Project Draft / CS565
Filling gaps of the GIW area time-series using Artificial Intelligence methods

Junehyeong Park

I. Introduction

The GIWs (Geographically Isolated Wetlands), one of the types of wetlands, were defined by landscape position as “wetlands with no apparent surface water connection to perennial rivers and streams, estuaries, or the ocean” [1]. These GIWs were surrounded by dry land, and the wetlands along intermittent streams connected to perennial streams were designated as non-isolated. To understand the interactions between wetlands and climate, surface later flows, and groundwater, neglecting large interventions of tidal effects or inundation from nearby perennial streams, GIWs are currently being surveyed.

There are no time-series data of the area or water amounts of wetlands not only officially published but also in research articles. Therefore, to analyze using temporal information of wetland, researchers should get in-situ information in their own. Or, only stationary analysis could be done. In this regard, my current research has focused on generating time-series maps of GIW by GIS works on the time-series of satellite observation [2] and the GIW map [3]. Now, I have generated time-series of GIW area for each sub-region in GA.

The main objective of my research is to represent climate controls on GIWs seasonally and spatially. It would propagate to the analysis for evaluating seasonal reactions, resiliency against drought events, spatial differences of sensitivity, and finally finding causations of disappeared wetlands. However, there is a problem that the generated time-series of GIW area has too much blanks because of the characteristics of satellite observation. To conduct researches mentioned above, the quality control procedure is needed. According to that, in this project, I will build some models to filling gaps with dependent variable as GIW area time-series and explanatory variables as climate records [4].

The dependent variable, artificial GIW are time-series, will be divided into available timesteps to use and blanks. During the available timesteps, GIW area and other variables will be used to fit the models. Some models, including ANN and RNN, will be adopted to compare the performance of simulation and prediction. Because there are no true values for GIW area, it is hard to find the answer model, but I need to compare models specifically for selecting the most appropriate (or the least error occurred) model.

II. Data

A. Research area

In this research, prairie potholes in North Dakota and coastal plain wetlands in Florida were selected as representative states of Northward and Southward of U.S.. The regions were spatially divided by the PRISM grids to use climate information of PRISM forcing data. Even though this report does not include any results using forcing information yet, this work is regarded as a primary research that will use climate information to analyze what will cause the changes of wetlands, so that the target regions are divided in PRISM grids. Each region includes 64 PRISM grids and 63 grids (around 1,000 km^2) that have less observation failures during 380 months (Mar 1984 - Oct 2015) compared to other grids, respectively. In the Fig. 1, selected regions are shown. The number of available observation during 380 months was represented as blue color, and the regions were selected considering NWI also.

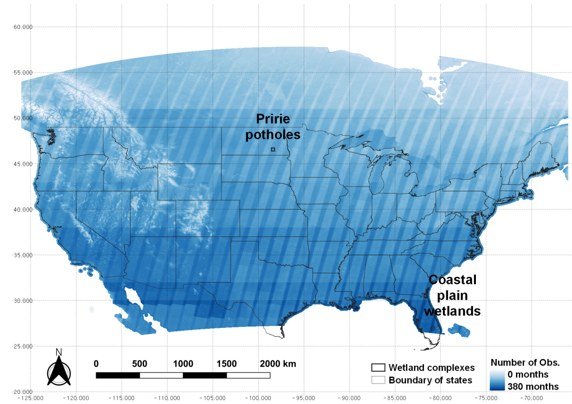

Fig. 1. Selected regions for prairie potholes and coastal plain wetlands

B. Global Surface Water

Global Surface Water (GSW) is a Landsat-based globally spatial information about the location and persistence of surface water (inland and coastal) developed by [2]. The GSW includes various types of data such as global surface water mapping layers, monthly water recurrence, monthly water history, and yearly water classification history. The monthly water history data is a basic result that includes raster pixels with 0 (no data), 1 (no water), and 2 (water) integers with 30 m resolution for the globe. Because it is a long-term (Mar 1984 Oct 2015), high-resolution satellite data, the basic result itself has some limitations such as observation failures due to could or others, or lack of information during winter even though there is no snow or ice happened. Other types include additional results of analysis based on the basic result.

