w_l22, NT12_w_l23, NT12_w_l24, NT12_w_l25, NT12_w_l26, NT12_w_l27,
NT12_w_l28, NT12_w_l29, NT12_w_l3, NT12_w_l30, NT12_w_l31, NT12_w_l32,
NT12_w_l33, NT12_w_l34, NT12_w_l35, NT12_w_l36, NT12_w_l37, NT12_w_l4,
NT12_w_l5, NT12_w_l6, NT12_w_l7, NT12_w_l8, NT12_w_l9, NT06_w_l1,
NT06_w_l10, NT06_w_l11, NT06_w_l12, NT06_w_l13, NT06_w_l14, NT06_w_l15,
NT06_w_l16, NT06_w_l17, NT06_w_l18, NT06_w_l19, NT06_w_l2, NT06_w_l20,
NT06_w_l21, NT06_w_l22, NT06_w_l23, NT06_w_l24, NT06_w_l25, NT06_w_l26,
NT06_w_l27, NT06_w_l28, NT06_w_l29, NT06_w_l3, NT06_w_l30, NT06_w_l31,
NT06_w_l32, NT06_w_l33, NT06_w_l34, NT06_w_l35, NT06_w_l36, NT06_w_l37,
NT06_w_l4, NT06_w_l5, NT06_w_l6, NT06_w_l7, NT06_w_l8, NT06_w_l9,
NT00_w_l1, NT00_w_l10, NT00_w_l11, NT00_w_l12, NT00_w_l13, NT00_w_l14,
NT00_w_l15, NT00_w_l16, NT00_w_l17, NT00_w_l18, NT00_w_l19, NT00_w_l2,
NT00_w_l20, NT00_w_l21, NT00_w_l22, NT00_w_l23, NT00_w_l24, NT00_w_l25,
NT00_w_l26, NT00_w_l27, NT00_w_l28, NT00_w_l29, NT00_w_l3, NT00_w_l30,
NT00_w_l31, NT00_w_l32, NT00_w_l33, NT00_w_l34, NT00_w_l35, NT00_w_l36,
NT00_w_l37, NT00_w_l4, NT00_w_l5, NT00_w_l6, NT00_w_l8, NT00_w_l9,
NT00_w_l7]
Group4, 0.02269292645906973, [NT18_pv_l10, NT18_pv_l11, NT18_pv_l12,
NT18_pv_l13, NT18_pv_l14, NT18_pv_l15, NT18_pv_l16, NT18_pv_l17,
NT18_pv_l18, NT18_pv_l19, NT18_pv_l20, NT18_pv_l21, NT18_pv_l22,
NT18_pv_l23, NT18_pv_l24, NT18_pv_l25, NT18_pv_l26, NT18_pv_l27,
NT18_pv_l28, NT18_pv_l29, NT18_pv_l3, NT18_pv_l30, NT18_pv_l31, NT18_pv_l32,
NT18_pv_l33, NT18_pv_l34, NT18_pv_l35, NT18_pv_l36, NT18_pv_l37, NT18_pv_l4,
NT18_pv_l6, NT18_pv_l7, NT18_pv_l8, NT18_pv_l9, NT12_pv_l10, NT12_pv_l11,
NT12_pv_l12, NT12_pv_l13, NT12_pv_l14, NT12_pv_l15, NT12_pv_l16,
NT12_pv_l17, NT12_pv_l18, NT12_pv_l19, NT12_pv_l20, NT12_pv_l21,
NT12_pv_l22, NT12_pv_l23, NT12_pv_l24, NT12_pv_l25, NT12_pv_l26,
NT12_pv_l27, NT12_pv_l28, NT12_pv_l29, NT12_pv_l3, NT12_pv_l30, NT12_pv_l31,
NT12_pv_l32, NT12_pv_l33, NT12_pv_l34, NT12_pv_l35, NT12_pv_l36,
NT12_pv_l37, NT12_pv_l4, NT12_pv_l5, NT12_pv_l6, NT12_pv_l7, NT12_pv_l8,
NT12_pv_l9, NT06_pv_l10, NT06_pv_l11, NT06_pv_l12, NT06_pv_l13, NT06_pv_l14,
NT06_pv_l15, NT06_pv_l16, NT06_pv_l17, NT06_pv_l18, NT06_pv_l19,
NT06_pv_l20, NT06_pv_l21, NT06_pv_l22, NT06_pv_l23, NT06_pv_l24,
NT06_pv_l25, NT06_pv_l26, NT06_pv_l27, NT06_pv_l28, NT06_pv_l29, NT06_pv_l3,
NT06_pv_l30, NT06_pv_l31, NT06_pv_l32, NT06_pv_l33, NT06_pv_l34,
NT06_pv_l35, NT06_pv_l36, NT06_pv_l37, NT06_pv_l4, NT06_pv_l5, NT06_pv_l6,
NT06_pv_l7, NT06_pv_l8, NT06_pv_l9, NT00_pv_l10, NT00_pv_l11, NT00_pv_l12,
NT00_pv_l13, NT00_pv_l14, NT00_pv_l15, NT00_pv_l16, NT00_pv_l17,

NT00_pv_l18, NT00_pv_l19, NT00_pv_l20, NT00_pv_l21, NT00_pv_l22,
NT00_pv_l23, NT00_pv_l24, NT00_pv_l25, NT00_pv_l26, NT00_pv_l27,
NT00_pv_l28, NT00_pv_l29, NT00_pv_l3, NT00_pv_l30, NT00_pv_l31, NT00_pv_l32,
NT00_pv_l33, NT00_pv_l34, NT00_pv_l35, NT00_pv_l36, NT00_pv_l37, NT00_pv_l4,
NT00_pv_l5, NT00_pv_l6, NT00_pv_l7, NT00_pv_l8, NT00_pv_l9, NT18_pv_l5]
Group5, -0.042245901124683405, [NT18_d_l1, NT18_d_l10, NT18_d_l11,
NT18_d_l12, NT18_d_l13, NT18_d_l14, NT18_d_l15, NT18_d_l16, NT18_d_l17,
NT18_d_l18, NT18_d_l19, NT18_d_l2, NT18_d_l20, NT18_d_l21, NT18_d_l22,
NT18_d_l23, NT18_d_l24, NT18_d_l25, NT18_d_l26, NT18_d_l28, NT18_d_l29,
NT18_d_l3, NT18_d_l30, NT18_d_l31, NT18_d_l4, NT18_d_l5, NT18_d_l6,
NT18_d_l7, NT18_d_l8, NT18_d_l9, NT12_d_l1, NT12_d_l10, NT12_d_l11,
NT12_d_l12, NT12_d_l13, NT12_d_l14, NT12_d_l15, NT12_d_l16, NT12_d_l17,
NT12_d_l18, NT12_d_l19, NT12_d_l2, NT12_d_l20, NT12_d_l21, NT12_d_l22,
NT12_d_l23, NT12_d_l24, NT12_d_l25, NT12_d_l26, NT12_d_l27, NT12_d_l28,
NT12_d_l29, NT12_d_l3, NT12_d_l30, NT12_d_l31, NT12_d_l32, NT12_d_l4,
NT12_d_l5, NT12_d_l6, NT12_d_l7, NT12_d_l8, NT12_d_l9, NT06_d_l1,
NT06_d_l10, NT06_d_l11, NT06_d_l12, NT06_d_l13, NT06_d_l14, NT06_d_l15,
NT06_d_l16, NT06_d_l17, NT06_d_l18, NT06_d_l19, NT06_d_l2, NT06_d_l20,
NT06_d_l21, NT06_d_l22, NT06_d_l23, NT06_d_l24, NT06_d_l25, NT06_d_l26,
NT06_d_l27, NT06_d_l28, NT06_d_l29, NT06_d_l3, NT06_d_l30, NT06_d_l31,
NT06_d_l32, NT06_d_l4, NT06_d_l5, NT06_d_l6, NT06_d_l7, NT06_d_l8,
NT06_d_l9, NT00_d_l1, NT00_d_l10, NT00_d_l11, NT00_d_l12, NT00_d_l13,
NT00_d_l14, NT00_d_l15, NT00_d_l16, NT00_d_l17, NT00_d_l18, NT00_d_l19,
NT00_d_l2, NT00_d_l20, NT00_d_l21, NT00_d_l22, NT00_d_l23, NT00_d_l24,
NT00_d_l25, NT00_d_l26, NT00_d_l27, NT00_d_l28, NT00_d_l29, NT00_d_l3,
NT00_d_l30, NT00_d_l31, NT00_d_l32, NT00_d_l4, NT00_d_l5, NT00_d_l6,
NT00_d_l7, NT00_d_l8, NT00_d_l9, NT18_d_l27]
Group6, -0.02193022436520109, [NT18_vo_l1, NT18_vo_l10, NT18_vo_l2,
NT18_vo_l20, NT18_vo_l21, NT18_vo_l22, NT18_vo_l23, NT18_vo_l24,
NT18_vo_l25, NT18_vo_l26, NT18_vo_l27, NT18_vo_l28, NT18_vo_l29, NT18_vo_l3,
NT18_vo_l30, NT18_vo_l31, NT18_vo_l32, NT18_vo_l33, NT18_vo_l34,
NT18_vo_l35, NT18_vo_l36, NT18_vo_l37, NT18_vo_l4, NT18_vo_l5, NT18_vo_l6,
NT18_vo_l7, NT18_vo_l8, NT18_vo_l9, NT12_vo_l1, NT12_vo_l10, NT12_vo_l2,
NT12_vo_l21, NT12_vo_l22, NT12_vo_l23, NT12_vo_l24, NT12_vo_l25,
NT12_vo_l26, NT12_vo_l27, NT12_vo_l28, NT12_vo_l29, NT12_vo_l3, NT12_vo_l30,
NT12_vo_l31, NT12_vo_l32, NT12_vo_l33, NT12_vo_l34, NT12_vo_l35,
NT12_vo_l36, NT12_vo_l37, NT12_vo_l4, NT12_vo_l5, NT12_vo_l6, NT12_vo_l7,
NT12_vo_l8, NT12_vo_l9, NT06_vo_l1, NT06_vo_l10, NT06_vo_l2, NT06_vo_l21,
NT06_vo_l22, NT06_vo_l23, NT06_vo_l24, NT06_vo_l25, NT06_vo_l26,
NT06_vo_l27, NT06_vo_l28, NT06_vo_l29, NT06_vo_l3, NT06_vo_l30, NT06_vo_l31,
NT06_vo_l32, NT06_vo_l33, NT06_vo_l34, NT06_vo_l35, NT06_vo_l36,
NT06_vo_l37, NT06_vo_l4, NT06_vo_l5, NT06_vo_l6, NT06_vo_l7, NT06_vo_l9,
NT00_vo_l1, NT00_vo_l10, NT00_vo_l2, NT00_vo_l21, NT00_vo_l22, NT00_vo_l23,
NT00_vo_l24, NT00_vo_l25, NT00_vo_l26, NT00_vo_l27, NT00_vo_l28,

