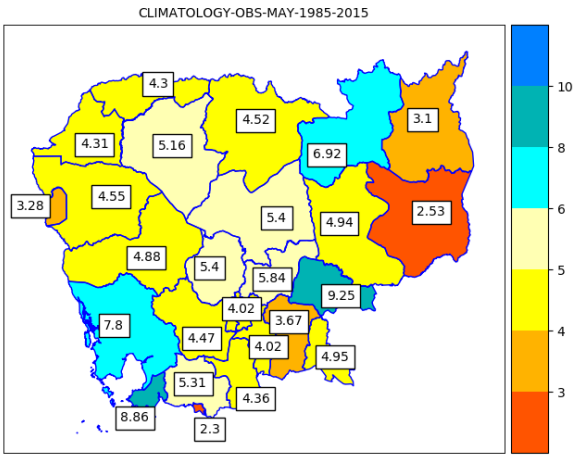
FORECAST CUSTOMIZATION SYSTEM



VERSION 2.0

FORECAST CUSTOMIZATION SYSTEM

User Guide

Version 2.0

Developed for reference Purpose of

Department of Meteorology

Cambodia

1. BACKGROUND

Climate information from the global climate models (GCMs) is intended to represent large scale oceanic and atmospheric phenomena. Utilization of this information for the sectoral application, which generally happens below the sub-national scale, is a challenging task. Several attempts have been made by pioneer institutions to try transforming the global information at the local scale and make it available at its most accurate form to the sectoral users. Climate Prediction tool (CPT) by International Research Institute (IRI), SCHOPIC by Bureau of Meteorology (BOM), CLIK by APCC etc., are, to name a few. Forecast customization System is similar to these tools aimed at providing extended range forecast information at a scale usable to an end-user such as the political/administrative level or based on the homogeneity of climate divisions.

The significance of the multi-model ensemble-based tool is to represent the forecast information in a simplified form in order to enhance the utility of seasonal forecast in sector-specific applications. The tool uses various standard statistical approaches to develop a model for the multi-model ensemble forecast system.

The foremost aspect involved in the experiment forecasting is the input data (rainfall) obtained from models North American Multi-Model Ensemble (NMME). The collaboration with the International Research Institute for Climate and Society (IRI), USA, has facilitated the regular availability of several NMME Models. The model data can be downloaded from the IRI Data Library (http://iridl.ldeo.columbia.edu). These models have a long time series of hindcast runs and real-time forecasts for the current year with different lead times. Seasonal climate forecast from the European center for medium-range weather forecast (ECMWF), Climate forecasting System (CFSv2) from NOAA.

This manual consists procedure of data downloading from the global centers, processing, and forecast result generation. The manual explains in detail about the data download, storing the data, combining the data, interpolating data, uploading observed data, and forecast generation for either province (administrative level, climate zone wise, or at grid point level) for a month or up to 3 months.

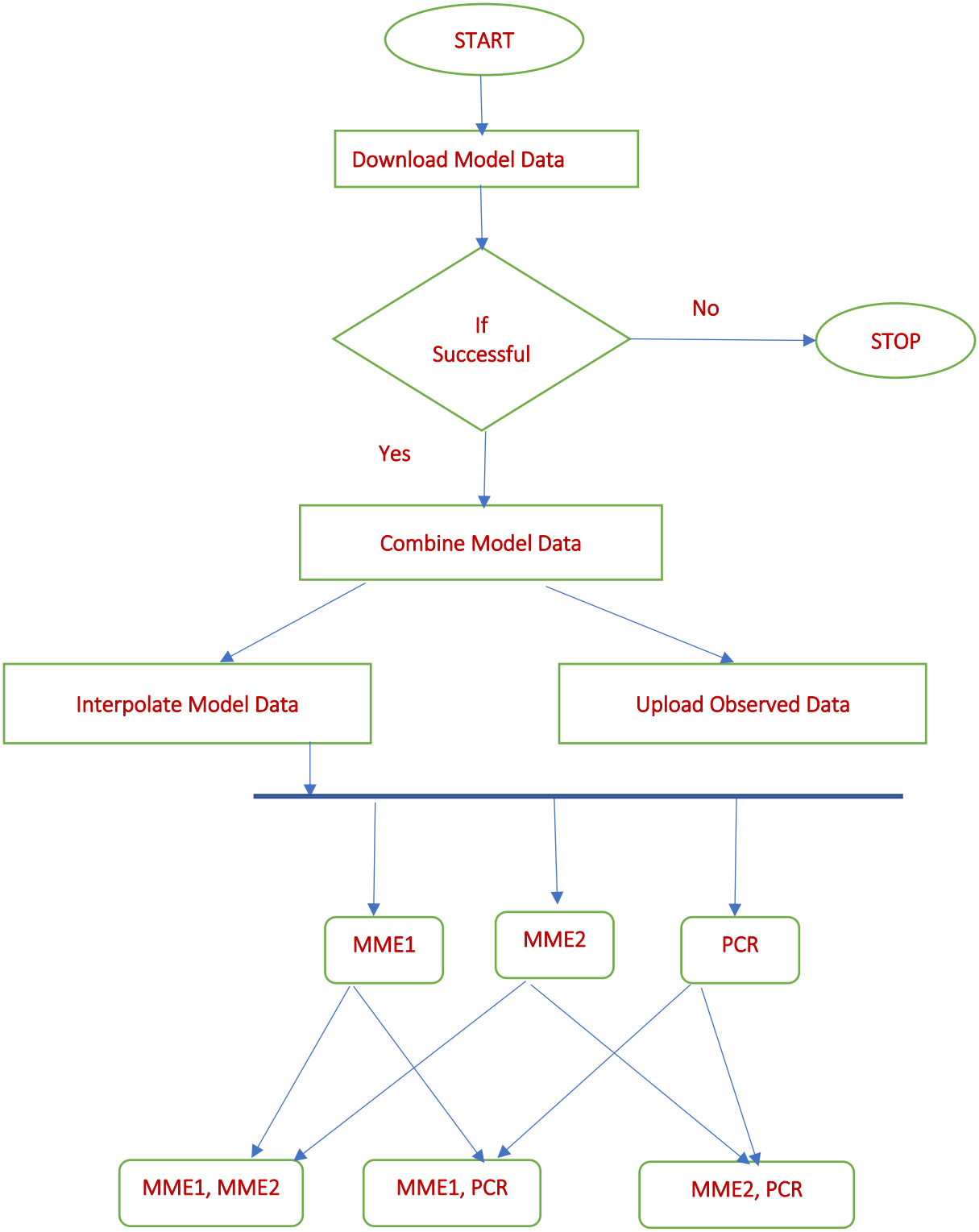


Figure 1. Overall Flow

2. MULTI-MODEL ENSEMBLE TECHNIQUES USED IN FOCUS

This document explains the different forecasting methods incorporated into the FOCUS environment, access to the data repository, and the verification methods in place as well. There are three different MMEs in place.

2.1. SIMPLE MEAN METHOD (SMM)

Suppose F1, F2, F3, Fm are m number of model-simulated results with a hindcast run for t times (years) where each model simulation consists of several numbers of ensemble members. Ensemble members are weighted equally to construct an ensemble mean as the prediction of an individual model to get a single forecast for (t+1) year.

The simplest of all, the SMM scheme, method based on simple averaging of all the individual models. In other words, all the individual member models have been assigned the same weight while carrying out the ensemble average. Therefore, for (t+1) year, take the arithmetic mean of all models individual forecast. There is an assumption in this method that each model is relatively independent and, to some extent. Before using the SMM method, all models forecast are normalized with their long-term mean (climatology) and long-term standard deviation (inter-annual variation). Then the simple mean of all normalized values was calculated. After that, these values were multiplied with observed inter-annual variation and added to the observed climatology for getting the final forecast. The SMM forecast constructed with bias-corrected data and the mathematical formulation is given below:

$$S_t = \bar{O} + \left[\frac{1}{N}\sum_{i=1}^{N}\left(\frac{F_{i,t} - \overline{F_i}}{\sigma_{F_i}}\right)\right] \times \sigma_O$$

Where,

S_t	=	SMM prediction at time t.
$F_{i,t}$	=	ith model forecast at time t.
$\overline{F_i}$	=	Climatology of the ith model forecast.
$\bar{O}$	=	Climatology of observations for the same period.
σ_{F_i}	=	Inter-annual variation of the ith model forecast.
σ_O	=	Inter-annual variation of the observations.
N	=	Total number of models.

This technique is applied to each GCM ensemble mean forecast. In the training datasets, the leave-one-out cross-validation method is used to understand the efficiency of the bias correction method. In the leave-one-out method, the forecasted year is not considered in the training dataset, and the remaining years' data are used to calculate model and observed climatological mean and standard deviation—the same procedure implemented for real-time forecasts.

2.2. WEIGHTED AVERAGE METHOD (WAM) VALUE DECOMPOSITION-BASED MULTIPLE REGRESSION

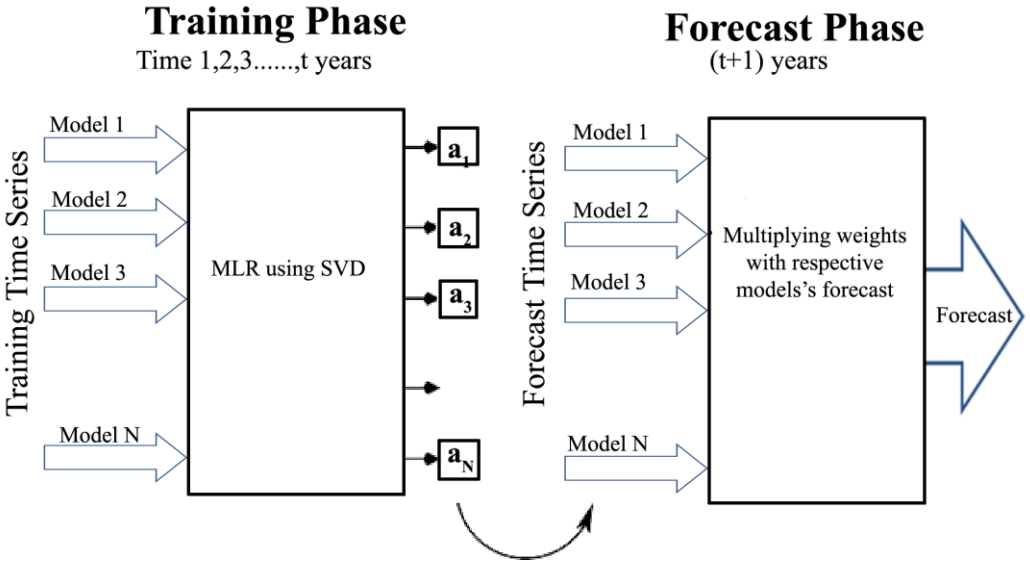
In this method for carrying out a weighted multi-model ensemble mean, multiple linear regression method has been employed. The weighted mean of all GCMs is evaluated using a point-by-point multiple regression method. The covariance matrix becomes singular while calculating the regression coefficients. Therefore, singular value decomposition (SVD) is applied for the computation of these coefficients. The SVD removes the problem of matrix singularity while calculating the regression coefficients. These regression coefficients are evaluated for the training dataset to obtain weights for the GCMs. These weights are used for the calculation of weighted MME. This scheme is also called "Superensemble." In this method, the regression coefficients are obtained using Gauss-Jordan elimination with pivoting. SVD based MME techniques were introduced for finding a robust weighted MME forecast. The WAM forecast constructed with bias-corrected data and can be mathematically expressed as:

$$S_t = \bar{O} + \left[a_i \sum_{i=1}^{N} \left(\frac{F_{i,t} - \bar{F}_i}{\sigma_{F_i}} \right) \right] \times \sigma_O$$

Where

S_t = SMM prediction at time t.
$F_{i,t}$ = i^{th} model forecast at time t.
$\bar{F}_i$ = Climatology of the ith model forecast.
$\bar{O}$ = Climatology of observations for the same period.
σ_{F_i} = Inter-annual variation of the ith model forecast.
σ_O = Inter-annual variation of the observations.
N = Total number of models.

a_i = Regression coefficient obtained by a minimization procedure during the training period between models forecasts F_i's an observation O.