Because of the uniqueness that has serial and spatial distribution of surface water presence in 30 m resolution, The global surface water data was used for many researches including wetland researches. For example, Krapu [5] used the global surface water for identifying wetlands consolidation in the North Dakota Prairie Pothole Region. In this research, the monthly water history data of the global surface water imported from the Google Earth Engine was used as a clue for defining there was a surface water body and an input data that was inserted into the algorithm matching surface water body with GIW polygons.

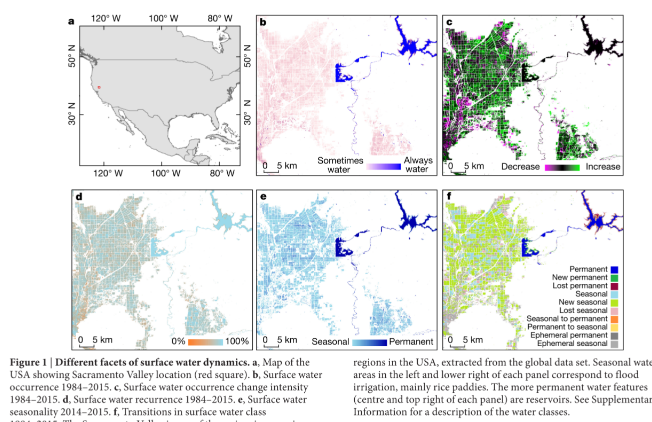

Fig. 2. An example figure of the GSW imported from [2]

C. National Wetlands Inventory and Geographically Isolated Wetlands

National Wetlands Inventory (NWI) [6] is a spatial data presented by the U.S. Fish and Wildlife Service that includes spatial distribution of all kinds of wetlands in the format such as marine, estuarine, riverine, lacustrine, and palustrine systems. The NWI data is regarded as a standard in wetland researches which sets the target region in CONUS.

GIW was delineated in several ways using various logics, tools and date, but Lane [3] showed the most recent and representative way to delineate GIW features from the NWI spatial data. In this research, the GIWs were derived by the algorithm suggested by Lane [3].

In this regard, this research also considered the NWI data and the GIW as the standard definition and delineations of wetlands and GIWs. In Fig. 3, NWI (green) and GIW (blue) features are plotted in the same region to look at in a comparative view. Among these green features, rivers and lakes are also included. However, the blue features, which represents the GIWs, do not include rivers or lakes and only includes GIWs that are apart from such large surface water bodies.

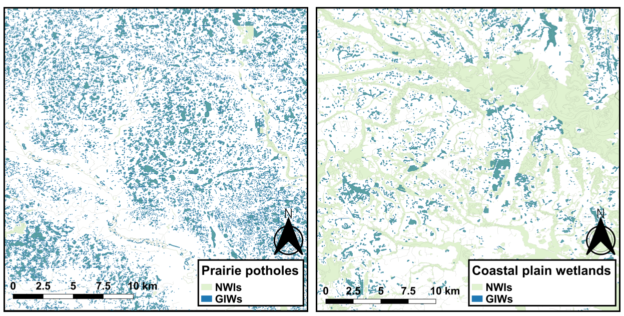

Fig. 3. NWI waterbodies and derived GIWs

D. Methodology to delineate GIWs from GSW maps

With hopeful assumptions, the water pixels in GSW raster data that have the value 2 can be categorized into GIW if the location of the pixel is on the GIW features. However, it is hard to follow that because of following reasons. First, if we use just pixels that is located on the GIW features, we can't capture the cases that wetlands were enlarging or merging with other nearby GIWs. Second, the spatial resolution of the GSW data is 30 m and GIWs are very small, there can be some unexpected cases to avoid. For example, because of the uncertainty of satellite data, the GSW pixels can represents a different place some meters apart from the accurate place. Thus, in this regard, this research generated following two steps with a time-step: 1) find water pixels (value 2) of GSW map that touch any water bodies of GIW. 2) find all water pixels of the GSW map that is located nearby the selected pixel in the step 1. 3) repeat step 2 until there are no more water pixels. 4) repeat from step 1 to 3 for all pixels in the GSW map.