NT00_vo_l29, NT00_vo_l3, NT00_vo_l30, NT00_vo_l31, NT00_vo_l32, NT00_vo_l33,
NT00_vo_l34, NT00_vo_l35, NT00_vo_l36, NT00_vo_l37, NT00_vo_l4, NT00_vo_l5,
NT00_vo_l6, NT00_vo_l7, NT00_vo_l8, NT00_vo_l9, NT06_vo_l8]
Group7, -0.053063711383979584, [NT18_clwc_l32, NT18_clwc_l33, NT18_q_l35,
NT18_q_l36, NT18_r_l37, NT18_t_l1, NT18_t_l12, NT18_t_l13, NT18_t_l2,
NT18_t_l23, NT18_t_l24, NT18_t_l25, NT18_t_l26, NT18_t_l27, NT18_t_l28,
NT18_t_l29, NT18_t_l3, NT18_t_l30, NT18_t_l31, NT18_t_l32, NT18_t_l33,
NT18_t_l34, NT18_t_l35, NT18_t_l36, NT18_t_l37, NT18_t_l4, NT12_clwc_l32,
NT12_clwc_l33, NT12_q_l15, NT12_q_l35, NT12_q_l36, NT12_t_l1, NT12_t_l13,
NT12_t_l2, NT12_t_l23, NT12_t_l24, NT12_t_l25, NT12_t_l26, NT12_t_l27,
NT12_t_l28, NT12_t_l29, NT12_t_l3, NT12_t_l30, NT12_t_l31, NT12_t_l32,
NT12_t_l33, NT12_t_l34, NT12_t_l35, NT12_t_l36, NT12_t_l37, NT12_t_l4,
NT06_clwc_l32, NT06_clwc_l33, NT06_r_l37, NT06_t_l1, NT06_t_l13, NT06_t_l2,
NT06_t_l23, NT06_t_l24, NT06_t_l25, NT06_t_l26, NT06_t_l27, NT06_t_l3,
NT06_t_l33, NT06_t_l34, NT06_t_l35, NT06_t_l36, NT06_t_l37, NT06_t_l4,
NT00_clwc_l32, NT00_clwc_l33, NT00_q_l16, NT00_r_l36, NT00_r_l37, NT00_t_l1,
NT00_t_l13, NT00_t_l2, NT00_t_l23, NT00_t_l24, NT00_t_l26, NT00_t_l3,
NT00_t_l35, NT00_t_l36, NT00_t_l37, NT00_t_l4, NT00_t_l25]
Group8, 0.0, [NT18_clwc_l1, NT18_clwc_l10, NT18_clwc_l11, NT18_clwc_l12,
NT18_clwc_l13, NT18_clwc_l14, NT18_clwc_l15, NT18_clwc_l16, NT18_clwc_l17,
NT18_clwc_l2, NT18_clwc_l3, NT18_clwc_l4, NT18_clwc_l5, NT18_clwc_l6,
NT18_clwc_l7, NT18_clwc_l8, NT18_clwc_l9, NT12_clwc_l1, NT12_clwc_l10,
NT12_clwc_l11, NT12_clwc_l12, NT12_clwc_l13, NT12_clwc_l14, NT12_clwc_l15,
NT12_clwc_l16, NT12_clwc_l17, NT12_clwc_l2, NT12_clwc_l3, NT12_clwc_l4,
NT12_clwc_l5, NT12_clwc_l6, NT12_clwc_l7, NT12_clwc_l8, NT12_clwc_l9,
NT06_clwc_l1, NT06_clwc_l10, NT06_clwc_l11, NT06_clwc_l12, NT06_clwc_l13,
NT06_clwc_l14, NT06_clwc_l15, NT06_clwc_l16, NT06_clwc_l17, NT06_clwc_l2,
NT06_clwc_l3, NT06_clwc_l4, NT06_clwc_l5, NT06_clwc_l6, NT06_clwc_l7,
NT06_clwc_l8, NT06_clwc_l9, NT00_clwc_l1, NT00_clwc_l10, NT00_clwc_l11,
NT00_clwc_l12, NT00_clwc_l13, NT00_clwc_l14, NT00_clwc_l15, NT00_clwc_l16,
NT00_clwc_l17, NT00_clwc_l3, NT00_clwc_l4, NT00_clwc_l5, NT00_clwc_l6,
NT00_clwc_l7, NT00_clwc_l8, NT00_clwc_l9, NT00_clwc_l2]
Group9, -0.026883331336958416, [NT18_ciwc_l24, NT18_o3_l12, NT18_o3_l13,
NT18_o3_l5, NT18_r_l10, NT18_t_l11, NT18_z_l11, NT18_z_l12, NT18_z_l13,
NT18_z_l14, NT18_z_l15, NT18_z_l16, NT18_z_l18, NT18_z_l19, NT18_z_l20,
NT18_z_l21, NT18_z_l22, NT18_z_l23, NT18_z_l24, NT18_z_l25, NT12_o3_l12,
NT12_o3_l13, NT12_r_l10, NT12_z_l11, NT12_z_l12, NT12_z_l13, NT12_z_l14,
NT12_z_l19, NT12_z_l20, NT12_z_l21, NT12_z_l22, NT12_z_l23, NT12_z_l24,
NT12_z_l25, NT06_r_l10, NT06_u_l24, NT06_z_l10, NT06_z_l11, NT06_z_l12,
NT06_z_l13, NT06_z_l20, NT06_z_l21, NT06_z_l22, NT06_z_l23, NT06_z_l24,
NT06_z_l25, NT00_r_l10, NT00_u_l24, NT00_u_l25, NT00_z_l10, NT00_z_l11,
NT00_z_l12, NT00_z_l20, NT00_z_l21, NT00_z_l22, NT00_z_l23, NT00_z_l24,
NT00_z_l25, NT18_z_l17]