Estimation of a_i

The regression equation between observation and model is given below:

$$O = F \times \beta + \epsilon$$

Where,

$$O_{t\times 1} = \begin{pmatrix} O_1 \\ O_2 \\ \vdots \\ O_t \end{pmatrix} \; F = \begin{bmatrix} F_{11}\, F_{21} \ldots F_{n1} \\ F_{12}\, F_{22} \ldots F_{n2} \\ \ldots\ldots\ldots\ldots \\ \ldots\ldots\ldots\ldots \\ \ldots\ldots\ldots\ldots \\ F_{1t}\, F_{2t} \ldots F_{nt} \end{bmatrix} \; \beta_{t\times 1} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_t \end{pmatrix} \; \epsilon_{t\times 1} = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_t \end{pmatrix}$$

Singular value decomposition (SVD) is a very robust method to solve the singularity problem. Singular value decomposition is a particular type of factorization of a matrix into a product of three matrices, of which the second is a diagonal matrix that has as the entry, entries on its diagonal the singular values of the original matrix.

Let $A(n \times p)$ be a rectangular matrix, then SVD theorem states:

$$A_{n\times p} = U_{n\times n} \,.\, S_{n\times p} \;.\; V^T_{p\times p}$$

Where, $U^T U = I_{n\times n}$, , $V^T V = I_{p\times p}$, and I = Identity matrix. (i.e., I and V are orthogonal).

2.3. PRINCIPAL COMPONENT REGRESSION (PCR)

In this method, the original set of predictor variables are transformed into a new set of variables that are orthogonal (independent) to each other. Moreover, it finds a new set of variables that capture maximum variance from the dataset through a linear combination of the original variables. The problem of multicollinearity can be solved with the use of principal component analysis (PCA), ridge regression (PCR) etc., by retaining only higher modes.

A two-stage procedure has been implemented. Firstly, the predictor variables are screened based on the threshold value of the correlation between predictor (model precipitation) and predictand (observed precipitation). In the second stage, principal component analysis (PCA) is applied, and the rank of PCA is determined based on the correlation between PCA and observations. Finally, PCA based on its rank is adding successively in the regression equation and find the minimum root mean square error for a threshold correlation of that model.

For example, the seasonal predictions of precipitation predictor ($\mathrm{F_{M\times N}}$) and predictand ($\mathrm{O_{M\times 1}}$) data sets are the bias-corrected GCM outputs and the observed precipitation, respectively. Here M is the number of years, and N is the number of predictors (models). These N predictors are correlated with the observation (O), resulting in N correlations. Now predictors correlating 0.2 or greater are screened from the F data set.

Let us assume that P predictors are screened from N predictors. Let the retained F matrix be $\mathrm{F_{M\times P}}$ where$\mathrm{P} \leq \mathrm{N}$. Now, the correlation matrix ($\mathrm{C_{P\times P}} = \mathrm{F'_{M\times P}} \times \mathrm{F_{M\times P}}$) is evaluated from which the eigenvalues ($\lambda_{\mathrm{P}\times 1}$) and eigenvector matrix ($\mathrm{V_{P\times P}}$) are computed. Firstly, the eigenvalues are arranged in decreasing order to identify the first few empirical orthogonal functions (EOFs) that can explain a majority of the variance of the data and then find the eigenvector matrix. Now, the PCs are evaluated as $(\mathrm{Z_{M\times P}} = \mathrm{F'}_{\mathrm{M\times P}} \times \mathrm{F_{M\times P}})$ Again correlation is found between each of the PCs ($\mathrm{Z_{M\times P}}$) and O, and the first three PCs with the highest correlation are selected. Here, it should be noted that the selection of PCs is based on the correlation computed with the observation, and not by their variances. Then forward selection approach is used to select PCs (from three PCs) for regression. At each stage in the selection process, after a new PC is added, a test is made to check if some PCs can be deleted without appreciably increasing the root mean square error (RMSE), and this selection ends when RMSE gets its minimum.

2.3.1. STEPS IN DEVELOPING THE PCR MODEL

1. As mentioned earlier, the members from different models are treated as predictors (X), and the Observation data is treated as predictand (Y) since the resolution of these

members is different, so they are first gridded/interpolated onto a same resolution to the observation data grid.

2. The interpolated data is then normalized to remove the bias.
3. The average is taken for each of the members.
4. These members are then screened according to their correlations with the observation. A threshold is fixed at 0.2. The predictors, with a correlation greater than 0.2, will be selected first.
5. Then principal components are calculated by first finding the correlation matrix.
6. Next, calculating the eigenvectors of the correlation matrix.

$$Z = X * V$$
$$Y = Z * \beta$$

Here X is a vector of independent variables (predictors), Y is a vector of the dependent variable (predictand), and V is the eigenvector matrix obtained from the correlation matrix of X, Z, the resulting Principal components and the regression coefficients.

7. Again correlation coefficient is found between these principal components (PCs), and the observation and the first three PCs with the highest correlation are selected.
8. Now, these selected PCs will enter the model (2) stepwise, and the model with the least RMSE is selected.
9. The above steps (steps 4 to 7) are repeated for seven different thresholds (0.2 to 0.5 in steps of 0.05), and a simple mean of all these models will be considered as forecast.

3. GENERAL CIRCULATION MODELS (GCMS) USED IN THE TOOL

The analysis of General Circulation Models (GCMs) from the IRI data library was analyzed. The GCMs with better results are identified for the use in the FOCUS system tool. The details of the GCMs identified are stated in Table-1. These are the best performing GCMs for months/seasons based on the initial statistical analysis conducted. Climatology, anomaly, the standard deviation were computed for the models, and the observations and comparative analysis were done for models and observations analysis, i.e., correlation coefficient, root mean square error.

Table-1: Global Circulation Models Information

Sr. No.	Model	Source	Resolution	Ensemble Members	Type	References
1	CFS v2(NCEP)	NMME	(T126) (0.9°x 0.9°)	24	Fully coupled	Saha et al. (2014)
2	COLA-RSMAS-CCSM4	NMME	(T106)(1.12°x1.12°)	06	Anomaly-Coupled	Kirtman et al. (2014)
3	GFDL-CM2p1-aer04	NMME	(T42)(2.7° x2.8°)	10	Fully coupled	Kirtman et al. (2014)
4	GFDL-CM2p5-FLOR-A06	NMME	(T42)(2.7° x2.8°)	12	Fully coupled	Kirtman et al. (2014)

5	GFDL-CM2p5-FLOR-B01 (GFDLB01)	NMME	(T42)(2.7° x2.8°)	12	Fully coupled	Kirtman et al. (2014)
6	NASA_GEOS_S2S	NMME	(0.5°×0.5°)	4/10	Fully Coupled	Borovikov et al (2017)
7	CanCM4i	NMME	T63(1.4°x0.94°)	10	Anomaly-Coupled	Merrifield et al. (2013)
8	CanSIPSv2	NMME	T63(1.4°x0.94°)	20	Anomaly-Coupled	H. Lin et. al. (2019)
9	ECMWF	Copernicus	(T159)(0.75°x 0.75°)	15	Fully coupled	Magnusson and Kallen (2013)

The General Circulation Models (GCMs) data from the IRI data library has downloaded and repository prepared for FOCUS at RCC, IMD Pune. Details of GCMs data available in FOCUS's repository for parameters, e.g., Precipitations, Surface Temperature, and Sea Surface Temperature (SST) at RCC are mentioned in Table-2.

Table-2: Global Circulation Models Details which are available in RCC, IMD, Pune FOCUS's Repository

Sr. No.	**Model**	**Source**	**Resolution**	**Ensemble Members**	**Period**
1	COLA-RSMAS-CCSM4	IRI/NMME	1°x1°	06	Jan-1982 to May-2020
2	GFDL-CM2p1-aer04	IRI/NMME	1°x1°	10	Jan-1982 to May -2020
3	GFDL-CM2p5-FLOR-A06	IRI/NMME	1°x1°	12	Jan-1982 to May -2020
4	GFDL-CM2p5-FLOR-B01 (GFDLB01)	IRI/NMME	1°x1°	12	Jan-1982 to May -2020
5	CFS v2(NCEP)	IRI/NMME	1°x1°	24	Jan-1982 to May -2020
6	NASA_GEOS_S2S	IRI/NMME	1°x1°	4/10	Jan-1982 to May -2020 (Missing Jan-2017 to Oct-2017)
7	CanCM4i	IRI/NMME	1°x1°	10	Jan-1982 to May -2020 (Missing Jan-2019 to July-2019)
8	CanSIPSv2	IRI/NMME	1°x1°	20	Jan-1982 to May -2020 (Missing Jan-2019 to July-2019)
9	ECMWF	IRI/ EU Copernicus	1°x1°	25/51	Jan-1982 to May -2020 (Missing Jan-2017 to Aug-2017)

4. FORECAST VERIFICATION METHOD

The FOCUS system incorporates several forecast verification methods, which includes verification of both deterministic and probabilistic forecasts. The methods are described in the following section.

4.1. PROBABILISTIC FORECAST VERIFICATION: USING ROC (AREA UNDER THE CURVE)

The WMO guideline for long-range predictions describes that ROC curves are usually constructed to give an insight on hit rates (H), and false alarm rates symbolized by (F). The ROC can be used in forecast verification to measure the ability of the forecasts to distinguish an event from a non-event. For seasonal forecasts with three or more categories, the first problem is to define the "event." One of the classes must be selected as the current category of interest, and an occurrence of this category is known as an event. An observation in any of the other categories is defined as a non-event. No distinction is made as to which of these two categories does occur. So, for example, if below-normal is selected as the event, normal and above-normal are treated equally as non-events. Given this requirement for a binary definition of outcomes, separate ROC graphs can be completed for each category. A measure of discrimination can then be defined to indicate how well the forecasts can distinguish the selected category from the other two categories.

Setting $p_{1,j}$ as the forecast probability for the j^{th} observed event, and $p_{0,i}$ as the forecast probability of an event for the i^{th} non-event, the ROC score, A, can be calculated using;

$$A = \frac{1}{n_0 n_1} \sum_{i=1}^{n_0} \sum_{j=1}^{n_1} I(P_{0,i}, P_{1,j})$$

where n_0 is the number of non-events and n_1 the number of events and the scoring rule

I ($P_{0,i}$, $P_{1,j}$) is defined as;

$$I(P_{0,i}, P_{1,j}) = \begin{cases} 0.0 \; if \; P_{1,j} < P_{0,i} \\ 0.5 \; if \; P_{1,j} = P_{0,i} \\ 1.0 \; if \; P_{1,j} > P_{0,i} \end{cases}$$

The ROC graph represents the ability of the forecasts to identify the events to that of its inability successfully. The chart can be constructed by starting with the forecasts with the highest probabilities. These forecasts should point to the observations that we are most confident are events. A "hit" is if the selected observations are events and the proportion of all events is known as the hit rate (HR), generally represented as:

$$HR = \frac{No.of\ Hits}{No.of\ Event}$$

False alarms are those instances that are incorrectly selected as events while they are non-events in reality. The proportion of non-events incorrectly selected computed as the false-alarm rate (FAR)] and are represented as:

$$FAR = \frac{No.of\ False\ alarms}{No.of\ non-Event}$$

The hit rate and false-alarm rates would be identical, meaning the models have no skills in predicting an event. In case the hit rates are more significant than the false-alarm rate, then the model shows some skills. Different levels of highest probabilities were selected to test the performance, where the hit and false-alarm rates are updated for each of these probabilities. Finally, ROC curves are the hit rates plotted against the false-alarm rates at each of these probability instances.