However, another problem can be occurred when the surface water of a GIW was merged with any large lakes or rivers for a short time during floods and the instant case became the case as the representative observation of the month. Then, the time-series of GSW pixels that regarded as GIWs would have extraordinary values. To solve this problem, another process was inserted earlier than mentioned above. The additional process is that 1) generate a map that represents NWI but not GIW (i.e. lakes and rivers), 2) find any water pixels of the GSW map that touch any water bodies of "NWI - GIW", 3) find all water pixels of the GSW map that is located nearby the selected pixel in the step 2, 4) repeat step 3 until there are no more water pixels, 5) repeat from step 2 to 4 for all pixels in the GSW map, and finally 6) define all water pixels of GSW map detected during step 5 as "no water" to neglect.

The whole process from the raw data to the results can be rephrase as below. Based on GSW map (top left) and the spatial information of NWI and GIW (top right), after applying the algorithm (bottom left),the water pixels regarded as GIWs can be selected (bottom right). In this research, the algorithm above was applied to the wettest month and defined the selected water pixels as 'possible GIWs'. Based on this, the water pixels located in the 'possible GIWs' were counted in each month.

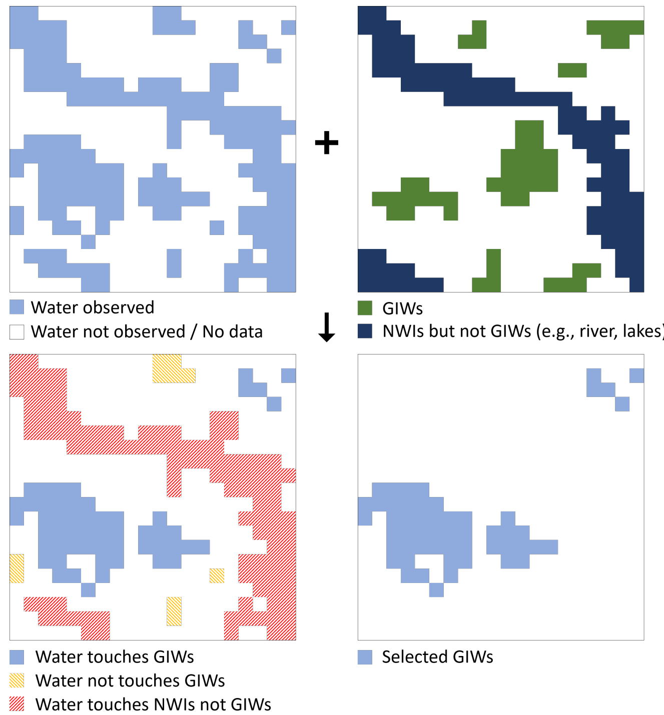

Fig. 4. An artificial sample of detecting water pixels defined as GIWs

III. Method

A. Artificial Neural Networks

Artificial Neural Networks (ANN) is a method that learning relationships using neurons located on the hidden layers. Even though it is a kind of a set of fitted equations like regressions, because of complex structure and meaning less nodes (each nodes cannot have identified physical meanings that input or output variables can have), ANN would be suggested for the simulation or re-generation of data but cannot be used for sensitivity analysis or understanding relationships. In this case, surroundings such as climate variables or observed discharge rates nearby target regions would be used as input variables and the area of GIW would be used as output variable. The months when the area of GIW is not available due to the fraction of observation failure is high or observation is absence, and the months will be used as prediction timesteps.