Group10, 0.006458219563131418, [NT18_o3_l25, NT18_o3_l26, NT18_o3_l27,
NT18_o3_l28, NT18_o3_l29, NT18_o3_l30, NT18_o3_l31, NT18_o3_l32,
NT18_o3_l33, NT18_o3_l34, NT18_o3_l35, NT12_o3_l24, NT12_o3_l25,
NT12_o3_l26, NT12_o3_l27, NT12_o3_l28, NT12_o3_l29, NT12_o3_l30,
NT12_o3_l31, NT12_o3_l32, NT12_o3_l33, NT12_o3_l34, NT12_o3_l35, NT12_z_l37,
NT06_o3_l24, NT06_o3_l25, NT06_o3_l26, NT06_o3_l27, NT06_o3_l28,
NT06_o3_l29, NT06_o3_l30, NT06_o3_l31, NT06_o3_l33, NT06_o3_l34,
NT06_o3_l35, NT06_z_l37, NT00_o3_l24, NT00_o3_l25, NT00_o3_l26, NT00_o3_l27,
NT00_o3_l28, NT00_o3_l29, NT00_o3_l30, NT00_o3_l31, NT00_o3_l32,
NT00_o3_l33, NT00_o3_l34, NT00_o3_l35, NT06_o3_l32]
Group11, -0.04429699024103351, [NT18_v_l27, NT18_v_l28, NT18_v_l29,
NT18_v_l30, NT18_v_l31, NT18_v_l32, NT18_v_l33, NT18_v_l34, NT18_v_l35,
NT18_v_l36, NT18_v_l37, NT12_v_l27, NT12_v_l28, NT12_v_l29, NT12_v_l30,
NT12_v_l31, NT12_v_l32, NT12_v_l33, NT12_v_l34, NT12_v_l35, NT12_v_l36,
NT12_v_l37, NT06_v_l27, NT06_v_l28, NT06_v_l29, NT06_v_l30, NT06_v_l32,
NT06_v_l33, NT06_v_l34, NT06_v_l35, NT06_v_l36, NT06_v_l37, NT00_v_l27,
NT00_v_l28, NT00_v_l29, NT00_v_l30, NT00_v_l31, NT00_v_l32, NT00_v_l33,
NT00_v_l34, NT00_v_l35, NT00_v_l36, NT00_v_l37, NT06_v_l31]
Group12, 0.016470284903957744, [NT18_vo_l11, NT18_vo_l12, NT18_vo_l13,
NT18_vo_l14, NT18_vo_l15, NT18_vo_l16, NT18_vo_l17, NT18_vo_l18,
NT18_vo_l19, NT12_vo_l11, NT12_vo_l12, NT12_vo_l13, NT12_vo_l14,
NT12_vo_l15, NT12_vo_l16, NT12_vo_l17, NT12_vo_l18, NT12_vo_l19,
NT12_vo_l20, NT06_vo_l12, NT06_vo_l13, NT06_vo_l14, NT06_vo_l15,
NT06_vo_l16, NT06_vo_l17, NT06_vo_l18, NT06_vo_l19, NT06_vo_l20,
NT00_vo_l11, NT00_vo_l12, NT00_vo_l13, NT00_vo_l14, NT00_vo_l15,
NT00_vo_l16, NT00_vo_l17, NT00_vo_l18, NT00_vo_l19, NT00_vo_l20,
NT06_vo_l11]
Group13, 0.023626937292063666, [NT18_cc_l25, NT18_cc_l26, NT12_cc_l24,
NT12_cc_l25, NT12_cc_l26, NT12_cc_l34, NT12_cc_l35, NT12_cc_l36, NT12_cc_l37,
NT06_cc_l24, NT06_cc_l25, NT06_cc_l26, NT06_cc_l34, NT06_cc_l35, NT06_cc_l36,
NT06_cc_l37, NT06_clwc_l34, NT06_clwc_l35, NT06_clwc_l36, NT06_clwc_l37,
NT00_cc_l23, NT00_cc_l24, NT00_cc_l25, NT00_cc_l26, NT00_cc_l27, NT00_cc_l28,
NT00_cc_l29, NT00_cc_l30, NT00_cc_l33, NT00_cc_l34, NT00_cc_l35, NT00_cc_l36,
NT00_cc_l37, NT06_cc_l27]
Group14, -0.07471531491655992, [NT18_v_l10, NT18_v_l2, NT18_v_l3, NT18_v_l4,
NT18_v_l5, NT18_v_l6, NT18_v_l7, NT18_v_l8, NT18_v_l9, NT12_v_l10,
NT12_v_l3, NT12_v_l4, NT12_v_l5, NT12_v_l6, NT12_v_l7, NT12_v_l8, NT12_v_l9,
NT06_v_l10, NT06_v_l3, NT06_v_l4, NT06_v_l5, NT06_v_l6, NT06_v_l7,
NT06_v_l8, NT06_v_l9, NT00_v_l10, NT00_v_l2, NT00_v_l3, NT00_v_l4,
NT00_v_l5, NT00_v_l7, NT00_v_l8, NT00_v_l9, NT00_v_l6]
Group15, 0.0, [NT18_ciwc_l1, NT18_ciwc_l2, NT18_ciwc_l3, NT18_ciwc_l4,
NT18_ciwc_l5, NT18_ciwc_l6, NT18_ciwc_l7, NT18_ciwc_l8, NT18_ciwc_l9,
NT12_ciwc_l1, NT12_ciwc_l2, NT12_ciwc_l3, NT12_ciwc_l4, NT12_ciwc_l5,
NT12_ciwc_l7, NT12_ciwc_l8, NT06_ciwc_l1, NT06_ciwc_l2, NT06_ciwc_l3,

NT06_ciwc_l4, NT06_ciwc_l5, NT06_ciwc_l6, NT06_ciwc_l7, NT06_ciwc_l8,
NT00_ciwc_l1, NT00_ciwc_l2, NT00_ciwc_l3, NT00_ciwc_l4, NT00_ciwc_l5,
NT00_ciwc_l6, NT00_ciwc_l7, NT00_ciwc_l8, NT12_ciwc_l6]
Group16, 0.025737968406863487, [NT18_r_l1, NT18_r_l2, NT18_r_l3, NT18_r_l4,
NT18_r_l5, NT18_r_l6, NT18_r_l7, NT18_r_l8, NT12_r_l1, NT12_r_l2, NT12_r_l3,
NT12_r_l4, NT12_r_l5, NT12_r_l6, NT12_r_l7, NT12_r_l8, NT06_r_l1, NT06_r_l2,
NT06_r_l3, NT06_r_l4, NT06_r_l5, NT06_r_l6, NT06_r_l7, NT06_r_l8, NT00_r_l1,
NT00_r_l2, NT00_r_l3, NT00_r_l4, NT00_r_l6, NT00_r_l7, NT00_r_l8, NT00_r_l5]
Group17, 0.0032523694495293354, [NT18_z_l26, NT18_z_l27, NT18_z_l28,
NT18_z_l29, NT18_z_l30, NT18_z_l31, NT18_z_l32, NT18_z_l33, NT12_z_l26,
NT12_z_l27, NT12_z_l28, NT12_z_l29, NT12_z_l30, NT12_z_l31, NT12_z_l32,
NT12_z_l33, NT06_z_l26, NT06_z_l27, NT06_z_l28, NT06_z_l29, NT06_z_l30,
NT06_z_l31, NT06_z_l32, NT00_z_l27, NT00_z_l28, NT00_z_l29, NT00_z_l30,
NT00_z_l31, NT00_z_l32, NT00_z_l26]
Group18, -0.012730578721368735, [NT18_v_l11, NT18_v_l12, NT18_v_l13,
NT18_v_l14, NT18_v_l15, NT18_v_l16, NT18_v_l17, NT12_v_l11, NT12_v_l12,
NT12_v_l13, NT12_v_l15, NT12_v_l16, NT12_v_l17, NT06_v_l11, NT06_v_l12,
NT06_v_l13, NT06_v_l14, NT06_v_l15, NT06_v_l16, NT00_v_l12, NT00_v_l13,
NT00_v_l14, NT00_v_l15, NT00_v_l16, NT12_v_l14]
Group19, -0.0462370812127475, [NT18_v_l21, NT18_v_l22, NT18_v_l23,
NT18_v_l24, NT18_v_l25, NT18_v_l26, NT12_v_l21, NT12_v_l22, NT12_v_l23,
NT12_v_l24, NT12_v_l26, NT06_v_l21, NT06_v_l22, NT06_v_l23, NT06_v_l24,
NT06_v_l25, NT06_v_l26, NT00_v_l21, NT00_v_l22, NT00_v_l23, NT00_v_l24,
NT00_v_l25, NT00_v_l26, NT12_v_l25]
Group20, -0.03416642562085348, [NT18_u_l20, NT18_u_l21, NT18_u_l22,
NT18_u_l23, NT18_u_l24, NT18_u_l9, NT12_u_l20, NT12_u_l21, NT12_u_l22,
NT12_u_l23, NT12_u_l24, NT12_u_l9, NT06_u_l20, NT06_u_l21, NT06_u_l22,
NT06_u_l23, NT06_u_l9, NT00_u_l20, NT00_u_l21, NT00_u_l22, NT00_u_l9,
NT00_u_l23]
Group21, -0.015980995488767524, [NT18_o3_l4, NT18_z_l10, NT18_z_l8, NT18_z_l9,
NT12_o3_l4, NT12_z_l8, NT12_z_l9, NT06_o3_l4, NT06_z_l7, NT06_z_l8,
NT06_z_l9, NT00_o3_l4, NT00_z_l5, NT00_z_l6, NT00_z_l7, NT00_z_l8, NT00_z_l9,
NT12_z_l10]
Group22, -0.022620312860484826, [NT18_z_l3, NT18_z_l4, NT18_z_l5, NT18_z_l6,
NT18_z_l7, NT12_z_l3, NT12_z_l4, NT12_z_l5, NT12_z_l6, NT12_z_l7, NT06_z_l3,
NT06_z_l4, NT06_z_l5, NT06_z_l6, NT00_z_l2, NT00_z_l3, NT00_z_l4]
Group23, -0.009525048312305717, [NT18_ciwc_l11, NT18_ciwc_l17, NT12_ciwc_l11,
NT12_q_l34, NT06_ciwc_l11, NT06_q_l34, NT06_q_l35, NT00_ciwc_l17,
NT00_clwc_l29, NT00_clwc_l30, NT00_clwc_l31, NT00_q_l34, NT00_q_l35,
NT06_ciwc_l17]
Group24, 0.016470284903957744, [NT18_o3_l37, NT12_o3_l37, NT06_o3_l37,
NT06_q_l36, NT00_o3_l37, NT00_q_l28, NT00_q_l29, NT00_q_l30, NT00_q_l31,
NT00_q_l32, NT00_q_l33, NT00_q_l36, NT06_q_l33]

