CE-657

PROJECT REPORT

"STREAM FLOW PREDICTION USING NON-LINEAR APPROACH (PHASE SPACE RECONSTRUCTION)"

Submitted

by

Abdul Wajed Farhat
(203041012)

Supervisor:

Prof. Riddhi Singh

Indian Institute of Technology Bombay
Department of Civil Engineering
Mumbai, 400076
October-2021

Contents

List of Figures:

1. Key Message(s):

1) Prediction of stream flow using nonlinear methods
2) Checking if the chaos is present in data or not
3) Finding an embedding dimension

2. Abstract

Stream flow prediction is one of the important aspects of hydrology. Its importance increases day by day as the influence of severe stream flow is increasing and affecting much more the lives of creature's day by day. Stream flow prediction is important for understanding the occurrence of flood, its time and quantity, its also important for designing the hydraulic structures or planning for agricultural irrigation.

In recent decades many scholars worked on this topic. Physically and black-box models are the two major category of flow prediction. As physically based models requires large amount of data and plenty of time and calculations, recently researchers are working on data-based models. Phase space reconstruction (PSR) is one of the data-based models which needs only one variable as input and give output according that.

A PSR-based model is used in report for predicting the stream flow. As this method requires doing the same procedure many and many times and doing it manually, definitely cause big errors, MATLAB is a very good choice for this purpose.

3. Introduction

Stream flow is occurring due to several and complex hydrological processes, which modeling is a very important aspect of hydrology. Prediction of stream flow has an important role on controlling the flood and minimizing the risk of flooding. Also, for construction of dams and optimization of reservoirs a better understanding of stream flow dynamics and its prediction is an important task.

There are multiple physical mechanisms acting on stream flow dynamic and has a high effect on its occurrence. However, collecting data about all these mechanisms and physical characteristics of the catchment is not an easy task, so researchers are always looking for a more convenient and easy way to model streamflow. From the past decades researchers approached several stochastic methods for predicting the stream flow.in the last two decades researchers used non-linear and chaotic approaches for understanding and predicting the stream flow. As the non-linear based methods only work with single variable input and give the output as a result, it's an easy and faster model to use for forecasting the stream flow.

4. Method

3.1 Phase space reconstruction

Phase space reconstruction is a very useful tool for characterizing the dynamical systems, such as rivers streamflow. The data from all stations are reconstructed for embedding dimensions ranging from 2 to 10 and delay time of 1 to 10.

$$Y_{j=}(X_j, X_{j+\tau}, X_{j+2\tau}, \dots\dots, X_{j+(m-1)\tau})$$

Where $j = 1,2, \dots, N-(m-1)\tau$, m is the dimension of vector Y_j , and called as embedding dimension and τ is the delay time.

3.2 Prediction using Phase space reconstruction

From a given single variable time series of X_i, $i = 1,2,3, \dots N$ the value X_{N+1} is predicted. Firs we assume an embedding dimension of 2 and a time delay of 1. We construct the phase space $Y_{j=}(X_j, X_{j+\tau}, X_{j+2\tau}, \dots\dots, X_{j+(m-1)\tau})$ according to the embedding dimension and time delay we assumed early. Each Y_j in the reconstructed phase represents a state an is a vector. The distance between each vector (Y_j) is calculated. The vector which is needed and will be predicted is Y_{j+T} , and we assume a functional relation between Y_j and Y_{j+T} such that; $Y_{j+T} = F_T(Y_j)$ where T=1. The nearest neighbor of the last vector $Y_j = Y_{N-1}$ is find and according to the nearest neighbor of the last vector the X_{N+1} value is predicted.

1. Given a single-variable time series Xi, i = 1, 2, …, N, X_N+1 = ?
2. Assume an Embedding dimension (m = 2) and Delay (τ = 1)
3. Construct a multi-dimensional phase-space, Yj
4. Assume a functional relationship between Yj and Yj+T