B. Recurrent Neural Networks

Recurrent Neural Networks (RNN) is an advanced method of ANNs dealing with time-series data. This method uses the current state as output variable, and uses the last state as input variable with other variables. It would be applied when we assume the variable has memory so that auto-correlation is important to simulate the variable.

The area of GIW is one of them that even though there was a storm, the result would be different when the last state was extremely dry or wet enough. Therefore, the RNN, especially LSTM, will be applied to the data to tran the model and predict GIW area at the gaps.

C. Simulations of GIW

In this research, ANN and RNN models were built and compared with the multiple linear regression model. For the multiple linear regression model, precipitation and temperature at same month, and GIW area at previous month were used as 3 predictors. For ANN and RNN, precipitation, temperature, and GIW area at previous month were used as 3 input variables.

The existed GIW area time-series was evaluated by the fraction of 'No Data'. If the fraction of 'No Data' was more than 5 %, the data of the month was eliminated. From this time-series, there would be blanks and the longest data that has longest serial data with 'No Data' fraction smaller than 5 %. The models were built based on this longest data, their performance were recorded, and blanks were filled by the models.

Fitting models, simulations, and generating figures were done by the MATLAB. The code was attached at the Appendix, and relevant files were uploaded in the Github (https://github.com/sai0259/CS565_Project).

IV. Result

A. Performances of methods

The multiple linear regression models were evaluated by R^2, while ANN and RNN were evaluated by relative RMSE. The grid-averaged R^2 of the multiple linear regression was 0.0, and the grid-averaged relative RMSE of ANN and RNN were 0.0 and 0.0, respectively. We can see that two neural network methods showed much better performance to fit and simulate the GIW area based on GIW area in different time and climate conditions.

There are three reasons of poor linear regression performances. First, the number of data points were not enough. The serial data cannot exceed 12 months usually because of the characteristic of GSW data. Due to using satellite data, many regions absolutely don't have images in some months such as November, December, and January when waterbodies become ice. Thus, less than 12 was not enough to generate a good linear regression model. Second, there are many of other predictors that affects on GIW area such as topography, groundwater, and human impacts. Finally, due to compare with RNN, the serial data was applied here. However, serial data less than 1 year occurred a problem of neglecting seasonality and inter-annual variations.

B. Simulated GIWs

Figure 5 and 6 showed filled and existed GIW area in FL and ND by MLR, ANN, and RNN methods. In both cases, MLR generated strange values that is not acceptable, and ANN generated underestimated values and had more variability compared to the RNN. From RNN, more consistent values were generated compared to others. It may occur because RNN depends on the previous value rather than other methods consider relationships with other inputs more. It would be better if the target data has consistent characteristic, but it can occur problems due to GIW area usually varies a lot and shows definite seasonality.