Group25, 0.003052627038463651, [NT06_ciwc_l25, NT06_ciwc_l26, NT06_ciwc_l27, NT06_ciwc_l29, NT06_ciwc_l30, NT06_ciwc_l31, NT06_ciwc_l32, NT06_ciwc_l33, NT06_ciwc_l34, NT06_ciwc_l35, NT06_ciwc_l36, NT06_ciwc_l37, NT06_ciwc_l28]
Group26, 0.025737968406863487, [NT18_ciwc_l25, NT18_ciwc_l26, NT18_ciwc_l27, NT18_ciwc_l29, NT18_ciwc_l30, NT18_ciwc_l31, NT18_ciwc_l32, NT18_ciwc_l33, NT18_ciwc_l34, NT18_ciwc_l35, NT18_ciwc_l36, NT18_ciwc_l37, NT18_ciwc_l28]
Group27, 0.031109314730230153, [NT18_o3_l18, NT18_o3_l19, NT18_o3_l20, NT12_o3_l18, NT12_o3_l19, NT12_o3_l20, NT06_o3_l18, NT06_o3_l19, NT06_o3_l20, NT00_o3_l19, NT00_o3_l20, NT00_o3_l18]
Group28, 0.0032523694495293354, [NT18_u_l4, NT18_u_l5, NT18_u_l6, NT12_u_l4, NT12_u_l5, NT12_u_l6, NT06_u_l4, NT06_u_l6, NT00_u_l4, NT00_u_l5, NT00_u_l6, NT06_u_l5]
Group29, 0.0712300559164073, [NT18_u_l1, NT18_u_l18, NT12_u_l1, NT12_u_l17, NT12_u_l18, NT06_u_l1, NT06_u_l17, NT00_t_l10, NT00_u_l16, NT00_u_l17, NT18_u_l17]
Group30, 0.04046369350173762, [NT18_u_l10, NT18_u_l7, NT12_u_l10, NT12_u_l7, NT12_u_l8, NT06_u_l10, NT06_u_l7, NT06_u_l8, NT00_u_l7, NT00_u_l8, NT18_u_l8]
Group31, 0.026777194098555834, [NT12_ciwc_l25, NT12_ciwc_l26, NT12_ciwc_l27, NT12_ciwc_l28, NT12_ciwc_l29, NT12_ciwc_l30, NT12_ciwc_l31, NT06_ciwc_l24, NT12_ciwc_l24]
Group32, -0.0032999087322507226, [NT18_d_l33, NT18_d_l34, NT18_d_l35, NT18_d_l36, NT18_d_l37, NT12_d_l33, NT12_d_l34, NT00_d_l33, NT18_d_l32]
Group33, -0.03088094284403675, [NT18_u_l3, NT12_u_l2, NT12_u_l3, NT06_u_l2, NT06_u_l3, NT00_u_l1, NT00_u_l2, NT00_u_l3, NT18_u_l2]
Group34, -0.015980995488767524, [NT12_d_l35, NT06_d_l34, NT06_d_l35, NT06_d_l36, NT06_d_l37, NT00_d_l34, NT00_d_l35, NT06_d_l33]
Group35, 0.016470284903957744, [NT18_v_l19, NT18_v_l20, NT12_v_l19, NT06_v_l19, NT06_v_l20, NT00_v_l19, NT00_v_l20, NT12_v_l20]
Group36, -0.015980995488767524, [NT18_o3_l7, NT12_o3_l7, NT12_o3_l8, NT06_o3_l7, NT06_o3_l8, NT00_o3_l7, NT00_o3_l8, NT18_o3_l8]
Group37, -0.009776487332806783, [NT18_o3_l22, NT12_o3_l21, NT12_o3_l22, NT06_o3_l21, NT06_o3_l22, NT00_o3_l21, NT00_o3_l22, NT18_o3_l21]
Group38, 0.012734297741761935, [NT00_ciwc_l30, NT00_ciwc_l31, NT00_ciwc_l33, NT00_ciwc_l34, NT00_ciwc_l35, NT00_ciwc_l36, NT00_ciwc_l37, NT00_ciwc_l32]
Group39, -0.01927724806311193, [NT18_cc_l35, NT18_cc_l36, NT18_cc_l37, NT18_clwc_l23, NT18_clwc_l36, NT18_clwc_l37, NT18_cc_l24]
Group40, 0.029881404183944138, [NT18_q_l24, NT18_q_l25, NT18_q_l26, NT18_q_l27, NT12_q_l24, NT12_q_l25, NT12_q_l26]
Group41, 0.02269292645906973, [NT18_q_l7, NT18_q_l8, NT12_q_l7, NT12_q_l8, NT06_q_l8, NT00_q_l8, NT06_q_l7]
Group42, -0.0032999087322507226, [NT00_ciwc_l23, NT00_ciwc_l25, NT00_ciwc_l26, NT00_ciwc_l27, NT00_ciwc_l28, NT00_ciwc_l29, NT00_ciwc_l24]

Group43, -0.022620312860484826, [NT18_q_l12, NT18_q_l13, NT12_q_l12, NT06_q_l12, NT00_q_l12, NT18_q_l14]
Group44, 0.010064393684136919, [NT12_u_l19, NT06_u_l18, NT06_u_l19, NT00_u_l10, NT00_u_l19, NT18_u_l19]
Group45, -0.025199275642583396, [NT12_ciwc_l32, NT12_ciwc_l33, NT12_ciwc_l34, NT12_ciwc_l36, NT12_ciwc_l37, NT12_ciwc_l35]
Group46, 0.04046369350173762, [NT12_z_l36, NT06_z_l35, NT00_z_l35, NT00_z_l36, NT00_z_l37, NT06_z_l36]
Group47, 0.03294056980791549, [NT18_clwc_l19, NT18_clwc_l20, NT00_clwc_l19, NT00_clwc_l20, NT06_clwc_l19]
Group48, 0.04348999451297464, [NT18_o3_l23, NT12_o3_l23, NT06_o3_l23, NT00_o3_l23, NT18_o3_l24]
Group49, 0.08554519157484475, [NT12_v_l18, NT06_v_l18, NT00_v_l17, NT00_v_l18, NT06_v_l17]
Group50, 0.026777194098555834, [NT12_cc_l10, NT12_ciwc_l10, NT06_ciwc_l10, NT00_cc_l10, NT06_cc_l10]
Group51, 0.013290747725846708, [NT18_clwc_l34, NT12_clwc_l35, NT12_clwc_l36, NT12_clwc_l34]
Group52, -0.019890565190348153, [NT12_z_l34, NT06_z_l34, NT00_z_l33, NT06_z_l33]
Group53, 0.02269292645906973, [NT18_o3_l3, NT06_o3_l3, NT00_o3_l3, NT12_o3_l3]
Group54, 0.010064393684136919, [NT18_o3_l36, NT12_o3_l36, NT00_o3_l36, NT06_o3_l36]
Group55, 0.01718412356692567, [NT12_pv_l2, NT06_pv_l2, NT00_pv_l2, NT18_pv_l2]
Group56, 0.026777194098555834, [NT18_q_l11, NT12_q_l11, NT00_q_l11, NT06_q_l11]
Group57, 0.0032523694495293354, [NT18_r_l9, NT12_r_l9, NT00_r_l9, NT06_r_l9]
Group58, -0.02851378448120112, [NT12_q_l9, NT06_q_l9, NT00_q_l9, NT18_q_l9]
Group59, -0.03528302096554892, [NT18_q_l10, NT06_q_l10, NT00_q_l10, NT12_q_l10]
Group60, -0.051288793867985794, [NT00_clwc_l35, NT00_clwc_l36, NT00_clwc_l34]
Group61, -0.01927724806311193, [NT18_z_l2, NT06_z_l2, NT12_z_l2]
Group62, 0.025737968406863487, [NT12_o3_l6, NT00_o3_l6, NT06_o3_l6]
Group63, 0.03892763335500793, [NT12_pv_l1, NT00_pv_l1, NT06_pv_l1]
Group64, 0.006458219563131418, [NT18_clwc_l18, NT06_clwc_l18, NT12_clwc_l18]
Group65, 0.05435473046696537, [NT00_q_l37, NT06_q_l37]
Group66, 0.026777194098555834, [NT12_d_l37, NT12_d_l36]
Group67, -0.013099278001989068, [NT00_d_l37, NT00_d_l36]
Group68, 0.00961854131774742, [NT18_z_l35, NT18_z_l34]
Group69, 0.02269292645906973, [NT18_z_l36, NT18_z_l37]
Group70, 0.006149179017971962, [NT06_t_l12, NT12_t_l12]
Group71, -0.009776487332806783, [NT18_ciwc_l10, NT18_cc_l10]

Group72, -0.03187470567985029, [NT00_cc_l9, NT00_ciwc_l9]
Group73, 0.03739216600762729, [NT06_cc_l9, NT06_ciwc_l9]
Group74, -0.013099278001989068, [NT12_cc_l9, NT12_ciwc_l9]
Group75, -0.022620312860484826, [NT00_ciwc_l22]
Group76, 0.01960401690104774, [NT18_clwc_l35]
Group77, 0.006458219563131418, [NT18_ciwc_l22]
Group78, 0.00961854131774742, [NT12_clwc_l37]
Group79, -0.06863045489336894, [NT00_clwc_l37]
Group80, 0.05516234025094624, [NT00_v_l11]
Group81, 0.04046369350173762, [NT00_u_l18]
Group82, 0.031109314730230153, [NT18_v_l18]
Group83, -0.016470284903957633, [NT00_z_l34]
Group84, 0.03294056980791549, [NT12_z_l35]
Group85, 0.034274390568905466, [NT12_v_l2]
Group86, -0.013099278001989068, [NT06_v_l2]
Group87, -0.03528302096554892, [NT00_clwc_l18]
Group88, 0.07965337355823643, [NT18_pv_l1]
Group89, 0.03294056980791549, [NT00_t_l12]
Group90, 0.026777194098555834, [NT00_ciwc_l10]
Group91, 0.02269292645906973, [NT06_ciwc_l23]
Group92, -0.0065008857713687584, [NT06_v_l1]
Group93, -0.05485916703871707, [NT18_z_l1]
Group94, -0.047768415870933056, [NT18_v_l1]
Group95, -0.047768415870933056, [NT18_ciwc_l23]
Group96, 0.016470284903957744, [NT12_ciwc_l23]
Group97, 0.0, [NT18_o3_l6]
Group98, -0.042245901124683405, [NT00_v_l1]
Group99, -0.053063711383979584, [NT12_v_l1]
Group100, 0.01960401690104774, [NT12_z_l1]
Group101, -0.03187470567985029, [NT00_z_l1]
Group102, 0.03294056980791549, [NT06_z_l1]
Group103, 0.0, [NT18_cc_l2]
Group104, 0.0, [NT06_cc_l1]
Group105, 0.0, [NT12_cc_l7]
Group106, 0.0, [NT06_cc_l8]
Group107, 0.0, [NT00_cc_l6]
Group108, 0.0, [NT18_cc_l8]
Group109, 0.0, [NT00_cc_l3]
Group110, 0.0, [NT00_cc_l4]
Group111, 0.0, [NT12_cc_l1]
Group112, 0.0, [NT12_cc_l5]
Group113, 0.0, [NT12_cc_l8]
Group114, 0.0, [NT12_cc_l2]
Group115, 0.0, [NT18_cc_l4]