4.2. DETERMINISTIC FORECAST VERIFICATION

4.2.1. COEFFICIENT OF DETERMINATION (R^2)

This coefficient measures the strength of a linear relationship between two variables, for example, between modeled and observed precipitation and temperature. This coefficient varies from 1 to 0, respectively, for the best to the most inferior result. This is given by the square of correlation coefficient (r) as;

$$r = \frac{n\sum xy - (\sum x)(\sum y)}{\sqrt{n(\sum x^2) - (\sum x)^2}\sqrt{n(\sum y^2) - (\sum y)^2}}$$

Where x is observed rainfall and y is model forecast.

4.2.2. STANDARD DEVIATION

This statistical parameter shows the variability of modeled and observed data to the observed mean or simply the climatology.

$$\sigma = [\frac{1}{N}\sum_{n=1}^{N}(X_n - \overline{X})^2]^{\frac{1}{2}}$$

Where σ and $\vec{x}$ represent standard deviation and mean value respectively

4.2.3. ROOT MEAN SQUARE ERROR (RMSE)

RMSE or the Root Mean Square Error is a measure of the difference (error) between values predicted by a model and the observed values from the environment that is being modeled (Koehler, 2006). RMSE aggregates the residuals from each of the individual differences and generates a single measure of predictive power. The RMSE of a model prediction concerning the estimated variable X_{mod} is defined as the square root of the mean squared error:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(X_{obs,i} - X_{\mathrm{mod},i})^2}{n}}$$

Where X_{obs} = observed values

X_{mod} = modelled values at time/place i.

n = total number of sample datasets

The RMSE can be used to distinguish model performance from a calibration period to a validation period as well as to compare the individual model performance to that of a predictive model or models.

4.2.4. THE CORRELATION COEFFICIENT (R)

Correlation or correlation coefficient indicates the linear relationship between two variables, such as the observed rainfall to the predicted rainfall, in the current context. Pearson's correlation coefficient (Pearson, 1895) is the most widely used coefficient. The correlation coefficient (r) is estimated by dividing the covariance of the observation and the forecast by the product of their standard deviations. With a time-series data of ***n*** observations and ***n*** models, then the Pearson correlation coefficient can be used to estimate the correlation between model and observations.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \cdot \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

The correlation coefficient value of +1 represents the case of a perfect increasing linear relationship. In contrast, -1 represents a decreasing linear relationship, and any values in between ($-1 \leq r \leq 1$) indicate the degree of the linear relationship between model and observations. A correlation coefficient of 0 means the there is no linear relationship between the variables. The coefficient of determination (r^2), which is the square of the correlation coefficient, describes how much of the variance between the two variables is described by the linear fit (Pearson, 1895).

4.2.5. INDEX OF AGREEMENT (D)

Index of Agreement quantifies the agreement between the observed climate to the model climate. This skill falls between 0 and 1 ($0 \leq d \leq 1$), where the closer the value to 1 indicates better agreement of the forecast to observation. The average error, as well as the distance from observed climatology, is quantified simultaneously. The measure provides an inference of better forecast concerning observed climatology with less error.

$$d = 1 - \frac{\sum_{i=1}^{N}(P_i - O_i)^2}{\sum_{i=1}^{N}(\left|P_i - \overline{O}\right| + \left|O_i - \overline{O}\right|)^2}$$

4.2.6. ANOMALY CORRELATION COEFFICIENT (ACC)

The quality of a seasonal forecast as an average over an ensemble can be assessed using the anomaly correlation coefficient (ACC). The question to be answered is how well the ensemble mean '***f***' meets the observed values '***o***' of all years ***'m'***. The equation of the anomaly correlation coefficient is:

$$ACC = \frac{\sum_{m=1}^{M} \mathrm{f}'_{\mathrm{m}} o'_m}{\left[\sum_{m=1}^{M}\left(\mathrm{f}'_{\mathrm{m}}\right)^2 \sum_{m=1}^{M}\left(\mathrm{o}'_{\mathrm{m}}\right)^2\right]^{1/2}}$$

$\mathrm{f}'_{\mathrm{m}} = \mathrm{f}_{\mathrm{m}} - c_m$ Forecast anomaly from climatology* at each grid point (m)

$o'_m = o_m - c_m$ Analysis anomaly

If the ACC value equals one, both data sets are positively correlated, which means a perfect forecast quality. If the ACC value equals minus one, the data sets have an inverse relationship. If the ACC value is zero, there is no correlation between the data.

5. ACCESSING THE WEB-BASED SYSTEM

FOCUS tool is a web-based system designed with Python3.4 (backend) and Microsoft's .Net framework as frontend. The System can be accessed with the web link from any browser as below;

http://focus.rimes.int/Login.aspx

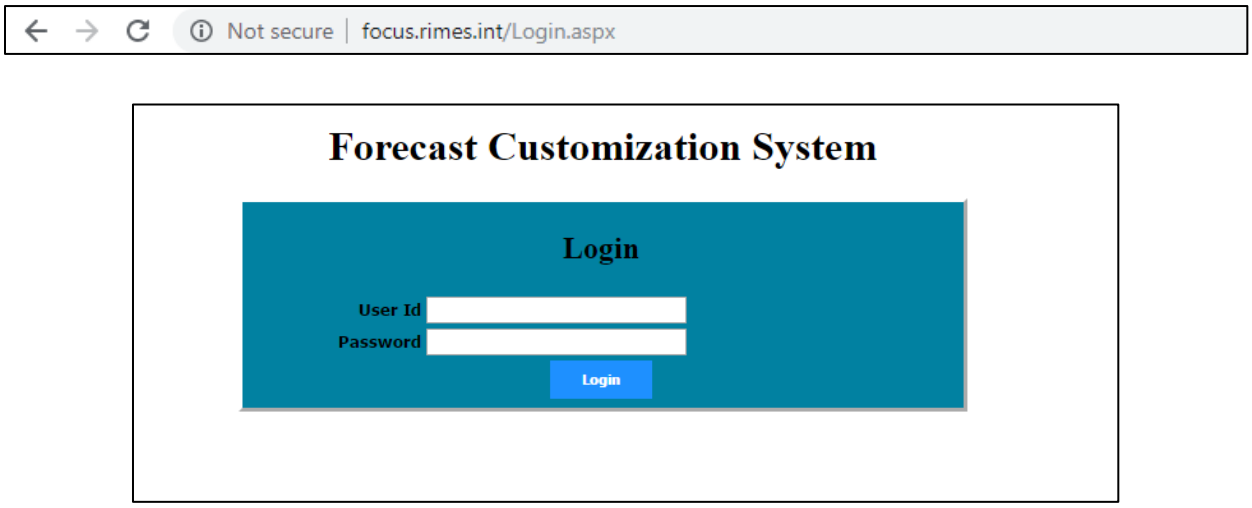


Figure 1: Login Page

Every user trying to access the FOCUS tool must pass the authentication to start using the resources. National met services which are using the FOCUS system, already are provided a set of username and password to log in. User accounts are previously defined and linked to a certain country forecasting system. For example, user "dom" is linked to the Cambodia country forecast. Moreover, the "DMH" user is linked to Myanmar.

Once logged in the System would provide several menu items, which include the; 1.) Forecast generation, 2) Standard forecast verification, 3) analysis of ensemble and spreads 4) View and download all output files individually or all at once.

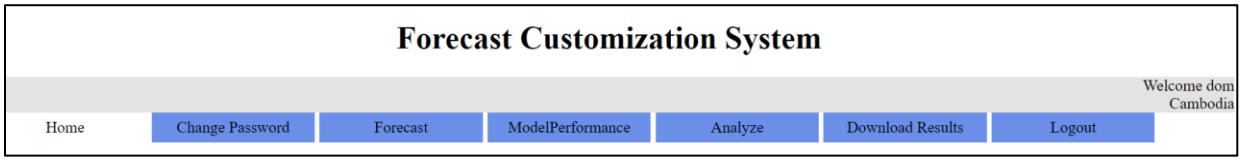


Fig 2: Menu Items

5.1. FORECAST

The forecast menu enables the user to perform preprocessing and statistical downscaling actions, including the data downloading, combining data, data interpolation to the desired grid, or the observed grid. Users must select the **start year**, **end year,** and the **model initialization month** from the drop-down and then check the models and then click on the "**Download**" button. The process would download the data for the selected models from the predefines source. This may take a few minutes based on the network speed.

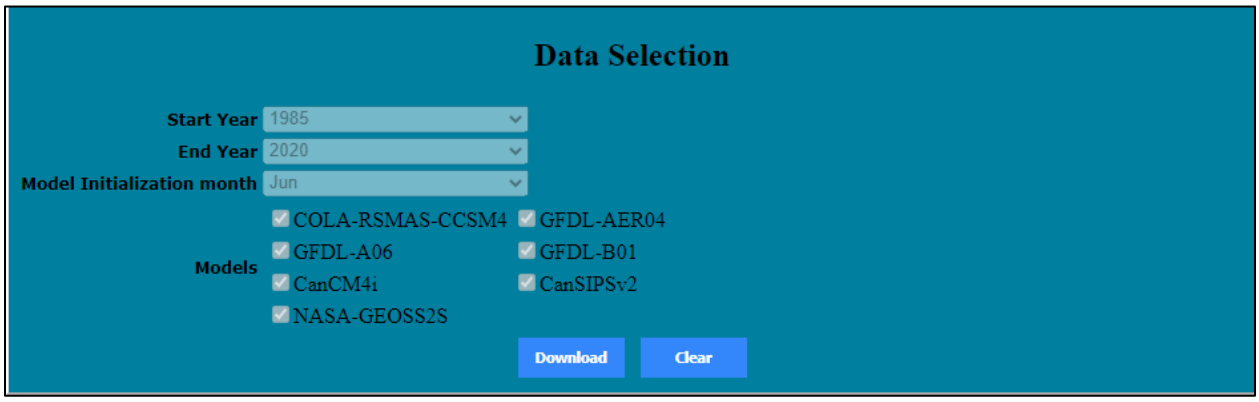


Figure 3: Forecast screen on the web page

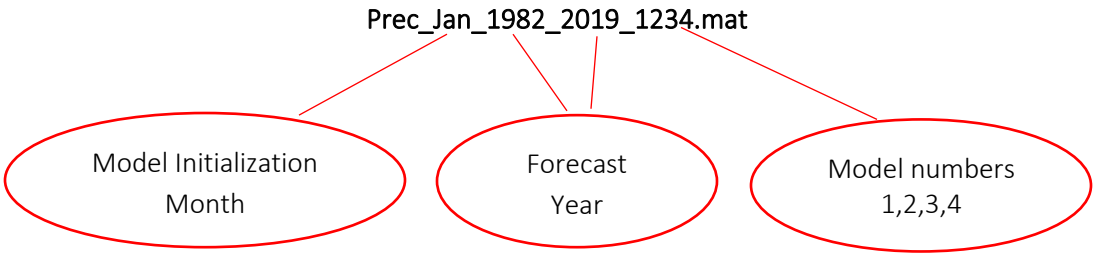
5.1.1. DOWNLOADING MODEL DATA

The focus tool uses seven GCM in the experimental model, including data sources, which are from the IRI data library. So all the model data were retrieved from its respective source, dynamically, every time the download button is clicked.