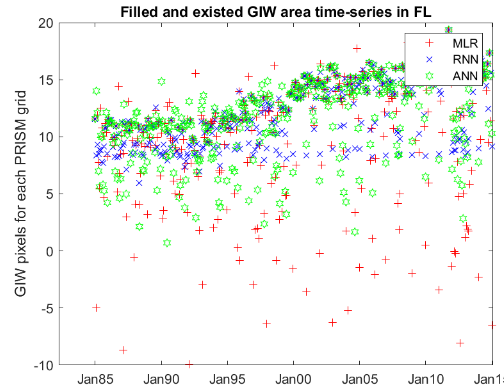

Fig. 5. Filled and simulated GIW area time-series in FL

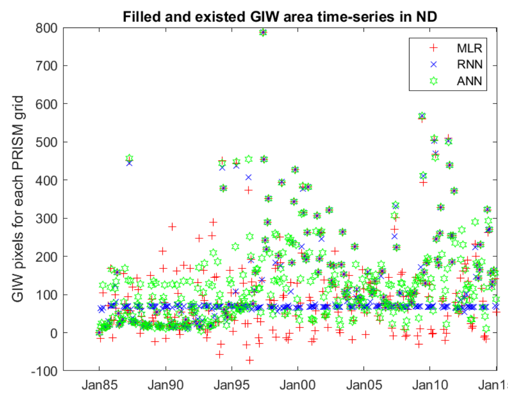

Fig. 6. Filled and simulated GIW area time-series in ND

V. Conclusion

In this research, blanks and low-quality observations in exported GIW time-series from satellite images were filled by MLR, ANN, and RNN methods in FL and ND. The performance metrics of MLR were very poor because of lack of data and ignoring seasonality, and so the filled data also showed low quality. On the other hand, ANN and RNN showed much better quality on filled data compared to MLR. It may imply that neural network can be used in big data, but also have great performance when there is not enough data and background information to generate an effective regressions.

What we can do better for this objective is that fitting linear regression better with data alignment in seasonal information or including additional information such as topography. Due to not enough time for research, such efforts couldn't conducted in this time. However, fitting each model as best condition can show different information.

References

[1] Grey R. Evenson et al. "Geographically isolated wetlands and watershed hydrology: a modified model analysis". In: *Journal of Hydrology* 529.P1 (2015), pp. 240–256. DOI: https://doi.org/10.1016/j.jhydrol.2015.07.039.

[2] Jean-François Pekel et al. "High-resolution mapping of global surface water and its long-term changes". In: *Nature* 540.7633 (2016), pp. 418–422. DOI: https://doi.org/10.1038/nature20584.

[3] Charles R. Lane and Ellen D'Amico. "Identification of Putative Geographically Isolated Wetlands of the Conterminous United States". In: *Journal of the American Water Resources Association* 52.3 (2016), pp. 705–722. DOI: https://doi.org/10.1111/1752-1688.12421.

[4] Christopher Daly, Ronald P. Neilson, and Donald L. Phillips. "A statistical-topographic model for mapping climatological precipitation over mountainous terrain". In: *Journal of Applied Meteorology* 33 (1994), pp. 140–158. DOI: https://doi.org/10.1175/1520-0450(1994)033⟨0140:ASTMFM⟩2.0.CO;2.

[5] Christopher Krapu, Mukesh Kumar, and Mark Borsuk. "Identifying wetland consolidation using remote sensing in the North Dakota Prairie Pothole Region". In: *Water Resources Research* 54.10 (2018), pp. 7478–7494. DOI: https://doi.org/10.1002/2018WRR023338.

[6] U.S. Fish and Wildlife Service. *National Wetlands Inventory*. 2019.