Group116, 0.0, [NT00_cc_l1]
Group117, 0.0, [NT00_cc_l7]
Group118, 0.0, [NT18_cc_l5]
Group119, 0.0, [NT00_cc_l8]
Group120, 0.0, [NT06_cc_l4]
Group121, 0.0, [NT06_cc_l3]
Group122, 0.0, [NT00_cc_l5]
Group123, 0.0, [NT06_cc_l7]
Group124, 0.0, [NT06_cc_l5]
Group125, 0.0, [NT18_cc_l6]
Group126, 0.0, [NT18_cc_l9]
Group127, 0.0, [NT12_cc_l3]
Group128, 0.0, [NT18_cc_l3]
Group129, 0.0, [NT06_cc_l6]
Group130, 0.0, [NT06_cc_l2]
Group131, 0.0, [NT18_cc_l1]
Group132, 0.0, [NT18_cc_l7]
Group133, 0.0, [NT00_cc_l2]
Group134, 0.0, [NT12_cc_l4]
Group135, 0.0, [NT12_cc_l6]

Table A3: Group level importance from LLE data filter with the group number, the number of variables in each group, the importance score, and the rank.

Group	Num	score	Rank
49	5	0.023614	1
88	1	0.021988	2
1	309	0.019687	3
29	11	0.019662	4
3	148	0.01728	5
80	1	0.015227	6
65	2	0.015004	7
48	5	0.012005	8
81	1	0.01117	9
46	6	0.01117	10
30	11	0.01117	11
63	3	0.010746	12
73	2	0.010322	13
85	1	0.009461	14
47	5	0.009093	15

84	1	0.009093	16
102	1	0.009093	17
89	1	0.009093	18
82	1	0.008587	19
27	12	0.008587	20
40	7	0.008249	21
90	1	0.007392	22
56	4	0.007392	23
50	5	0.007392	24
66	2	0.007392	25
31	9	0.007392	26
62	3	0.007105	27
16	32	0.007105	28
26	13	0.007105	29
13	34	0.006522	30
91	1	0.006264	31
41	7	0.006264	32
53	4	0.006264	33
69	2	0.006264	34
4	140	0.006264	35
100	1	0.005412	36
76	1	0.005412	37
55	4	0.004744	38
35	8	0.004546	39
96	1	0.004546	40
12	39	0.004546	41
24	13	0.004546	42
51	4	0.003669	43
38	8	0.003515	44
54	4	0.002778	45
44	6	0.002778	46
78	1	0.002655	47
68	2	0.002655	48
77	1	0.001783	49
10	49	0.001783	50
64	3	0.001783	51

70	2	0.001697	52
17	30	0.000898	53
57	4	0.000898	54
28	12	0.000898	55
25	13	0.000843	56
135	1	0	57
115	1	0	58
114	1	0	59
113	1	0	60
112	1	0	61
15	33	0	62
111	1	0	63
117	1	0	64
110	1	0	65
109	1	0	66
108	1	0	67
107	1	0	68
106	1	0	69
8	68	0	70
105	1	0	71
104	1	0	72
116	1	0	73
120	1	0	74
118	1	0	75
119	1	0	76
134	1	0	77
133	1	0	78
132	1	0	79
131	1	0	80
130	1	0	81
129	1	0	82
128	1	0	83
127	1	0	84
126	1	0	85
125	1	0	86
124	1	0	87

123	1	0	88
122	1	0	89
121	1	0	90
97	1	0	91
103	1	0	92
32	9	0	93
42	7	0	94
92	1	0	95
23	14	0	96
71	2	0	97
37	8	0	98
18	25	0	99
74	2	0	100
67	2	0	101
86	1	0	102
21	18	0	103
34	8	0	104
36	8	0	105
83	1	0	106
61	3	0	107
39	7	0	108
52	4	0	109
6	109	0	110
75	1	0	111
22	17	0	112
43	6	0	113
45	6	0	114
9	59	0	115
58	4	0	116
33	9	0	117
72	2	0	118
101	1	0	119
20	22	0	120
87	1	0	121
59	4	0	122
2	255	0	123

5	127	0	124
98	1	0	125
11	44	0	126
19	24	0	127
95	1	0	128
94	1	0	129
60	3	0	130
99	1	0	131
7	86	0	132
93	1	0	133
79	1	0	134
14	34	0	135

Table A4: COR-SHIPS model feature importance and its ranking.

Importance score	Variable	Ranking
0.0362	BD12	1
0.0217	DTL	2
0.0207	CFLX	3
0.0206	SHRD	4
0.0205	G150	5
0.0204	jd	6
0.0199	VMAX	7
0.0199	IRM1_5	8
0.0191	PW08	9
0.019	VMPI	10
0.0187	SHTD	11
0.0183	IR00_12	12
0.018	HE07	13
0.0177	MTPW_1	14
0.0177	XD18	15
0.0175	SHTS	16
0.0173	PW14	17
0.0172	TWXC	18
0.0168	R000	19
0.0167	V300	20

0.0165	OAGE	21
0.0162	PSLV_1	22
0.0161	Z850	23
0.0161	SHRS	24
0.0157	SDDC	25
0.0156	VVAC	26
0.0156	PCM3	27
0.0154	TGRD	28
0.0153	T150	29
0.0153	CD26	30
0.0152	TADV	31
0.0151	V850	32
0.0148	PSLV_4	33
0.0145	PSLV_3	34
0.0145	REFC	35
0.0142	RD26	36
0.014	MTPW_19	37
0.0138	ND20	38
0.0138	XDML	39
0.0137	PENV	40
0.0137	EPSS	41
0.0136	G200	42
0.0134	IR00_3	43
0.0131	D200	44
0.013	NTFR	45
0.0124	T250	46
0.0124	O500	47
0.0124	IR00_20	48
0.012	NSST	49
0.0119	IRM1_16	50
0.0119	TLAT	51
0.0115	E000	52
0.0112	IRM3_17	53
0.0112	IRM3_11	54
0.0109	HIST_1	55
0.0109	G250	56

0.0109	RHMD	57
0.0106	NDFR	58
0.0104	IR00_17	59
0.0099	IRM1_17	60
0.0098	NOHC	61
0.0089	PEFC	62
0.0083	IR00_14	63
0.0075	IRM3_14	64
0.0063	PCM1	65
0.0057	IRM1_14	66
0.0052	NDTX	67
0.0043	HIST_8	68
0.0041	XDTX	69
0.0031	PC00	70
0.0023	IRM3_19	71
0.0019	HIST_15	72

Table A5: LLE-SHIPS model feature importance score and its ranking.

Importance score	Variable	Ranking
0.0188	BD12	1
0.0167	VMAX	2
0.0138	DTL	3
0.013	SHRD	4
0.0115	TWXC	5
0.0113	G150	6
0.0113	VMPI	7
0.0113	REFC	8
0.0111	TGRD	9
0.0107	IRM1_5	10
0.0107	IR00_12	11
0.0105	V300	12
0.0105	VVAC	13
0.0103	G200	14
0.0096	PEFC	15
0.0096	MTPW_1	16

0.0095	XDTX	17
0.0095	PSLV_1	18
0.0095	T150	19
0.0094	CFLX	20
0.0094	HIST_2	21
0.0092	HE07	22
0.0091	SHTS	23
0.0089	PSLV_3	24
0.0085	SHTD	25
0.0085	G250	26
0.0085	CD26	27
0.0082	Ile84	28
0.0082	EPSS	29
0.0077	R000	30
0.0076	SDDC	31
0.0075	IRM3_19	32
0.0075	RD26	33
0.0074	PW08	34
0.0074	SHRS	35
0.0074	NDTX	36
0.0073	Ile75	37
0.0073	jd	38
0.0073	TADV	39
0.0072	NDFR	40
0.0072	E000	41
0.0072	HIST_9	42
0.0072	PSLV_4	43
0.0071	MTPW_19	44
0.007	IRM1_16	45
0.007	PW14	46
0.007	OAGE	47
0.007	Ile78	48
0.0068	XD18	49
0.0068	Ile1	50
0.0066	Ile49	51
0.0066	Ile24	52

0.0065	PCM1	53
0.0065	ND20	54
0.0064	Ile2	55
0.0064	IR00_20	56
0.0063	NSST	57
0.0062	Z850	58
0.0062	NTFR	59
0.0061	IR00_14	60
0.0061	NOHC	61
0.0061	Ile89	62
0.006	IRM3_14	63
0.006	Ile71	64
0.006	Ile3	65
0.006	Ile52	66
0.0059	Ile19	67
0.0059	Ile51	68
0.0059	Ile72	69
0.0059	IR00_17	70
0.0059	Ile4	71
0.0058	O500	72
0.0058	Ile53	73
0.0058	V850	74
0.0058	TLAT	75
0.0057	Ile66	76
0.0057	Ile30	77
0.0056	RHMD	78
0.0056	Ile60	79
0.0056	Ile16	80
0.0055	IR00_3	81
0.0055	Ile76	82
0.0055	Ile57	83
0.0054	Ile69	84
0.0054	Ile54	85
0.0054	PC00	86
0.0053	Ile8	87
0.0052	Ile77	88