Note - The model data is automatically downloaded when available to save time for the operational staffs. So, when you click the download button, the process is instant.*

5.1.2. DATA CONVERSION TO MAT FORMAT

This is the process of converting the data models from binary and save it in a single MATLAB file. The python method used to do this operation is "**Prec_Convert_Data.py**". The output file will be stored in the location of $ROOTDIR\RESULTS\(JAN,FEB,.....DEC). The output file will be in the below format.



Once the data is successfully combined, a new window will be opened.

5.1.3. Combining Model Data (forecast and hindcast)

Combine command is for combining model data (forecast with the hindcast data) – is also done in the background. In the background Python method used to perform this procedure is **COMBINE_BIN.py**. The output file will be stored in the same location as that of the input file.

The program combines the Hindcast data for the forecast month (1982-2018) model data to the forecast data (2019). For example, $May_{hindcast\_1982\text{-}2018} + May_{Forecast\_2019}$

5.1.4. Upload Observation Data

Observed monthly data of the provinces, which is in excel format, need to be uploaded by the user. The format of the data should be as shown below. The data should be arranged for all the stations in different worksheets. The uploaded data is used for the model calibration and is averaged over boundaries (either administrative or climate zones) or gridded format.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	
2	1981	0	13.5	12.3	301.9	150.7	69	145.3	112	135.2	82.5	212.3	0.5	
3	1982	0	39.3	23.4	56.9	96.5	151.8	126.9	149.1	300	175.7	48.3	0	
4	1983	0.3	0	53.7	0	111.6	131.7	167.8	191.3	186.3	297.1	150.6	4.6	
5	1984	0.5	0	55.1	22.1	171.5	150.8	242.4	147.8	203.7	149.3	15.2	0.1	
6	1985	43.1	56.8	56.6	137.9	222.4	101.4	212.3	130.7	375.6	213.9	66.4	0	
7	1986	0	12.5	37.4	42.8	87.3	118.8	247.6	227.2	254.9	202	32.3	32.8	
8	1987	0	38.6	14.2	56.9	96.5	176.8	126.9	149.1	301	176.1	48.3	0	
9	1988	0.3	0	46.8	0.3	112.2	125.7	168.2	191.3	183.3	395.7	201	4.6	

Figure 4: Window to upload observation data and interpolate model data

The output will be two files in MATLAB format as listed below are stored in the location

$ROOTDIR\RESULTS\OBS\COUNTRY_ABRIVIATION

- **CMB_PROVINCE_OBS_(JAN,FEB,…DEC)_1982_2015.mat**
- **CMB_PROVINCE_OBS_(JFM,FMA,…DJF)_1982_2015.mat**

If the model initialization month is JAN, the OBS data selected for the monthly prediction will be CMB_PROVINCE_OBS_FEB_1982_2015.

5.1.5. Data Interpolation

This procedure is to cut the required domain from the global data and interpolate all the multiple model data to a common resolution. Users have to click on the interpolate button to perform this operation. The python method used for this operation is "**Domain_Cut_Interpolation.py**" at the back end of the application.

The file '**Prec_Jan_1982_2019_1234.mat**' is taken as the input file to do the interpolation, and the output file will be generated and saved in the same location as that of the input file. The format of the output file will be

"**CMB_MOD_PREC_DATA_STRUCT_PROVINCE_JAN_1982_2019_1234.mat**"

After interpolation and Observed data upload, users need to click on the "Next" button to continue. The current window will be hidden and a new window will be opened.

5.1.6. RUN MME MODELS

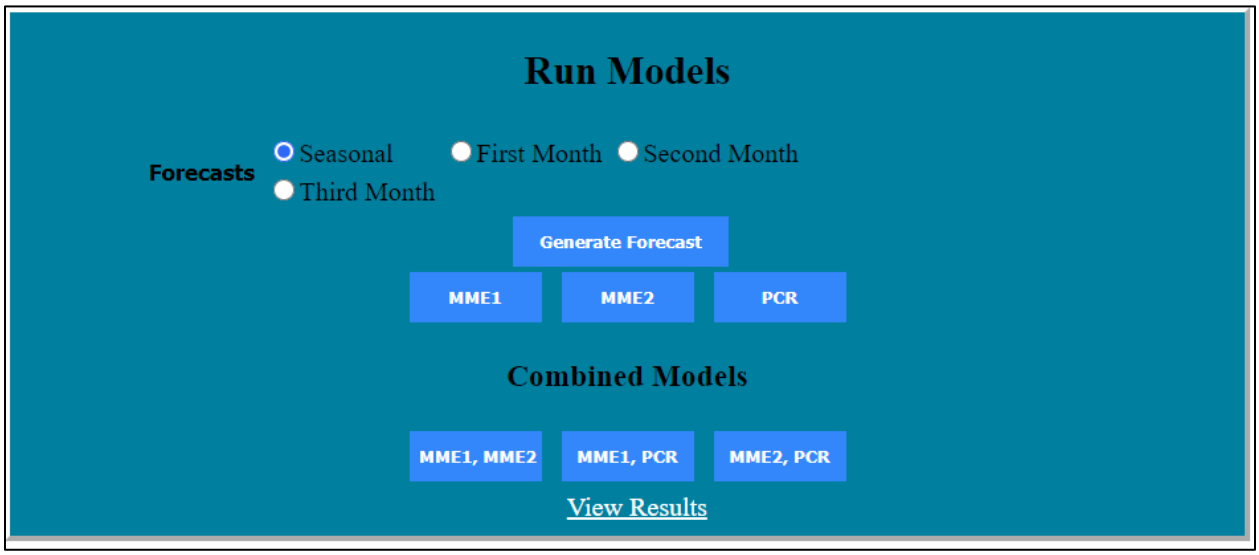


Figure 5: Forecast generation window

Once the preprocessing is over the model run stage is ready to be run. Users can click on each option to generate the forecast individually for each MME scheme or can click on "**Generate Forecast**" to generate all MMEs at once. Also, select between **monthly** or **seasonal** to generate the forecast for the corresponding month or the season (3 months). Various techniques used to generate the monthly or seasonal forecast, which is listed below and will see the techniques in detail. All processed data and results are stored in a specific directory structure in the backend of the server and are provided access to the user through the interface. Users will not have access to the data from the intermediate steps, rather can access the final products only.

5.1.7. MME1

The python method used for this operation is "**MME1.py**". To generate the monthly MME1 forecast for February 2019, the preprocessed combined and interpolated mat data file is used.

CMB_MOD_PREC_DATA_STRUCT_PROVINCE_JAN_1982_2019_1234.mat and
CMB PROVINCE OBS FEB 1982 2015.mat

And to generate the seasonal MME1 forecast for FMA(Feb,Mar,Apr) 2019, for example the input file will be:

CMB_MOD_PREC_DATA_STRUCT_PROVINCE_JAN_1982_2019_1234.mat and
CMB PROVINCE OBS FMA 1982 2015.mat

The output file will be stored in the location $ROOTDIR\RESULTS\CMB\(JAN,FEB,..DEC)\FCST_2019 which contains result files and plots of Climatology, Anomaly, Correlation, and percentage Departure and are listed below.

- MME1_ANOM_MAR_APR_2019_1234.mat
- MME1_ANOM_SERIES_PROVINCE_MAR_APR_2019_1234.mat
- MME1_CMB_MAR_APR_2019_1234.mat
- OBS_CLIM_CMB_MAR_APR_1982_2015.mat
- MME1_ANOM_MAR_APR_2019_1234.PNG
- MME1_DEP_MAR_APR_2019_1234.PNG
- MME1_CORR_MAR_APR_2019_1234.PNG
- OBS_CLIM_MAR_APR_1982_2015.PNG

5.1.8. MME2

To generate the monthly MME2 forecast, the python subroutine used for this operation is "**MME2.py**". For example, to generate MME2 based forecast for February 2019, the input file will be:

CMB_MOD_PREC_DATA_STRUCT_PROVINCE_JAN_1982_2019_1234.mat and
CMB PROVINCE OBS FEB 1982 2015.mat

And to generate the Seasonal MME2 forecast for FMA(Feb,Mar,Apr) 2019, the input file will be

CMB_MOD_PREC_DATA_STRUCT_PROVINCE_JAN_1982_2019_1234.mat and
CMB PROVINCE OBS FMA 1982 2015.mat

The output file will be stored in the location $ROOTDIR\RESULTS\CMB\(JAN,FEB,..DEC)\FCST_2019 which contains result files and plots of Anomaly, Correlation, and percentage Departure and are listed below.

- MME2_ANOM_MAR_APR_2019_1234.mat
- MME2_ANOM_SERIES_PROVINCE_MAR_APR_2019_1234.mat
- MME2_CMB_MAR_APR_2019_1234.mat
- MME2_ANOM_MAR_APR_2019_1234.PNG
- MME2_DEP_MAR_APR_2019_1234.PNG
- MME2_CORR_MAR_APR_2019_1234.PNG

5.1.9. PCR

The python method used for this operation is "**PCR.py**". To generate the PCR based monthly forecast for February 2019, the input file will be:

CMB_MOD_PREC_DATA_STRUCT_PROVINCE_JAN_1982_2019_1234.mat and
CMB PROVINCE OBS FEB 1982 2015.mat

And to generate the Seasonal PCR forecast for FMA (Feb,Mar,Apr) 2019, the input file will be:

CMB_MOD_PREC_DATA_STRUCT_PROVINCE_JAN_1982_2019_1234.mat and
CMB PROVINCE OBS FMA 1982 2015.mat

The output file will be stored in the location $ROOTDIR\RESULTS\CMB\(JAN,FEB,..DEC)\FCST_2019 which contains result files and plots of Anomaly, Correlation, and percentage Departure and are listed below.

- PCR_ANOM_MAR_APR_2019_1234.mat
- PCR_ANOM_SERIES_PROVINCE_MAR_APR_2019_1234.mat
- PCR_CMB_MAR_APR_2019_1234.mat
- PCR_ANOM_MAR_APR_2019_1234.PNG
- PCR_DEP_MAR_APR_2019_1234.PNG
- PCR_CORR_MAR_APR_2019_1234.PNG

All file nomenclature is based on the following format:

countryCODE_method_month/season_year_ModelNumber.file_extn

5.1.10. COMBINED FORECAST (OPTIONAL)

The python method used for this operation is "**COMBINED_FCST.py**". The combined forecast can be MME1-MME2 or MME1-PCR or MME2-PCR.