Appendix: Matlab code

```
%%
clear;
clc;
currentdirectory = pwd;

%% Case: FL
num_FL = 63;

fileID = fopen(fullfile(currentdirectory,'\FL\
    ALL_Cohen_G.txt'),'r');
formatSpec = '%d ';
sizeA = [num_FL Inf];
A = fscanf(fileID,formatSpec,sizeA);
fclose(fileID);
FL_All = A';

fileID = fopen(fullfile(currentdirectory,'\FL\
    Water_Cohen_G.txt'),'r');
formatSpec = '%d ';
sizeA = [num_FL Inf];
A = fscanf(fileID,formatSpec,sizeA);
fclose(fileID);
FL_Water = A';
FL_Water_MLR = FL_Water;
FL_Water_rnn = FL_Water;
FL_Water_ann = FL_Water;
FL_Water_raw = FL_Water;

fileID = fopen(fullfile(currentdirectory,'\FL\
    NoData_Cohen_G.txt'),'r');
formatSpec = '%d ';
sizeA = [num_FL Inf];
A = fscanf(fileID,formatSpec,sizeA);
fclose(fileID);
FL_NoData = A';

fileID = fopen(fullfile(currentdirectory,'\FL\
    PPT_Cohen_G.txt'),'r');
formatSpec = '%f ';
sizeA = [num_FL Inf];
A = fscanf(fileID,formatSpec,sizeA);
fclose(fileID);
FL_PPT = A';

fileID = fopen(fullfile(currentdirectory,'\FL\
    TMT_Cohen_G.txt'),'r');
formatSpec = '%f ';
sizeA = [num_FL Inf];
A = fscanf(fileID,formatSpec,sizeA);
fclose(fileID);
FL_TMT = A';

FL_NoDataFrac = FL_NoData./FL_All;
FL_Avail = FL_All(1,:)~=0;
num_FL_new = sum(FL_Avail);

MLR_R2 = zeros(num_FL,1);
RNN_RRMSE = zeros(num_FL,1);
ANN_RRMSE = zeros(num_FL,1);

for iik=1:num_FL
    if FL_Avail(iik)==1
        ii = iik;

        %ii=1;

        temp_NoData = FL_NoDataFrac(:,ii)
            >0.05;
        FL_Water_raw(temp_NoData,1) = NaN;
        aa = zeros(360,1);
        for iii = 2:360
            if FL_NoDataFrac(iii,ii)>0.05
                if FL_NoDataFrac(iii-1,ii)
                    <0.05
                    aa(iii) = 1;
                else
                    aa(iii) = aa(iii-1) + 1;
                end
            end
        end
        maxlen = max(aa);
        [r,c] = find(aa==maxlen);
        temp_st = r(1) - maxlen + 1;
        temp_ed = r(1);

        % Method 1: Multiple Linear Regression

        XX = [FL_Water(temp_st-1:temp_ed-1, ii
            ) FL_PPT(temp_st:temp_ed,ii)
            FL_TMT(temp_st:temp_ed,ii)];
        YY = FL_Water(temp_st:temp_ed, ii);
        lm = fitlm(XX,YY);

        MLR_R2(ii) = lm.Rsquared.Ordinary;

        for mi = 2:360
            if FL_NoDataFrac(mi,ii)>0.05
                YYnew = predict(lm,[FL_Water(
                    mi-1,ii) FL_PPT(mi-1,ii)
                    FL_TMT(mi-1,ii)]);
                FL_Water_MLR(mi,ii) = YYnew;
            end
        end

        % Method 2: RNN

        XTrain = [FL_Water(temp_st:temp_ed-1,
            ii) FL_PPT(temp_st:temp_ed-1,ii)
            FL_TMT(temp_st:temp_ed-1,ii)]';
        YTrain = FL_Water(temp_st+1:temp_ed,
            ii)';

        numFeatures = 3;
        numResponses = 1;
        numHiddenUnits = 200;

        layers = [ ...
            sequenceInputLayer(numFeatures)
            lstmLayer(numHiddenUnits)
            fullyConnectedLayer(numResponses)
            regressionLayer];

        options = trainingOptions('adam', ...
            'MaxEpochs',250, ...
```

```
                'GradientThreshold',1, ...
                'InitialLearnRate',0.005, ...
                'LearnRateSchedule','piecewise', ...
                'LearnRateDropPeriod',125, ...
                'LearnRateDropFactor',0.2, ...