0.0052	IRM3_11	89
0.0052	IRM1_14	90
0.0052	Ile33	91
0.0051	Ile10	92
0.0051	IRM1_17	93
0.0051	Ile55	94
0.0051	Ile56	95
0.0051	T250	96
0.005	Ile26	97
0.005	Ile81	98
0.005	Ile73	99
0.0049	XDML	100
0.0049	Ile13	101
0.0049	D200	102
0.0048	Ile17	103
0.0048	Ile37	104
0.0048	Ile21	105
0.0048	Ile20	106
0.0048	Ile48	107
0.0048	Ile9	108
0.0047	Ile7	109
0.0047	Ile39	110
0.0047	Ile29	111
0.0046	Ile11	112
0.0046	IRM3_17	113
0.0046	Ile44	114
0.0046	Ile12	115
0.0046	Ile42	116
0.0045	Ile27	117
0.0045	Ile74	118
0.0045	Ile22	119
0.0044	Ile62	120
0.0044	Ile82	121
0.0044	PCM3	122
0.0044	Ile47	123
0.0044	Ile32	124

0.0043	Ile43	125
0.0043	Ile80	126
0.0043	Ile83	127
0.0043	Ile31	128
0.0042	Ile6	129
0.0042	Ile15	130
0.0042	Ile70	131
0.0042	Ile63	132
0.0042	Ile38	133
0.0042	Ile46	134
0.0042	Ile23	135
0.0041	Ile34	136
0.0041	PENV	137
0.0041	Ile58	138
0.004	Ile36	139
0.0039	Ile59	140
0.0039	Ile68	141
0.0039	Ile65	142
0.0039	Ile25	143
0.0039	Ile28	144
0.0038	Ile45	145
0.0038	Ile14	146
0.0038	Ile64	147
0.0038	Ile61	148
0.0037	Ile41	149
0.0037	Ile85	150
0.0037	Ile90	151
0.0036	Ile88	152
0.0036	Ile79	153
0.0035	Ile87	154
0.0035	Ile5	155
0.0035	Ile35	156
0.0034	Ile18	157
0.0034	Ile67	158
0.0034	Ile86	159
0.0032	Ile50	160

0.003	Ile40	161
0.0018	HIST_16	162

Table A6: DL-SHIPS model feature importance score, and its ranking.

Importance score	Variable	Ranking
0.0197	BD12	1
0.0176	VMAX	2
0.0148	SHRD	3
0.0144	DTL	4
0.0137	IRM1_5	5
0.0133	o31	6
0.0131	G150	7
0.0131	q7	8
0.0129	u3	9
0.0129	q4	10
0.0129	G200	11
0.0127	vo3	12
0.0124	REFC	13
0.0122	vo5	14
0.012	vo8	15
0.012	PEFC	16
0.0118	d3	17
0.0116	CFLX	18
0.0116	PSLV_3	19
0.0114	T150	20
0.0114	jd	21
0.0114	R000	22
0.0112	TWXC	23
0.0112	u8	24
0.0112	PW08	25
0.0112	q3	26
0.0112	XDTX	27
0.0109	CD26	28
0.0109	q8	29
0.0107	pv3	30

0.0107	v4	31
0.0105	r1	32
0.0101	u1	33
0.0099	q5	34
0.0099	IR00_12	35
0.0099	vo4	36
0.0097	HE07	37
0.0097	u6	38
0.0097	q2	39
0.0094	r6	40
0.0094	vo6	41
0.0094	MTPW_1	42
0.0092	u2	43
0.0092	r4	44
0.0092	pv7	45
0.0092	pv6	46
0.009	PSLV_1	47
0.009	TADV	48
0.0088	v8	49
0.0088	HIST_2	50
0.0088	VMPI	51
0.0088	V300	52
0.0088	SHRS	53
0.0086	VVAC	54
0.0086	MTPW_19	55
0.0086	v5	56
0.0082	t1	57
0.0082	RD26	58
0.0082	SDDC	59
0.0082	q6	60
0.0082	O500	61
0.0082	v7	62
0.0082	IRM3_11	63
0.0079	E000	64
0.0079	PW14	65
0.0077	z2	66

0.0077	G250	67
0.0075	pv1	68
0.0075	cc1	69
0.0075	XDML	70
0.0075	pv8	71
0.0073	vo1	72
0.0073	ciwc1	73
0.0073	v3	74
0.0073	SHTS	75
0.0073	v6	76
0.0071	ciwc2	77
0.0071	w1	78
0.0071	IRM3_19	79
0.0071	IR00_17	80
0.0069	Z850	81
0.0069	SHTD	82
0.0069	NOHC	83
0.0067	OAGE	84
0.0064	XD18	85
0.0064	IR00_3	86
0.0064	IRM1_16	87
0.0064	PSLV_4	88
0.0062	NTFR	89
0.0062	HIST_9	90
0.006	ND20	91
0.006	IR00_14	92
0.0058	IRM3_17	93
0.0058	EPSS	94
0.0058	clwc2	95
0.0058	D200	96
0.0058	V850	97
0.0056	PC00	98
0.0056	r8	99
0.0054	u5	100
0.0054	NDFR	101
0.0052	PCM1	102

0.0052	NSST	103
0.0052	PENV	104
0.0052	TGRD	105
0.0047	IRM3_14	106
0.0047	IR00_20	107
0.0047	T250	108
0.0047	RHMD	109
0.0047	IRM1_14	110
0.0047	IRM1_17	111
0.0045	cc2	112
0.0034	NDTX	113
0.0034	r7	114
0.0032	TLAT	115
0.0024	PCM3	116
0.0015	HIST_16	117
0.0006	r2	118
0	r3	119
0	r5	120

3206
3207
3208
3209
3210

REFERENCES

Asif, A., M. Dawood, B. Jan, J. Khurshid, M. DeMaria and F. Minhas, 2020: PHURIE: hurricane intensity estimation from infrared satellite imagery using machine learning. *Neural Comput & Applic*, **32,** 4821–4834.

Akbani, R., S. Kwek, and N. Japkowicz, 2004: Applying support vector machines to imbalanced datasets. In *European conference on machine learning*. Springer, Berlin, Heidelberg, 39-50.

Anyfantis, D., M. Karagiannopoulos, S. Kotsiantis, and P. Pintelas, 2007: Robustness of learning techniques in handling class noise in imbalanced datasets. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer, Boston, MA, 21-28.

Astier, N. , M. Plu, and C. Claud, 2015: Associations between tropical cyclone activity in the Southwest Indian Ocean and El Niño Southern Oscillation. Atmos. Sci. Lett., **16**, 506-511.

Batista, G. E., R. C. Prati, and M. C. Monard, 2004: A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1), 20-29.

Berrisford, P., D. Dee, P. Poli, R. Brugge, K. Fielding, M. Fuentes, P. Kallberg, S. Kobayashi, S. Uppala, and A. Simmons, 2011: The ERA-Interim archive, version 2.0.

Berger, E, 2016: The European forecast model already kicking America's butt just improved. *Ars Technica*, 11 March 2016. Retrieved 16 August 2016.

Breiman, L., 2001: Random forests. *Machine learning*, ***45***(1), 5-32.

Breiman, L., 2017: Classification and regression trees. Routledge.

Bunkhumpornpat, C., K. Sinapiromsaran, and C. Lursinsap, 2009: Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In *Pacific-Asia conference on knowledge discovery and data mining*. Springer, Berlin, Heidelberg, 475-482.

Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, 2015: Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 1–9.

Cangialosi, J. P., and J. L. Franklin, 2017: 2016 National Hurricane Center Forecast Verification Report. National Hurricane Center.

Castro, C. L., and A. P. Braga, 2013: Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data. *IEEE transactions on neural networks and learning systems*, *24*(6), 888-899

Chawla, N. V., K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, 2002: SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, **16**, 321-357.

Chawla, N. V., N. Japkowicz, and A. Kotcz, 2004: Special issue on learning from imbalanced data sets. *ACM SIGKDD explorations newsletter*, **6**(1): 1–6.

Chen, T., and C. Guestrin, 2016: Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 785-794.

Cohen, W. W., 1995: Fast effective rule induction. In *Machine Learning Proceedings 1995*, 115-123.

Cummings, J. A., 2005: Operational multivariate ocean data assimilation. *Quarterly Journal of the Royal Meteorological Society*, **131**(613), 3583-3604.

Dee, D. P., S. M. Uppala, A. J. Simmons, P. Berrisford, P. Poli, S. Kobayashi, U. Andrae, M.A. Balmaseda, G. Balsamo, P. Bauer, P. Bechtold, A. C. M. Beljaars, L. van de Berg, J. Bidlot, N. Bormann, C. Delsol, R. Dragani, M. Fuentes, A. J. Geer, L. Haimberger, S. B. Healy, H. Hersbach, E.V. Hólm, L. Isaksen, P. Kållberg, M. Köhler, M. Matricardi, A. P. McNally, B. M. Monge-Sanz, J. -J. Morcrette, B. -K. Park, C. Peubey, P. de Rosnay, C. Tavolato, J. -N. Thépaut, and F. Vitart, 2011: The ERA-Interim reanalysis: Configuration and performance of the data assimilation system. *Quarterly Journal of the royal meteorological society*, **137**, 553-597.