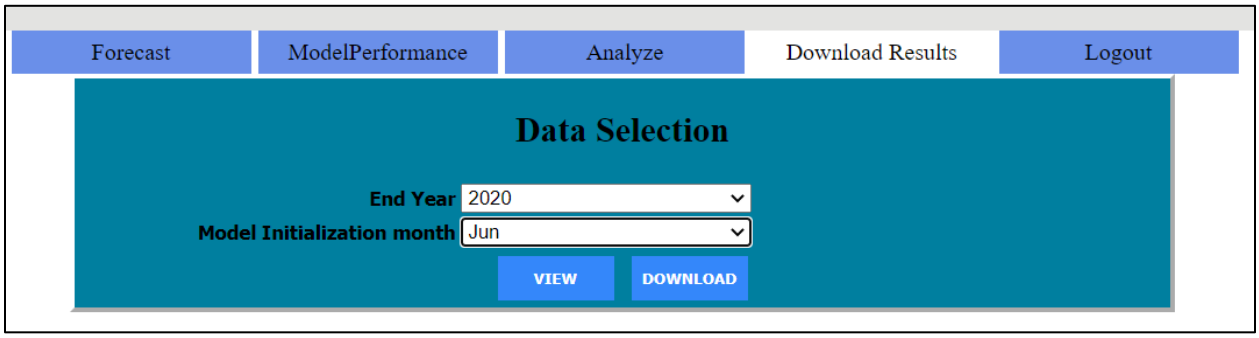
To generate the combined monthly forecast of MME1 and MME2, the input files required are observed climatology and the results of SMM and MME2 forecast. Thus the user has to perform those first. The output of the forecast is the average of the forecast we combine.

The output file will be stored in the location $ROOTDIR\RESULTS\CMB\(JAN, FEB,..DEC)\FCST_2019, which contains result files and plots of Anomaly, percentage Departure, and probabilistic forecast. The combined forecast for MME1 and MME2 are listed below.

- COMB_STD_ANOM_MME1_MME2_CMB_MAR_APR_2019_1234.mat
- COMB_MME1_MME2_CMB_MAR_APR_2019_1234.mat
- COMB_MME1_MME2_ANOM_MAR_APR_2019_1234.mat
- COMB_MME1_MME2_ANOM_MAR_APR_2019_1234.PNG
- COMB_MME1_MME2_DEP_MAR_APR_2019_1234.PNG
- PROB_FCST_MME1_MME2_MAR_APR_2019_1234.PNG

5.1.11. Download Results

Users can move to the Download tab and select the month and year for which they intend to download the forecast. Once clicked on **View**, the model tool will show all the model results for that month.



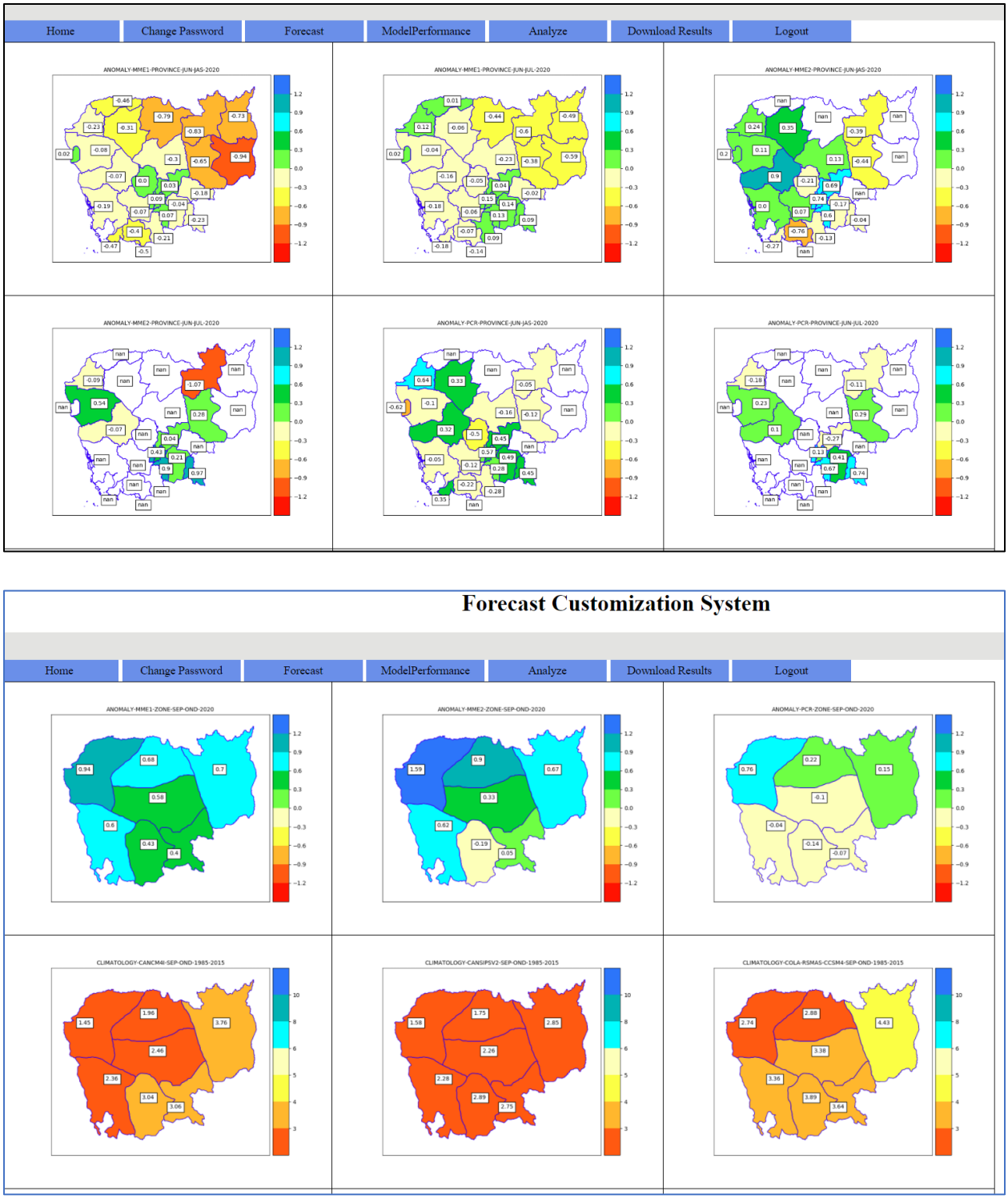
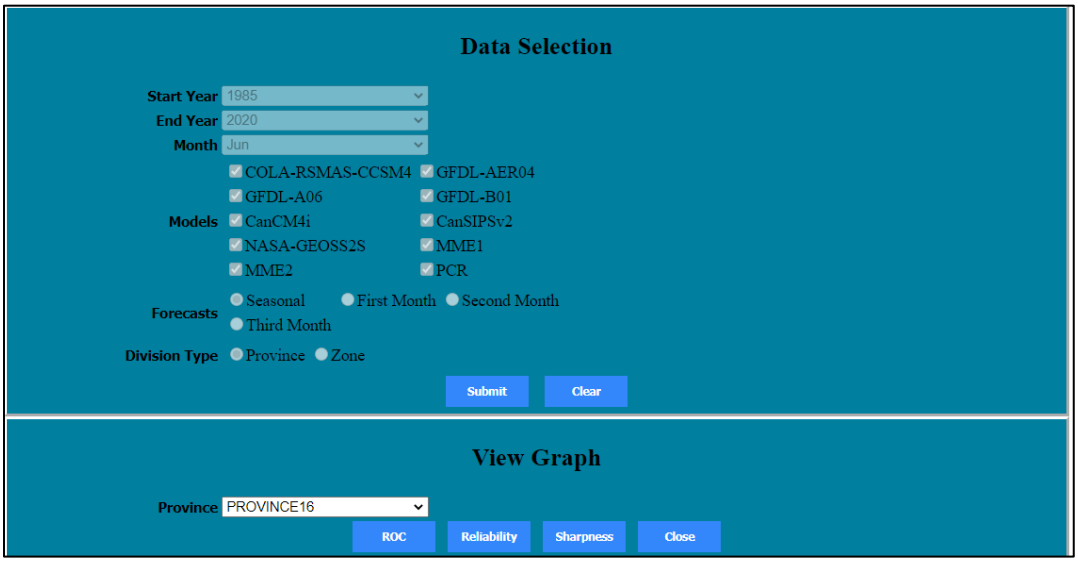


Figure 6. Model results availabe both for Provincial level as well as Climate zone wise

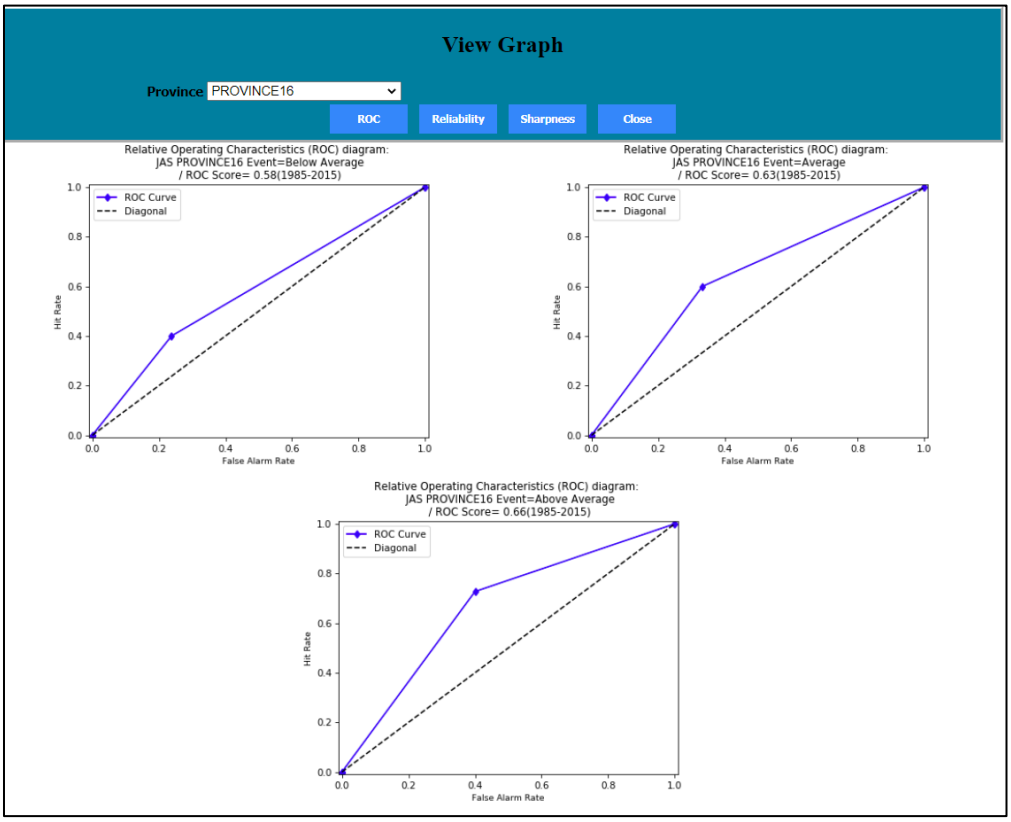
6. VERIFICATION USING ROC AND RELIABILITY

ROC and Reliability are for evaluating the model performance of the country for the respective month and the season. Open the tab **ROC and Reliability**



Select all models with the MMEs and select monthly or seasonal and then click submit.

The graph option will appear and will show either to do for the country as a whole or for regions/zones. For all three tercile categories. Like these.