                'Verbose',0, ...
                'Plots','training-progress');

            net = trainNetwork(XTrain,YTrain,layers,options);
            Yb = predict(net, XTrain);
            RNN_RRMSE(ii) = sqrt(mean((Yb-YTrain).^2))/mean(YTrain);

            net = predictAndUpdateState(net,XTrain);
            for mi = 2:360
                if FL_NoDataFrac(mi,ii)>0.05
                    [net,YPred] = predictAndUpdateState(net,[FL_Water(mi-1,ii) FL_PPT(mi-1,ii) FL_TMT(mi-1,ii)]');
                    FL_Water_rnn(mi,ii) = YPred;
                end
            end

            % Method 3: ANN

            XXTrain = [FL_Water(temp_st:temp_ed-1,ii) FL_PPT(temp_st:temp_ed-1,ii) FL_TMT(temp_st:temp_ed-1,ii)]';
            YYTrain = FL_Water(temp_st+1:temp_ed,ii)';

            net0 = feedforwardnet(10);
            [net0,tr] = train(net0,XXTrain,YYTrain);
            Ya = net0(XXTrain);
            ANN_RRMSE(ii) = sqrt(mean((Ya-YYTrain).^2))/mean(YYTrain);

            for mi = 2:360
                if FL_NoDataFrac(mi,ii)>0.05
                    YYPred = net0([FL_Water(mi-1,ii) FL_PPT(mi-1,ii) FL_TMT(mi-1,ii)]');
                    FL_Water_ann(mi,ii) = YYPred;
                end
            end
        end
    end


FL_Water_MLR1 = mean(FL_Water_MLR,2);
FL_Water_rnn1 = mean(FL_Water_rnn,2);
FL_Water_ann1 = mean(FL_Water_ann,2);

startDate = datenum('01-01-1985');
endDate = datenum('01-01-2015');
xData = linspace(startDate,endDate,360);

plot(xData, FL_Water_MLR1, '+r')
hold on
plot(xData, FL_Water_rnn1, 'xb')
hold on
plot(xData, FL_Water_ann1, 'hg')
title('Filled and existed GIW area time-series in FL')
datetick('x','mmmyy','keeplimits')
legend('MLR', 'RNN', 'ANN')
ylabel('GIW pixels for each PRISM grid')

saveas(gcf,'Result_FL.png')
close(gcf)

%% Case: ND

num_ND = 64;

fileID = fopen(fullfile(currentdirectory,'\ND\ALL_Cohen_B.txt'),'r');
formatSpec = '%d ';
sizeA = [num_ND Inf];
A = fscanf(fileID,formatSpec,sizeA);
fclose(fileID);
ND_All = A';

fileID = fopen(fullfile(currentdirectory,'\ND\Water_Cohen_B.txt'),'r');
formatSpec = '%d ';
sizeA = [num_ND Inf];
A = fscanf(fileID,formatSpec,sizeA);
fclose(fileID);
ND_Water = A';
ND_Water_MLR = ND_Water;
ND_Water_rnn = ND_Water;
ND_Water_ann = ND_Water;
ND_Water_raw = ND_Water;

fileID = fopen(fullfile(currentdirectory,'\ND\NoData_Cohen_B.txt'),'r');
formatSpec = '%d ';
sizeA = [num_ND Inf];
A = fscanf(fileID,formatSpec,sizeA);
fclose(fileID);
ND_NoData = A';

fileID = fopen(fullfile(currentdirectory,'\ND\PPT_Cohen_B.txt'),'r');
formatSpec = '%f ';
sizeA = [num_ND Inf];
A = fscanf(fileID,formatSpec,sizeA);
fclose(fileID);
ND_PPT = A';

fileID = fopen(fullfile(currentdirectory,'\ND\TMT_Cohen_B.txt'),'r');
formatSpec = '%f ';
sizeA = [num_ND Inf];
A = fscanf(fileID,formatSpec,sizeA);
fclose(fileID);
ND_TMT = A';

ND_NoDataFrac = ND_NoData./ND_All;
```

```
ND_Avail = ND_All(1,:)~=0;
num_ND_new = sum(ND_Avail);

MLR_R2 = zeros(num_ND,1);
RNN_RRMSE = zeros(num_ND,1);
ANN_RRMSE = zeros(num_ND,1);

for iik=1:num_ND
    if ND_Avail(iik)==1
        ii = iik;

        %ii=1;

        temp_NoData = ND_NoDataFrac(:,ii)>0.05;
        ND_Water_raw(temp_NoData,1) = NaN;
        aa = zeros(360,1);
        for iii = 2:360
            if ND_NoDataFrac(iii,ii)>0.05