DeMaria, M., 2009: A simplified dynamical system for tropical cyclone intensity prediction. *Monthly Weather Review*, **137**(1), 68-82.

DeMaria, M., and J. Kaplan, 1994: A statistical hurricane intensity prediction scheme (SHIPS) for the Atlantic basin. *Weather and Forecasting*, **9**(2), 209-220.

DeMaria, M., and J. Kaplan, 1999: An Updated Statistical Hurricane Intensity Prediction Scheme (SHIPS) for the Atlantic and Eastern North Pacific Basins Mark. *Weather and Forecasting*, **14**(3), 326–337.

DeMaria, M., J. A. Knaff, and C. Sampson, 2007: Evaluation of long-term trends in tropical cyclone intensity forecasts. *Meteorology and Atmospheric Physics*, **97**(1-4), 19.

DeMaria, M., M. Mainelli, L. K. Shay, J. A. Knaff, and J. Kaplan, 2005: Further improvements to the statistical hurricane intensity prediction scheme (SHIPS). *Weather and Forecasting*, **20**(4), 531-543.

Dempster, A. P., N. M. Laird, and D. B. Rubin, 1977: Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society*. Series B (methodological), **39**(1), 1-22

Enz, R., P. Zimmerli, and S. Schwarz, 2009: Natural catastrophes and man-made disasters in 2008: North America and Asia suffer heavy losses. *National Emergency Training Center*.

Ferrara, M., F. Groff, Z. Moon, K. Keshavamurthy, S. M. Robeson, and C. Kieu, 2017: Large-scale control of the lower stratosphere on variability of tropical cyclone intensity. *Geophysical Research Letters*, **44**(9), 4313-4323.

Fraley, C., and A. E. Raftery, 1998: How many clusters? Which clustering method? Answers via model-based cluster analysis. *The computer journal*, **41**(8), 578-588.

Freund, Y., and L. Mason, 1999: The alternating decision tree learning algorithm. In *icml*, **99**, 124-133.

Friedl, M. A., and C. E. Brodley, 1997: Decision tree classification of land cover from remotely sensed data. *Remote Sensing of Environment*, **61**(3), 399-409.

Friedman, J., T. Hastie, and R. Tibshirani, 2001: The elements of statistical learning. *New York: Springer series in statistics*, **1**, 337-387.

Galar, M., A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, 2012: A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, **42**(4), 463-484

Géron, A., 2017: Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems. *O'Reilly Media, Inc*.

Ginsburg, S. B., G. Lee, S. Ali, and A. Madabhushi, 2016: Feature importance in nonlinear embeddings (FINE): applications in digital pathology. *IEEE transactions on medical imaging*, **35**(1), 76-88.

Gogna, A., and A. Majumdar, 2019: Discriminative Autoencoder for Feature Extraction: Application to Character Recognition. *Neural Processing Letters*, **49**(3), 1723-1735.

Gray, W. M., and L. R. Brody, 1967: Global view of the origin of tropical disturbances and storms. *Colorado State University, Department of Atmospheric Science.*

Han, H., W. Y. Wang, and B. H. Mao, 2005: Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In *International Conference on Intelligent Computing*. Springer, Berlin, Heidelberg, 878-887.

Hinton, G. E., N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, 2012: Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Hodgkin, A. L., and A. F. Huxley, 1952: A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, **117**(4), 500-544.

Holmes, G., A. Donkin, and I. H. Witten, 1994: Weka: A machine learning workbench. In *Proceedings of ANZIIS'94-Australian New Zealnd Intelligent Information Systems Conference.* IEEE. 357-361.

Holyoak, K. J., 1987: Parallel distributed processing: explorations in the microstructure of cognition. *Science*, **236**, 992-997.

Hotelling, H., 1933: Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, **24**(6), 417.

James, G., D. Witten, T. Hastie, and R. Tibshirani, 2013: *An Introduction to Statistical Learning*. Springer.

Japkowicz, N., 2000: The class imbalance problem: Significance and strategies. In *Proceeding of the Int'l Conf. on Artificial Intelligence.*

Jarvinen, B. R., and C. J. Neumann, 1979: Statistical forecasts of tropical cyclone intensity for the North Atlantic basin.

Jo, T., and N. Japkowicz, 2004: Class imbalances versus small disjuncts. *ACM Sigkdd Explorations Newsletter*, **6**(1), 40-49

Kaplan, J., and M. DeMaria, 2003: Large-scale characteristics of rapidly intensifying tropical cyclones in the North Atlantic basin. *Weather and Forecasting*, **18**(6), 1093-1108.

Kaplan, J., M. DeMaria, and J. A. Knaff, 2010: A revised tropical cyclone rapid intensification index for the Atlantic and eastern North Pacific basins. *Weather and Forecasting*, **25**(1), 220-241.

Kaplan, J., C. M. Rozoff, M. DeMaria, C. R. Sampson, J. P. Kossin, C. S. Velden, J. J. Cione, J. P. Dunion, J. A. Knaff, J. A. Zhang, J. F. Dostalek, J. D. Hawkins, T. F. Lee, and J. E. Solbrig, 2015: Evaluating environmental impacts on tropical cyclone rapid intensification predictability utilizing statistical models. *Weather and Forecasting*, **30**(5), 1374-1396.

Karpathy, A., G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, 2014: Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 1725-1732.

Kidder, S. Q., and A. S. Jones, 2007: A blended satellite total precipitable water product for operational forecasting. *Journal of Atmospheric and Oceanic Technology*, **24**(1), 74-81.

Knaff, J. A., DeMaria, M., Sampson, C. R., & Gross, J. M. (2003). Statistical, 5-day tropical cyclone intensity forecasts derived from climatology and persistence. *Weather and Forecasting*, **18**(1), 80-92.

Knaff, J. A., C. R. Sampson, and M. DeMaria, 2005: An operational statistical typhoon intensity prediction scheme for the western North Pacific. *Weather and Forecasting*, **20**(4), 688-699.

Knaff, J. A., T. A. Cram, A. B. Schumacher, J. P. Kossin, and M. DeMaria, 2008: Objective identification of annular hurricanes. *Weather and Forecasting*, **23**(1), 17–88.

Kohavi, R., 1995: A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, **14**(2), 1137-1145.

Kohavi, R., 1996: Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid. In *KDD*, **96**, 202-207.

Krawczyk, B., M. Woźniak, and G. Schaefer, 2014: Cost-sensitive decision tree ensembles for effective imbalanced classification. *Applied Soft Computing*, **14**, 554-562.

Krizhevsky, A., I. Sutskever, and G. E. Hinton, 2012: Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097-1105.

Kurihara, Y., R. E. Tuleya, and M. A. Bender, 1998: The GFDL hurricane prediction system and its performance in the 1995 hurricane season. *Monthly weather review*, **126**(5), 1306-1322.

Labib, K., and V. R. Vemuri, 2006: An application of principal component analysis to the detection and visualization of computer network attacks. In *Annales des télécommunications*, Springer-Verlag, **61**(1-2), 218-234.

Lai, S., L. Xu, K. Liu, and J. Zhao, 2015: Recurrent Convolutional Neural Networks for Text Classification. In *AAAI*, **333**, 2267-2273.

Landsea, C. W. and J. L. Franklin, 2013: Atlantic Hurricane Database Uncertainty and Presentation of a New Database Format. *Monthly Weather Review*, **141**, 3576-3592.

Landwehr, N., M. Hall, and E. Frank, 2005: Logistic model trees. *Machine learning*, **59**(1-2), 161-205.

Last, F., G. Douzas, and F. Bacao, 2017: Oversampling for Imbalanced Learning Based on K-Means and SMOTE. *arXiv preprint arXiv*:1711.00837.

Li, H., and P. Ralph, 2019: Local PCA shows how the effect of population structure differs along the genome. *Genetics*, **211**(1), 289-304.

Li, J., S. Pan, Y. Chen, and Y. Pan, 2017: Assessment of tropical cyclones in ECMWF reanalysis data over Northwest Pacific Ocean. In *The 27th International Ocean and Polar Engineering Conference*. International Society of Offshore and Polar Engineers

Liu, Y., A. An, and X. Huang, 2006: Boosting prediction accuracy on imbalanced datasets with SVM ensembles. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, Berlin, Heidelberg, 107-118.

Liu, Y., E. Racah, J. Correa, A. Khosrowshahi, D. Lavers, K. Kunkel, M. Wehner, and W. Collins, 2016: Application of deep convolutional neural networks for detecting extreme weather in climate datasets, *arXiv preprint arXiv:1605.01156*.

López, V., S. del Río, J. M. Benítez, and F. Herrera, 2015: Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data. *Fuzzy Sets and Systems*, **258**, 5-38.

Lundberg, S. M., and S. I. Lee, 2017: A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, 4765-4774.

Martinez, J., M. M. Bell, R. F. Rogers, and J. D. Doyle, 2019: Axisymmetric Potential Vorticity Evolution of Hurricane Patricia (2015). *J. Atmos. Sci.*, **76**, 2043–2063, https://doi.org/10.1175/JAS-D-18-0373.1.

Merrill, R. T., 1987: An experiment in the statistical prediction of tropical cyclone intensity change. In *17th Conf. Hurricanes and Tropical Meteorology*, Hurricanes and Tropical Meteorology, Miami, FL. *Amer. Meteor. Soc.*, 302-304.

Merrill, R. T. (1988). Environmental influences on hurricane intensification. *Journal of the Atmospheric Sciences*, **45**(11), 1678-1687.