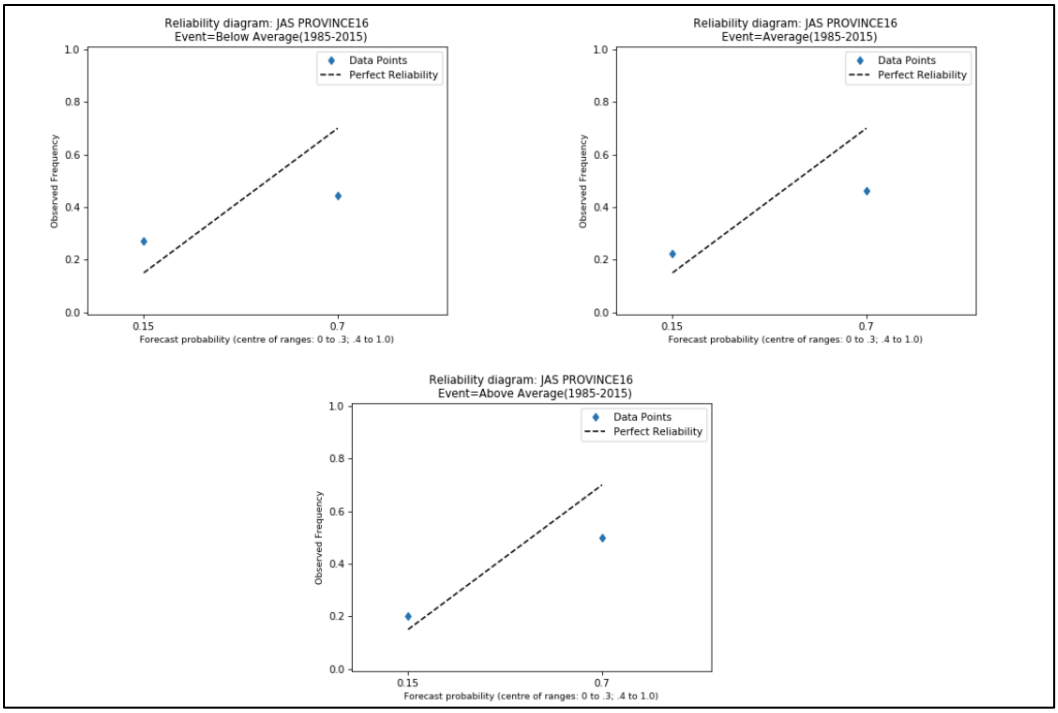


Figure 7: ROC and Reliability Diagram for the province in Cambodia

Annex 1

Model details

1.1. North American Multi-Model Ensemble (NMME)

The NMME database is the **most comprehensive seasonal prediction dataset available to the public**. Model data downloaded from IRI data library includes COLA-RSMAS-CCSM3, GFDL-CM2p1-aer04, GFDL-CM2p5-FLOR-A06, GFDL-CM2p5-FLOR-B01.

COLA-RSMAS-CCSM3

The RSMAS-COLA forecasts are made with the NCAR CCSM3.0 (Collins et al. 2006) and described in detail in Kirtman and Min (2009). Collins, W. D., and co-authors, 2006a: The Community Climate System Model version 3 (CCSM3). J. Climate, 19, 2122-2143. Kirtman, B. P., and D. Min, 2009: Multi-model ensemble ENSO prediction with CCSM and CFS. The retrospective forecast covers the period 1982-2019, with 0-7 months as lead time. So far, 6 ensemble member forecasts are available; the initial conditions are obtained from large ensemble simulations of each corresponding member.

Description

Ensemble Members 6, Resolution 1° x 1° from 0E to 359E and 90S to 90N (360 x 181 Longitude/Latitude), Anomaly-Coupled, Total Precipitation

Web Address 1

http://iridl.ldeo.columbia.edu/SOURCES/.Models/.NMME/.COLA-RSMAS CCSM3/.MONTHLY/.prec/S/(0000%201%20Jan%202018-2018)/VALUES/L/(0.5)/(6.5)/RANGEEDGES/

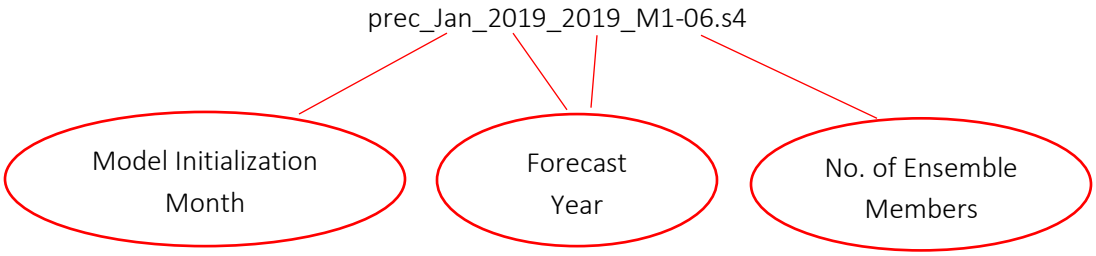
Highlighted part of the web address given above changes according to the start year, end year, and model initialization month.

Formats

Interpolated data over Cambodia domain in ASCII format and global data in binary format. Unit for data set: mm/day.

Download Procedure

Binary Fortran Sequential Access Data is downloaded automatically from the web address and saved in the $ROOTDIR\DATA\COLA_RSMAS_CCSM3\(Jan, Feb,.....Dec) directory. The ROOTDIR exists in the server. Data selection is based on the current available data. If hindcast data from 1982 to 2018 is available, then forecast data for 2019 only need to be downloaded.



Combine the downloaded the data with the hind cast data for further processing.

ie, prec_Jan_1982_2018_M1-06.s4 and prec_Jan_2019_2019_M1-06.s4 is combined to get prec_Jan_1982_2019_M1-06.s4

GFDL-CM2p1-aer04

The GFDL-CM2p1-aer04 data in the IRI/LDEO collection of climate data. The retrospective forecast covers the period 1982-2019, with 0-7 months as lead time. So far, 10 ensemble member forecasts are available; the initial conditions are obtained from large ensemble simulations of each corresponding member.

Description

Ensemble Members 10, Resolution 1° x 1° from 0E to 359E and 90S to 90N (360 x 181 Longitude/Latitude), Fully-Coupled, Total Precipitation

Web Address 2

http://iridl.ldeo.columbia.edu/SOURCES/.Models/.NMME/.GFDL-CM2p1-aer04/.MONTHLY/.prec/S/%280000%201%20Jan%202019-2019%29VALUES/L/%280.5%29%286.5%29RANGEEDGES/

Formats

Interpolated data over Cambodia domain in ASCII format and global data in binary format. Unit for data set: mm/day and stored in $ROOTDIR\DATA\ GFDL_CM2p1_aer04\(Jan, Feb,.....Dec)

NOTE: Download Date: 10-15th of every month. Data downloading procedure is same as that illustrated for the model COLA-RSMAS-CCSM3.

GFDL-CM2p5-FLOR-A06

The GFDL-CM2p5-FLOR-A06 data in the IRI/LDEO collection of climate data. The retrospective forecast covers the period 1982-2019, with 0-7 months as lead time. So far, 12 ensemble member forecasts are available; the initial conditions are obtained from large ensemble simulations of each corresponding member.

Description

Ensemble Members 12, Resolution 1° x 1° from 0E to 359E and 90S to 90N (360 x 181 Longitude/Latitude), Fully-Coupled, Total Precipitation

Web Address

http://iridl.ldeo.columbia.edu/SOURCES/.Models/.NMME/.GFDL-CM2p5-FLOR-A06/.MONTHLY/.prec/S/%280000%201%20Jan%202019-2019%29VALUES/L/%280.5%29%286.5%29RANGEEDGES/

Formats

Interpolated data over Cambodia domain in ASCII format and global data in binary format.
Unit for data set: mm/day and stored in
$ROOTDIR\DATA\GFDL_CM2p5_FLOR_A06\(Jan,Feb,.....Dec).

NOTE: Download Date: 15-25th of every month. Data downloading procedure is same as that illustrated for the model COLA-RSMAS-CCSM3.

GFDL-CM2p5-FLOR-B01

The GFDL-CM2p5-FLOR-B01 data in the IRI/LDEO collection of climate data. The retrospective forecast covers the period 1982-2019, with 0-7 months as lead time. So far, 12 ensemble member forecasts are available; the initial conditions are obtained from large ensemble simulations of each corresponding member.

Description

Ensemble Members 10, Resolution 1° x 1° from 0E to 359E and 90S to 90N (360 x 181 Longitude/Latitude), Fully-Coupled, Total Precipitation

Web Address

http://iridl.ldeo.columbia.edu/SOURCES/.Models/.NMME/.GFDL-CM2p5-FLOR-B01/.MONTHLY/.prec/S/%280000%201%20Jan%202019-2019%29VALUES/L/%280.5%29%286.5%29RANGEEDGES/

Formats

Interpolated data over Cambodia domain in ASCII format and global data in binary format.

Unit for data set: mm/day and stored in

$ROOTDIR\DATA\GFDL_CM2p5_FLOR_B01\(Jan,Feb,.....Dec).

NOTE: Download Date: 15-25th of every month. Data downloading procedure is same as that illustrated for the model COLA-RSMAS-CCSM3.

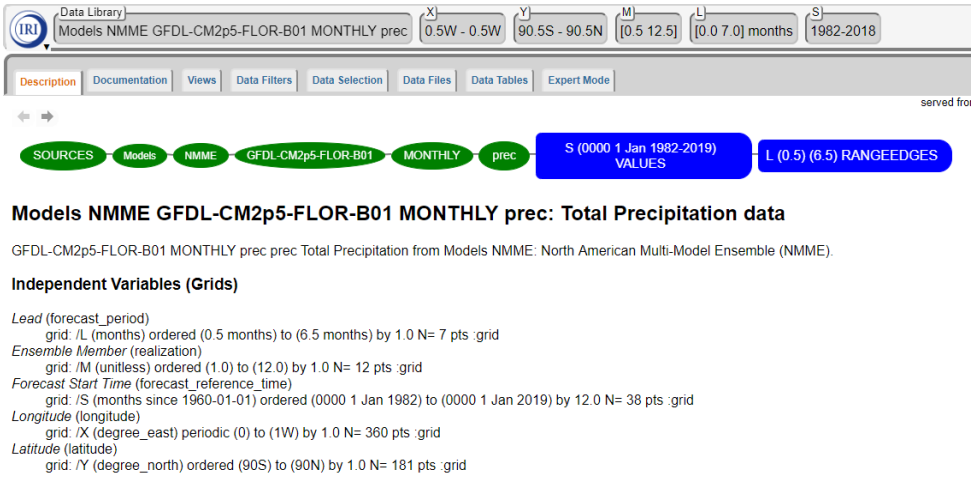


Figure 2: Sample Data Selection

Annex 2

Information about GCM Data Repository

Detail information about the repository is listed

1. **COLA-RSMAS-CCSM4**
    1. Web Address : http://iridl.ldeo.columbia.edu/SOURCES/.Models/.NMME/.COLA-RSMAS-CCSM4/.MONTHLY/
    2. Directories : "PREC","TEMP","SST"
    3. Parameters : "prec","tref","sst"
    4. Leads : 0.5-8.5
    5. Ensemble Members : 1-10
    6. Latitudes : Y-90S to 90N-181
    7. Longitudes : X-0 to 1W-361
    8. Units
        a) Total Precipitation : mm/day
        b) Reference Temperature : Kelvin
        c) Sea Surface Temperature : Kelvin
    9. Time Period : Jan-1982 to Current