                if ND_NoDataFrac(iii-1,ii)<0.05
                    aa(iii) = 1;
                else
                    aa(iii) = aa(iii-1) + 1;
                end
            end
        end
        maxlen = max(aa);
        [r,c] = find(aa==maxlen);
        temp_st = r(1) - maxlen + 1;
        temp_ed = r(1);

        % Method 1: Multiple Linear Regression

        XX = [ND_Water(temp_st-1:temp_ed-1, ii) ND_PPT(temp_st:temp_ed,ii) ND_TMT(temp_st:temp_ed,ii)];
        YY = ND_Water(temp_st:temp_ed, ii);
        lm = fitlm(XX,YY);

        MLR_R2(ii) = lm.Rsquared.Ordinary;

        for mi = 2:360
            if ND_NoDataFrac(mi,ii)>0.05
                YYnew = predict(lm,[ND_Water(mi-1,ii) ND_PPT(mi-1,ii) ND_TMT(mi-1,ii)]);
                ND_Water_MLR(mi,ii) = YYnew;
            end
        end

        % Method 2: RNN

        XTrain = [ND_Water(temp_st:temp_ed-1,ii) ND_PPT(temp_st:temp_ed-1,ii) ND_TMT(temp_st:temp_ed-1,ii)]';
        YTrain = ND_Water(temp_st+1:temp_ed,ii)';

        numFeatures = 3;
        numResponses = 1;
        numHiddenUnits = 200;

        layers = [ ...
            sequenceInputLayer(numFeatures)
            lstmLayer(numHiddenUnits)
            fullyConnectedLayer(numResponses)
            regressionLayer];

        options = trainingOptions('adam', ...
            'MaxEpochs',250, ...
            'GradientThreshold',1, ...
            'InitialLearnRate',0.005, ...
            'LearnRateSchedule','piecewise', ...
            'LearnRateDropPeriod',125, ...
            'LearnRateDropFactor',0.2, ...
            'Verbose',0, ...
            'Plots','training-progress');

        net = trainNetwork(XTrain,YTrain,layers,options);
        Yb = predict(net, XTrain);
        RNN_RRMSE(ii) = sqrt(mean((Yb-YTrain).^2))/mean(YTrain);

        net = predictAndUpdateState(net,XTrain);
        for mi = 2:360
            if ND_NoDataFrac(mi,ii)>0.05
                [net,YPred] = predictAndUpdateState(net,[ND_Water(mi-1,ii) ND_PPT(mi-1,ii) ND_TMT(mi-1,ii)]');
                ND_Water_rnn(mi,ii) = YPred;
            end
        end

        % Method 3: ANN

        XXTrain = [ND_Water(temp_st:temp_ed-1,ii) ND_PPT(temp_st:temp_ed-1,ii) ND_TMT(temp_st:temp_ed-1,ii)]';
        YYTrain = ND_Water(temp_st+1:temp_ed,ii)';

        net0 = feedforwardnet(10);
        [net0,tr] = train(net0,XXTrain,YYTrain);
        Ya = net0(XXTrain);
        ANN_RRMSE(ii) = sqrt(mean((Ya-YYTrain).^2))/mean(YYTrain);

        for mi = 2:360
            if ND_NoDataFrac(mi,ii)>0.05
                YYPred = net0([ND_Water(mi-1,ii) ND_PPT(mi-1,ii) ND_TMT(mi-1,ii)]');
                ND_Water_ann(mi,ii) = YYPred;
            end
        end
    end
end
```

```
ND_Water_MLR1 = mean(ND_Water_MLR,2);
ND_Water_rnn1 = mean(ND_Water_rnn,2);
ND_Water_ann1 = mean(ND_Water_ann,2);

startDate = datenum('01-01-1985');
endDate = datenum('01-01-2015');
xData = linspace(startDate,endDate,360);

plot(xData, ND_Water_MLR1, '+r')
hold on
plot(xData, ND_Water_rnn1, 'xb')
hold on
plot(xData, ND_Water_ann1, 'hg')
title('Filled and existed GIW area time-series
    in ND')
datetick('x','mmmyy','keeplimits')
legend('MLR', 'RNN', 'ANN')
ylabel('GIW pixels for each PRISM grid')

saveas(gcf,'Result_ND.png')
close(gcf)
```