Miller, D., 2007: NEW ADVANCED HURRICANE MODEL AIDS NOAA FORECASTERS. http://www.noaanews.noaa.gov/stories2007/s2885.html. Accessed 29 May 2017

Molinari, J., and D. Vollaro, 1989: External influences on hurricane intensity. Part I: Outflow layer eddy angular momentum fluxes. *Journal of the Atmospheric Sciences*, **46**(8), 1093-1105.

Molnar, C., 2018: Interpretable machine learning: A guide for making black box models explainable. *Christoph Molnar, Leanpub*.

Nguyen, H. M., E. W. Cooper, and K. Kamei, 2011: Borderline over-sampling for imbalanced data classification. *International Journal of Knowledge Engineering and Soft Data Paradigms*, **3**(1), 4-21.

NHC (National Hurricane Center), 2019: *NHC Track and Intensity Models*. Web site: http://www.nhc.noaa.gov/modelsummary.shtml. Last accessed on August 18, 2020.

Pacific Disaster Center. (n.d.): Tropical Cyclones. Retrieved November 20, 2017, from http://www.pdc.org/resources/natural-hazards/tropical-cylcones/.

Pearson, K., 1901: LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, **2**(11), 559-572.

Piñeros, M. F., E. A. Ritchie, and J. S. Tyo, 2011: Estimating tropical cyclone intensity from infrared image data. *Weather and Forecasting*, **26**(5), 690–698.

Platt, J. C., 1999: 12 fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods*, 185-208.

Qian, Y., C. Liang, S. Peng, S. Chen, and S. Wang, 2016: A Horizontal Index for the Influence of Upper-Level Environmental Flow on Tropical Cyclone Intensity. *Weather and Forecasting*, **31**, 237–253, https://doi.org/10.1175/WAF-D-15-0091.1

Quinlan, J. R., 2014: C4. 5: programs for machine learning. *Elsevier*.

Racah, E., C. Beckham, T. Maharaj, and C. Pal, 2016: Semi-Supervised Detection of Extreme Weather Events in Large Climate Datasets. *arXiv preprint arXiv:1612.02095*.

Reid, W., 1846: An Attempt to Develop the Law of Storms by Means of Facts, Arranged According to Place and Time: And Hence to Point Out a Cause for the Variable Winds, with the View to Practical Use in Navigation. *Illustrated by Charts and Wood Cuts*. John Weale.

Rennick, M. A., 1999: Performance of the Navy's tropical cyclone prediction model in the western North Pacific basin during 1996. *Weather and Forecasting*, **14**(3), 297-305.

Reynolds, D. A., T. F. Quatieri, and R. B. Dunn, 2000: Speaker verification using adapted Gaussian mixture models. *Digital signal processing*, **10** (1-3), 19-41.

Rhome, J. R., 2007: Technical summary of the National Hurricane Center track and intensity models.

Ribeiro, M. T., S. Singh, and C. Guestrin, 2016: " Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135-1144.

Ritchie EA, G. Valliere-Kelley, M. F. Piñeros, J. S. Tyo, 2012: Tropical cyclone intensity estimation in the North Atlantic Basin using an improved deviation angle variance technique. *Weather and Forecasting*, **27**(5): 1264–1277.

Ron, U., 2000: A practical guide to swap curve construction. Ottawa: Bank of Canada, **2000** (17).

Roweis, S. T., and L. K. Saul, 2000: Nonlinear dimensionality reduction by locally linear embedding. *science*, **290**(5500), 2323-2326.

Rozoff, C. M., and J. P. Kossin, 2011: New probabilistic forecast models for the prediction of tropical cyclone rapid intensification. *Weather and Forecasting*, **26**(5), 677-689.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Sampson, C. R., J. L. Franklin, J. A. Knaff, and M. DeMaria, 2008: Experiments with a simple tropical cyclone intensity consensus. *Weather and Forecasting*, **23**(2), 304-312.

Santos, d. C., M. Gatti, 2014: Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*,69-78.

Schölkopf, B., A. Smola, and K. R. Müller, 1997: Kernel principal component analysis. In *International Conference on Artificial Neural Networks,* Springer, Berlin, Heidelberg, 583-588.

Schölkopf, B., A. Smola, and K. R. Müller, 1998: Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, **10**(5), 1299-1319.

Shahriari, B., K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, 2015: Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, **104**(1), 148-175.

Shaiba, H. and M. Hahsler, 2016: Research Article Applying Machine Learning Methods for Predicting Tropical Cyclone Rapid Intensification Events. *Research Journal of Applied Sciences*, Engineering and Technology **13**(8), 638-651.

Sheets, R. C., 1990: The National Hurricane Center—past, present, and future. *Weather and Forecasting*, **5**(2), 185-232.

SHIPS, 2018a: SHIPS statistical tropical cyclone intensity forecast technique development, developmental data. [Available online at http://rammb.cira.colostate.edu/research/tropical_cyclones/ships/developmental_data.asp.]

SHIPS, 2018b: http://rammb.cira.colostate.edu/research/tropical_cyclones/ships/docs/AL/lsdiaga_1982_2017_sat_ts.dat (a link to the 2018 version of the SHIPS developmental data. [last accessed on February 3, 2020.]

SHIPS, 2018c: http://rammb.cira.colostate.edu/research/tropical_cyclones/ships/docs/ships_predictor_file_2018.doc (a link to the 2018 version of the SHIPS developmental data variables. [last accessed on February 3, 2020.]

Simonyan, K., and A. Zisserman, 2014: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Snoek, J., H. Larochelle, and R. P. Adams, 2012: Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, 2951-2959.

Song, J., X. Huang, S. Qin, and Q. Song, 2016: A bi-directional sampling based on K-means method for imbalance text classification. In *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, 1-5, IEEE.

Stoer, J., and R. Bulirsch, 2013: Introduction to numerical analysis. *Courier Corporation*.

Tallapragada, V., 2014: Performance of the 2013 NCEP Operational HWRF and Plans for 2014 Hurricane Season. Retrieved from https://www.ofcm.noaa.gov/meetings/TCORF/ihc14/presentations/Session4/s04-02tallapragada.pdf

Tan, P.-N., M. Steinbach, and V. Kumar, 2015: Introduction to data mining. *Dorling Kindersley: Pearson*.

Torres-Carrasquillo, P. A., E. Singer, M. A. Kohler, R. J. Greene, D. A. Reynolds, and J. R. Deller Jr, 2002: Approaches to language identification using Gaussian mixture models and shifted delta cepstral features. In *Seventh International Conference on Spoken Language Processing*

Tran, D., L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, 2015: Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, 4489-4497.

Trevor, H., T. Robert, and J. Friedman, 2009: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Science & Business Media.

Tsujino, S., and H. C. Kuo, 2020: Potential Vorticity Mixing and Rapid Intensification in the Numerically Simulated Supertyphoon Haiyan (2013). *Journal of the Atmospheric Sciences*, ***77***(6), 2067-2090.

Volinsky, C. T., and A. E. Raftery, 2000: Bayesian information criterion for censored survival models. *Biometrics*, **56**(1), 256-262.

Wang, Y., Y. Rao, Z. M. Tan, and D. Schönemann, 2015: A statistical analysis of the effects of vertical wind shear on tropical cyclone intensity change over the western North Pacific. *Monthly Weather Review*, **143**(9), 3434-3453.

Wang, Z., 2018: What is the key feature of convection leading up to tropical cyclone formation?. *Journal of the Atmospheric Sciences*, **75**(5), 1609-1629.

Wei, Y., L. Sartore, J. Abernethy, D. Miller, K. Toppin, and M. Hyman, 2018: Deep Learning for Data Imputation and Calibration Weighting. In *JSM Proceedings*, Statistical Computing Section. Alexandria, VA: American Statistical Association, 1121-1131.

Wilks, D. S., 2011: *Statistical Methods in the Atmospheric Sciences*. 3rd ed. Elsevier, 676.

Yang, R., 2016: A Systematic Classification Investigation of Rapid Intensification of Atlantic Tropical Cyclones with the SHIPS Database. *Weather and Forecasting*, **31**(2), 495-513.

Yang, R., J. Tan, and M. Kafatos, 2006: A Pattern Selection Algorithm in Kernel PCA Applications. In *proceedings of the first International Conference on Software and Data Technologies*, Setubal, Portugal, 195-202.

Yang, R., J. Tang, and M. Kafatos, 2007: Improved associated conditions in rapid intensifications of tropical cyclones. *Geophysical Research Letters*, **34**(20).

Yang, R., J. Tang, and D. Sun, 2011: Association rule data mining applications for Atlantic tropical cyclone intensity changes. *Weather and Forecasting*, **26**(3), 337-353.

Zeiler, M. D., D. Krishnan, G. W. Taylor, and R. Fergus, 2010: Deconvolutional networks. In *2010 IEEE Computer Society Conference on computer vision and pattern recognition, IEEE*, 2528-2535.

Zeiler, M. D., G. W. Taylor, and R. Fergus, 2011: Adaptive deconvolutional networks for mid and high level feature learning. In *2011 International Conference on Computer Vision*, IEEE, 2018-2025.

Zeiler, Matthew D, and R. Fergus, 2014: Visualizing and understanding convolutional neural networks. In *Proceedings of the 13th European Conference Computer Vision and Pattern Recognition, Zurich, Switzerland*, 6-12.

BIOGRAPHY

Yijun Wei graduated from Nanshan high School, Mianyang, Sichuan, China, in 2006. He received his Bachelor of Arts from Sichuan University in 2011. He received his Master of Arts in Math and Master of Science in Statistics from University of Michigan – Ann Arbor in 2013.