2. **GFDL-CM2p1-aer04**
    1. Web Address : http://iridl.ldeo.columbia.edu/SOURCES/.Models/.NMME/.GFDL-CM2p1-aer04/.MONTHLY/
    2. Directories : "PREC","TEMP","SST"
    3. Parameters : "prec","tref","sst"
    4. Leads : 0.5-8.5
    5. Ensemble Members : 1-10
    6. Latitudes : Y-90S to 90N-181
    7. Longitudes : X-0 to 1W-361
    8. Units
        a) Total Precipitation : mm/day
        b) Reference Temperature : Kelvin
        c) Sea Surface Temperature : Kelvin
    9. Time Period : Jan-1982 to Current

3. **GFDL-CM2p5-FLOR-A06**

1. Web Address : http://iridl.ldeo.columbia.edu/SOURCES/.Models/.NMME/.GFDL-CM2p5-FLOR-A06/.MONTHLY/
2. Directories : "PREC","TEMP","SST"
3. Parameters : "prec","tref","sst"
4. Leads : 0.5-8.5
5. Ensemble Members : 1-12
6. Latitudes : Y-90S to 90N-181
7. Longitudes : X-0 to 1W-361
8. Units
    a) Total Precipitation : mm/day
    b) Reference Temperature : Kelvin
    c) Sea Surface Temperature : Kelvin
9. Time Period : Jan-1982 to Current

4. GFDL-CM2p5-FLOR-B01

1. Web Address : http://iridl.ldeo.columbia.edu/SOURCES/.Models/.NMME/.GFDL-CM2p5-FLOR-B01/.MONTHLY/
2. Directories : "PREC","TEMP","SST"
3. Parameters : "prec","tref","sst"
4. Leads : 0.5-8.5
5. Ensemble Members : 1-12
6. Latitudes : Y-90S to 90N-181
7. Longitudes : X-0 to 1W-361
8. Units
    a) Total Precipitation : mm/day
    b) Reference Temperature : Kelvin
    c) Sea Surface Temperature : Kelvin
9. Time Period : Jan-1982 to Current

5. NCEP-CFSv2

- **Hindcast**
    1. Web Address : http://iridl.ldeo.columbia.edu/SOURCES/.Models/.NMME/.NCEP-CFSv2/.HINDCAST/.MONTHLY/
    2. Directories : "PREC","TEMP","SST"
    3. Parameters : "prec","tref","sst"
    4. Leads : 0.5-8.5
    5. Ensemble Members : 1-24

6. Latitudes : Y-90S to 90N-181
7. Longitudes : X-0 to 1W-361
8. Units
    a) Total Precipitation : mm/day
    b) Reference Temperature : Kelvin
    c) Sea Surface Temperature : Kelvin
9. Time Period : Jan-1982 to Dec-2010
10. Missing Data : Jan-2011 to Mar-2011

- **Forecast**

1. Web Address : http://iridl.ldeo.columbia.edu/SOURCES/.Models/.NMME/.NCEP-CFSv2/.FORECAST/.EARLY_MONTH_SAMPLES/.MONTHLY/

2. Directories : "PREC","TEMP","SST"
3. Parameters : "prec","tref","sst"
4. Leads : 0.5-8.5
5. Ensemble Members : 1-24
6. Latitudes : Y-90S to 90N-181
7. Longitudes : X-0 to 1W-361
8. Units
    a) Total Precipitation : mm/day
    b) Reference Temperature : Kelvin
    c) Sea Surface Temperature : Kelvin
9. Time Period : Apr-2011 to Current

6. NASA-GEOS-S2S

- **Hindcast**

1. Web Address : http://iridl.ldeo.columbia.edu/SOURCES/.Models/.NMME/.NASA-GEOSS2S/.HINDCAST/.MONTHLY/
2. Directories : "PREC","TEMP","SST"
3. Parameters : "prec","tref","sst"
4. Leads : 0.5-8.5
5. Ensemble Members : 1-4
6. Latitudes : Y-90S to 90N-181
7. Longitudes : X-0 to 1W-361
8. Units
    a) Total Precipitation : mm/day
    b) Reference Temperature : Kelvin
    c) Sea Surface Temperature : Kelvin
9. Time Period : Jan-1982 to Dec-2016
10. Missing Data : Jan-2017 to Oct-2017

- **Forecast**
  1. Web Address : http://iridl.ldeo.columbia.edu/SOURCES/.Models/.NMME/.NASA-GEOSS2S/.FORECAST/.MONTHLY/
  2. Directories : "PREC","TEMP","SST"
  3. Parameters : "prec","tref","sst"
  4. Leads : 0.5-8.5
  5. Ensemble Members : 1-10
  6. Latitudes : Y-90S to 90N-181
  7. Longitudes : X-0 to 1W-361
  8. Units
     a) Total Precipitation : mm/day
     b) Reference Temperature : Kelvin
     c) Sea Surface Temperature : Kelvin
  9. Time Period : Nov-2017 to Current

7. CanSIPSv2

- **Hindcast**
  1. Web Address : http://iridl.ldeo.columbia.edu/SOURCES/.Models/.NMME/.CanSIPSv2/.HINDCAST/.MONTHLY/
  2. Directories : "PREC","TEMP","SST"
  3. Parameters : "prec","tref","sst"
  4. Leads : 0.5-8.5
  5. Ensemble Members : 1-20
  6. Latitudes : Y-90S to 90N-181
  7. Longitudes : X-0 to 1W-361
  8. Units
     a) Total Precipitation : mm/day
     b) Reference Temperature : Kelvin
     c) Sea Surface Temperature : Kelvin
  9. Time Period : Jan-1982 to Dec-2018
  10. Missing Data : Jan-2019 to Jul-2019
- **Forecast**
  1. Web Address : http://iridl.ldeo.columbia.edu/SOURCES/.Models/.NMME/.CanSIPSv2/.FORECAST/.MONTHLY/
  2. Directories : "PREC","TEMP","SST""
  3. Parameters : "prec","tref","sst"
  4. Leads : 0.5-8.5
  5. Ensemble Members : 1-20
  6. Latitudes : Y-90S to 90N-181

7. Longitudes : X-0 to 1W-361
8. Units
    a) Total Precipitation : mm/day
    b) Reference Temperature : Kelvin
    c) Sea Surface Temperature : Kelvin
9. Time Period : Aug 2019 to Current

8. CanCM4i

- **Hindcast**
    1. Web Address : http://iridl.ldeo.columbia.edu/SOURCES/.Models/.NMME/.CanCM4i/.HINDCAST/.MONTHLY/
    2. Directories : "PREC","TEMP","SST"
    3. Parameters : "prec","tref","sst"
    4. Leads : 0.5-8.5
    5. Ensemble Members : 1-10
    6. Latitudes : Y-90S to 90N-181
    7. Longitudes : X-0 to 1W-361
    8. Units
        a) Total Precipitation : mm/day
        b) Reference Temperature : Kelvin
        c) Sea Surface Temperature : Kelvin
    9. Time Period : Jan-1982 to Dec-2018
    10. Missing Data : Jan-2019 to Jul-2019
- **Forecast**
    1. Web Address : http://iridl.ldeo.columbia.edu/SOURCES/.Models/.NMME/.CanCM4i/.FORECAST/.MONTHLY/
    2. Directories : "PREC","TEMP","SST"
    3. Parameters : "prec","tref","sst"
    4. Leads : 0.5-8.5
    5. Ensemble Members : 1-10
    6. Latitudes : Y-90S to 90N-181
    7. Longitudes : X-0 to 1W-361
    8. Units
        a) Total Precipitation : mm/day
        b) Reference Temperature : Kelvin
        c) Sea Surface Temperature : Kelvin
    9. Time Period : Aug 2019 to Current

9. ECMWF

- **Hindcast**
    1. Web Address : http://iridl.ldeo.columbia.edu/SOURCES/.EU/.Copernicus/.CDS/.C3S/.ECMWF/.SEAS5/.hindcast/

2. Directories : "PREC","TEMP","SST"
3. Parameters : "prcp","t2m","sst"
4. Leads : 0.5-5.5
5. Ensemble Members : 1-25
6. Latitudes : Y-90S to 90N-181
7. Longitudes : X-0 to 1W-361
8. Units
    a) Total Precipitation : m/s
    b) 2-meter Temperature : Kelvin
    c) Sea Surface Temperature : Kelvin
9. Time Period : Jan 1982 to Dec 2016
10. Missing Data : Jan-2017 to Aug-2017

- **Forecast**

1. Web Address : http://iridl.ldeo.columbia.edu/SOURCES/.EU/.Copernicus/.CDS/.C3S/.ECMWF/.SEAS5/.forecast/
2. Directories : "PREC","TEMP","SST"
3. Parameters : "prcp","t2m","sst"
4. Leads : 0.5-5.5
5. Ensemble Members : 1-51
6. Latitudes : Y-90S to 90N-181
7. Longitudes : X-0 to 1W-361
8. Units
    a) Total Precipitation : m/s
    b) 2-meter Temperature : Kelvin
    c) Sea Surface Temperature : Kelvin
9. Time Period : Sep 2017 to Current.

Annex 4

Forecast Prerequisites

Several MATLAB files are required to accomplish the forecast generation. These files are placed in the server location before generating the forecast.

COLA_GFDL_latlon.mat: placed inside the file folder named MODEL. Another folder, ZONES/CMB, will contain the below MATLAB files.

Cambodia_Coordinates.mat: This file contains the height, length, latitude, and longitude boundaries to plot the results. Also, it contains the lat-long coordinates of each province to show the result on the plot.

Cambodia_latlon.mat: The file contains the longitude and latitude values by 0.25 grids

lat: 10, 10.25, 10.50, ……., 15

long: 102, 102.25, 102.50, ……, 108

cambodia_sub_division_provinces.mat: The file contains all the coordinates of each of the provinces which are extracted from the shape file.

Cambodia-Province-Stations.mat: This contains station names in each of the provinces.

cambodia_province(1,2,…,25)_mask: The file is generated by creating a 0.25 grid for longitude and latitude data, then find the area covering each of the provinces and masking the rest of the area.

The root folder for data download, results etc are defined in the database table 'tblConfigurations'.

21x25 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	NaN	NaN	NaN	1	NaN	NaN
3	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1	1	1	NaN	NaN	NaN	1	1	1	1	1	1	NaN	NaN	NaN
4	NaN	NaN	NaN	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NaN	NaN	NaN
5	NaN	NaN	NaN	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NaN	NaN	NaN
6	NaN	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NaN	NaN
7	NaN	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NaN	NaN
8	NaN	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NaN	NaN
9	NaN	NaN	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NaN	NaN
10	NaN	NaN	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NaN	NaN
11	NaN	NaN	NaN	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NaN	NaN
12	NaN	NaN	NaN	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NaN	NaN	NaN	NaN
13	NaN	NaN	NaN	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN
14	NaN	NaN	NaN	1	1	1	1	1	1	1	1	1	1	1	1	1	NaN	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
15	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1	1	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
16	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1	1	1	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
17	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1	1	1	1	NaN	1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
18	NaN	NaN	NaN	NaN	NaN	1	1	1	1	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
19	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	NaN	1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
20	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
21	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Figure 1: Mask file for Cambodia Province

Calling Python Scripts

C# code used to call the python script is given below.

```
string python = ConfigurationManager.AppSettings.Get("PythonExePath").ToString();
string myPythonApp = @"D:\Seasonal\" + pythoncmd;
ProcessStartInfo myProcessStartInfo = new ProcessStartInfo(python);
myProcessStartInfo.UseShellExecute = false;
myProcessStartInfo.RedirectStandardOutput = true;
myProcessStartInfo.RedirectStandardError = true;
myProcessStartInfo.Arguments = myPythonApp;
Process myProcess = new Process();
myProcess.StartInfo = myProcessStartInfo;
myProcess.Start();
string errors = myProcess.StandardError.ReadToEnd();
myProcess.WaitForExit();
myProcess.Close();
if (errors != "")
{
 if (errors.Substring(0, "Traceback".Length) == "Traceback")
   return errors;
 else
   return "";
}
else
  return "";
```

PythonExePath is specified in the app.config as

```
<add key="PythonExePath" value="C://Python34//Python.exe"/>
```

Pythoncmd changes according to the method called.

For eg: "Prec_Convert_Data.py" + " " + DownloadFilePath + " " + ResultFilePath + " " + ModelFilePath + " " + string.Join(",", selmodels) + " " + StartYear + " " + EndYear + " " + Month
is the pythoncmd for data combine.

Model Data Combine

Data Combine is the process of converting model data from binary, calculate the average of all the members and then save it as a single MATLAB file. The model data is renamed as dat1, dat2 etc. with a shape of [ld, yr, lat, lon]

Where ld: lead time in months (7)

yr: Number of years of forecast data(ie, from 1982 to 2019, value is 38).

lat: Latitude range from 90S to 90N(181)

lon: Longitude range from 0 to 359(360)

Process observation Data

Here the python function reads the monthly observation data available in excel format, for each province the data is calculate the mean of observation of all stations in that province. Then creates the MATLAB files for monthly and seasonal categories.

The python code used is given below

```
dt[j, :, k] = worksheet.col_values(colx=k + 1, start_rowx=1, end_rowx=num_yr + 1)
OBS_CN_Zone[i,:,:] = np.nanmean(dt, axis=0);
OBS = np.squeeze(OBS_CN_Zone[:,:, i]); # for monthly
OBS = np.squeeze(OBS_CN_Zone[:, :, i: i + 2]);
OBS = np.squeeze(np.nansum(OBS, axis=2));  # adding  2  months data
```

Data Interpolation

Interpolation technique used here is a linear interpolation. Python code to interpolate COLA data is given below

```
mdl_lat_lon = sio.loadmat(models_path + 'COLA_GFDL_latlon.mat');
xi = mdl_lat_lon["lon_col_gfd"]
yi = mdl_lat_lon["lat_col_gfd"];
dat1 = result["dat1"];
[ld, yr, lat, lon] = dat1.shape;
[x1, y1] = np.meshgrid(xi, yi);  # 181 X 360 grid
[x, y] = np.meshgrid(cn_lon, cn_lat);  # 21 X 25  grid
y = np.flipud(y); # lat top to down
points = np.array((x1.flatten(), y1.flatten())).T
del xi, yi
# Interpolation
[longi, lati] = x.shape;
cola = np.zeros(shape=(ld, yr, longi, lati))
for i in range(0, ld):
   for j in range(0, yr):
      t = dat1[i, j, :, :];
      # t = np.around(t, decimals=4);
      values = t.flatten()
      zi = interp.griddata(points, values, (x, y), method='linear');
      cola[i, j, :, :] = zi;
      del t, values, zi;
```

```
del dat1;
[ld, yr, lt, ln] = cola.shape;
COLA = np.zeros(shape=(ld, yr, int(NoofZones) +1))

for ii in range(0, ld):
   for jj in range(0, yr):
      dex = np.squeeze(cola[ii, jj, :, :]);
      for m in range(0, int(NoofZones)):
         Pts = sio.loadmat(zones_path + Country_Code + "//" + Country_name.lower() + "_" +
TypeName.lower() + str(
            m + 1) + '_mask.mat');
         COLA[ii, jj, m] = np.nanmean(np.reshape((dex * Pts[Country_name + "_" + TypeName + str(m + 1)
+ '_Points']), lt * ln,
                  0));
      Pts2 = sio.loadmat(zones_path + Country_Code + "//" + Country_name.lower() + '_mask.mat');
      COLA[ii, jj, m + 1] = np.nanmean(np.reshape(dex * Pts2[Country_name + '_Points'], lt * ln, 0));
      del dex
fil['COLA'] = COLA;
```

After interpolation, the data is extracted for each of the provinces using the mask file. The final result will be the models with the shape [ld, yr, zn], Where zn: 25 provinces + Cambodia (Total 26).

Annex 5
Probabilistic Forecast

Theory and Methodology

We assume that seasonal variations of a scalar quantity X (here rainfall) can be represented as a sum of potentially predictable signal β and non-predictable variability that will be treated as stochastic noise ε, that is,

$$X = \beta + \varepsilon$$

The term β comprises all potentially predictable signals arising from external and internal sources. The noise term ε represents the unpredictable effects of day-to-day weather. It is assumed that all terms in (1) are stochastic processes with the zero mean, that is, E(β) = E(ε) = 0, where the symbol E denotes expectation or the mean of a random variable, and they have standard deviations σβ, σε respectively.

Forecasting method

In equation (1), X can be represented as a sum of potentially predictable signal β and non-predictable variability that will be treated as stochastic noise ε,

Now, the problem is to find potentially predictable signal β and non-predictable variability that will be treated as stochastic noise ε for construct the distribution.

Equation (1) leads one more relation

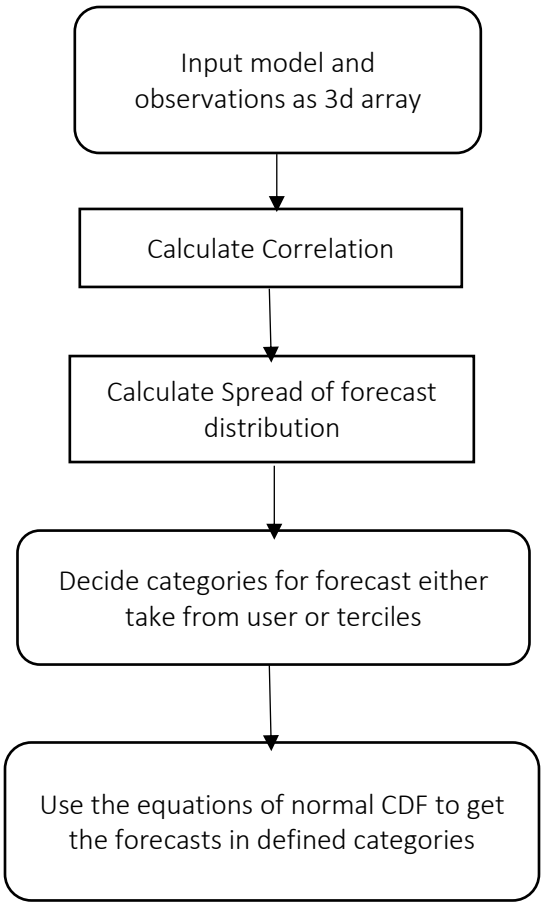
$$\sigma_x^2 = \sigma_\beta^2 + \sigma_\varepsilon^2 \quad \text{as } Cov(\beta, \varepsilon) = 0$$

$$\sigma_x = \sqrt{\sigma_\beta^2 + \sigma_\varepsilon^2}$$

$$\sigma_x = \sqrt{1 + \sigma_\varepsilon^2} \quad if\ \sigma_\beta = 1$$

So, β is the mean of the distribution, and σx is spread of the distribution, and it is a function of σε. There are various methods to calculate σε Ensemble spread method, error residual method, and we use the correlation method to calculate probabilistic MME.

The below flow chart shows how the forecast is calculated.



Python Program

The following python function gave a probabilistic prediction for the current year using the Model predictions and observed data for past years. The correlation between hindcasts and observations is used in this code to get the spread of forecast distribution. The required subprogram is correlation2d1m.py which is also given below.

```
def ProbMME_Tippetetal_all(MMMa, OBS, ind, varargin):
    # MMMa --- A 3D array of Multimodel ensembles
    # OBS  --- A 3D array of observations
    # the sequence is (nx,ny,years) (nx and ny) nlong nlat
    # nx and ny should be same for MMMa and OBS
    # ind --- Vector containing indices to calculate the correlation in hind cast
    # smooth_ind matrix used for smoothening the correlation matrix
    # r may be empty matrix
    # Pa Pb Pn are the Probailistic forecast
    c=correlation_2dlm.correlation2d1m(MMMa[:,:, ind],OBS[:,:,ind]);
    r=c;
    sigerr = np.divide(1,np.sqrt(np.divide(np.square(r),np.subtract(1,
np.square(r)))));
    sigy=np.sqrt(np.add(np.square(sigerr),1));
    n=MMMa.shape[2];
    M=np.nanmean(MMMa[:,:,ind],2);
    stdM=np.nanstd(MMMa[:,:,ind],axis=2, ddof=1);
    if(len(varargin.keys())==0):
```

```
        b13=np.multiply(invgauss.norm.ppf(1/3),sigy);
        b23=np.multiply(invgauss.norm.ppf(2/3),sigy);
        Pb = invgauss.norm.cdf(b13, np.divide((MMMa[:,:,n-1]-M),stdM),sigerr);
        Pa = 1-invgauss.norm.cdf(b23, np.divide((MMMa[:,:,n-1]-M),stdM),sigerr);
        Pn = 1-Pa-Pb;
        varargout = np.empty(shape=(3, OBS.shape[1]));
        varargout[0,:]=Pa;
        varargout[1,:]=Pb;
        varargout[2,:]=Pn;
    else:
        Area_norminv=varargin[0];
        sz=Area_norminv.shape;
        nl=sz[0];
        aera_n=Area_norminv[0,:];
        b13=np.multiply(invgauss.norm.ppf(aera_n),sigy);
        Pb1 = invgauss.norm.cdf(b13,np.divide((MMMa[:,:,n-1]-M), stdM),sigerr);
        varargout = {};
        varargout[0]=Pb1;
        for i in range(1,nl):
            aera_n=Area_norminv[i,:];
            b13=np.multiply(invgauss.norm.ppf(aera_n),sigy);
            Pb =  invgauss.norm.cdf(b13,np.divide((MMMa[:,:,n-1]-M),stdM),sigerr);
            varargout[i]=Pb-Pb1;
            Pb1=Pb;
        b13= np.multiply(invgauss.norm.ppf(1),sigy);
        Pb = invgauss.norm.cdf(b13, np.divide((MMMa[:,:,n-1]-M) ,stdM),sigerr);
        varargout[nl]=Pb-Pb1;
    return varargout;
##########################################################
def correlation2d1m(x,y):
    dim_n=np.ndim(x);
    if (dim_n==4):
        [n1,n2,nt,im]=x.shape;
    if (dim_n == 3):
        [n1,n2,nt]=x.shape;
        im=1;
    xm = np.nanmean(x, 2);
    ym = np.nanmean(y, 2);
    if (dim_n == 4):
        xm = np.repeat(xm[:, :, np.newaxis, :], nt, 2);
    if (dim_n == 3):
        xm = np.repeat(xm[:, :, np.newaxis], nt, 2);
    ym = np.repeat(ym[:, :, np.newaxis], nt, 2);
    xa = x - xm;
    ya = y - ym;
    yv = np.nanmean(np.square(ya), 2);
    xv = np.nanmean(np.square(xa), 2);
    yok = (abs(yv) > 0)*1;
    xok = (abs(xv) > 0)*1;
    ok = xok & yok;
    top = np.nanmean(np.multiply(xa, ya), 2);
    bottom = np.sqrt(np.multiply(xv, yv));
    c = np.zeros(shape=ok.shape);
    ok=(ok==1);
    c[ok] = np.divide(top[ok],bottom[ok]);
    c[~ok] = np.nan;
    c = np.squeeze(c);
    return c;
##########################################################
```