$$Yj+T = FT(Yj) \rightarrow (T = 1)$$

5. Compute the distances between the vectors
6. Find neighbor (k) of Yj based on minimum values of ||Yj – Yj'||, j' < j

7. Prediction of Xj+T would be Xj'+T

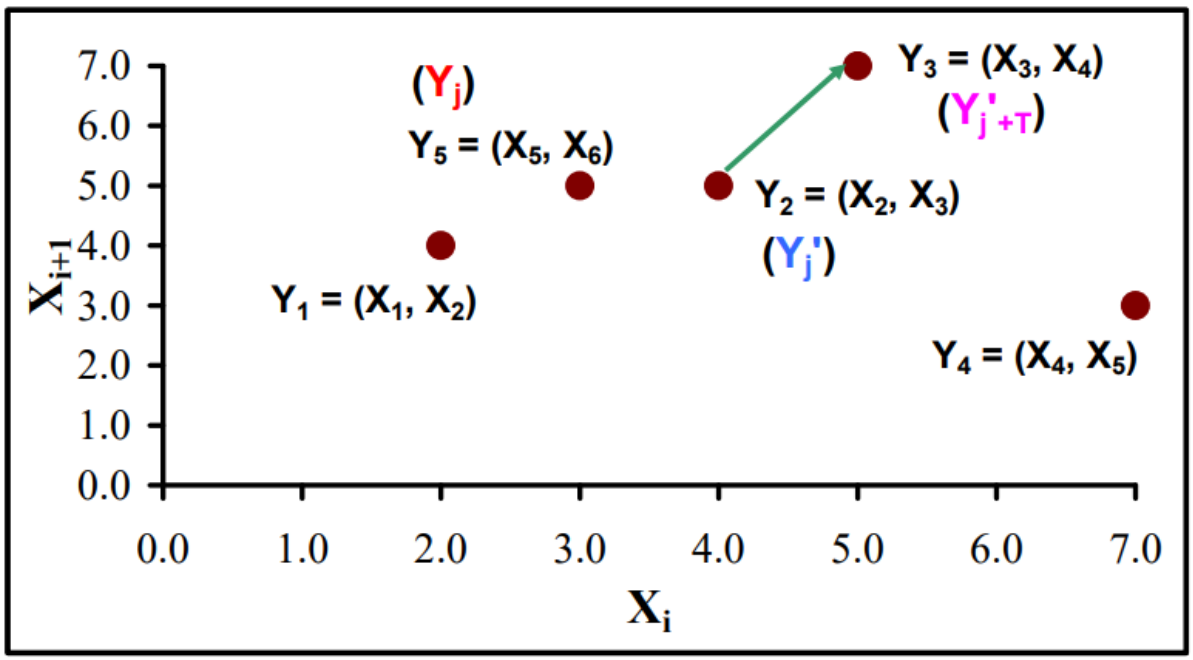

Figure 1. A simple example of phase space reconstruction m=2 & t=1 (Sivakumar, lecture ppts)

3.1 Prediction accuracy measuring method

3.3.1 Correlation coefficient:

Correlation coefficient is a method to examine how strong is the relationship between two variables. This method is applied in prediction models to check how close is the predicted values to the observed values. If O is the observed values and P is the predicted values, then the correlation coefficient of O and P will be:

$$CC = \left\{ \frac{\sum_{i=1}^{N}(O_i - \bar{O})(P_i - \bar{P})}{\left[\sum_{i=1}^{N}(O_i - \bar{O})^2\right]^{1/2}\left[\sum_{i=1}^{N}(P_i - \bar{P})^2\right]^{1/2}} \right\}$$

For the above formula $\bar{O}$ is the mean value of the observed values and $\bar{P}$ is mean value of the predicted values.

3.3.2 Root mean square error

Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. Root mean square error is commonly used in climatology, forecasting, and regression analysis to verify experimental results.

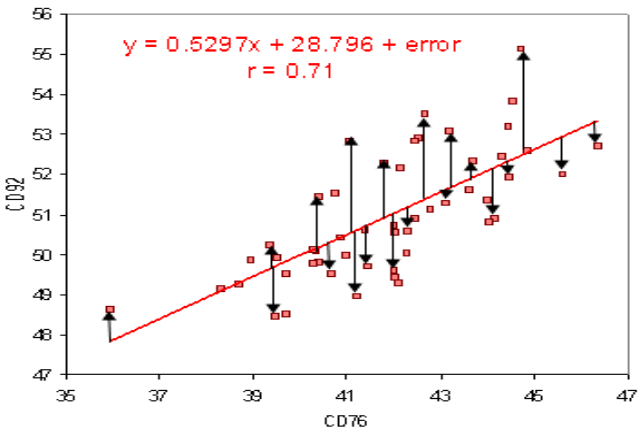

Figure 2. Residuals on a scatter plot. (source: nws.noaa.gov)

$$RMSE = \sqrt{\frac{\sum_i^N (O_i - P_i)^2}{N}}$$

5. Study area and data

The data used in this report is from USA, Fish River near Fort Kent, Maine (01013500) station downloaded from: https://waterdata.usgs.gov/nwis/rt. The period of data used in this report is from (1950-2021) and a total number of 856 monthly data is used. 75% percent of the data which is a total number of 641 monthly data is used as training data and remaining 25% of the data is used as testing data.

The time series and phase space (time delay=1 & embedding dimension =1) of the data is shown in figure below.

Codes in MATLAB:

```
%For plotting the time series of the data
load('flowdata.mat')
data=flowdata(:,1);
plot (data)
xlabel('time series')
ylabel('stream flow')
xlabel('Time (month)')
ylabel('Stream Flow (cfs)')
title ('Time Series Plot of the Station 01013500')


%For plotting the phase space of the data
plot(psr(:,1),psr(:,2))
xlabel('Xi')
ylabel('Xi+1')
title ('Phase Space Plot of the Station 01013500 for time delay=1 & m=2')
```

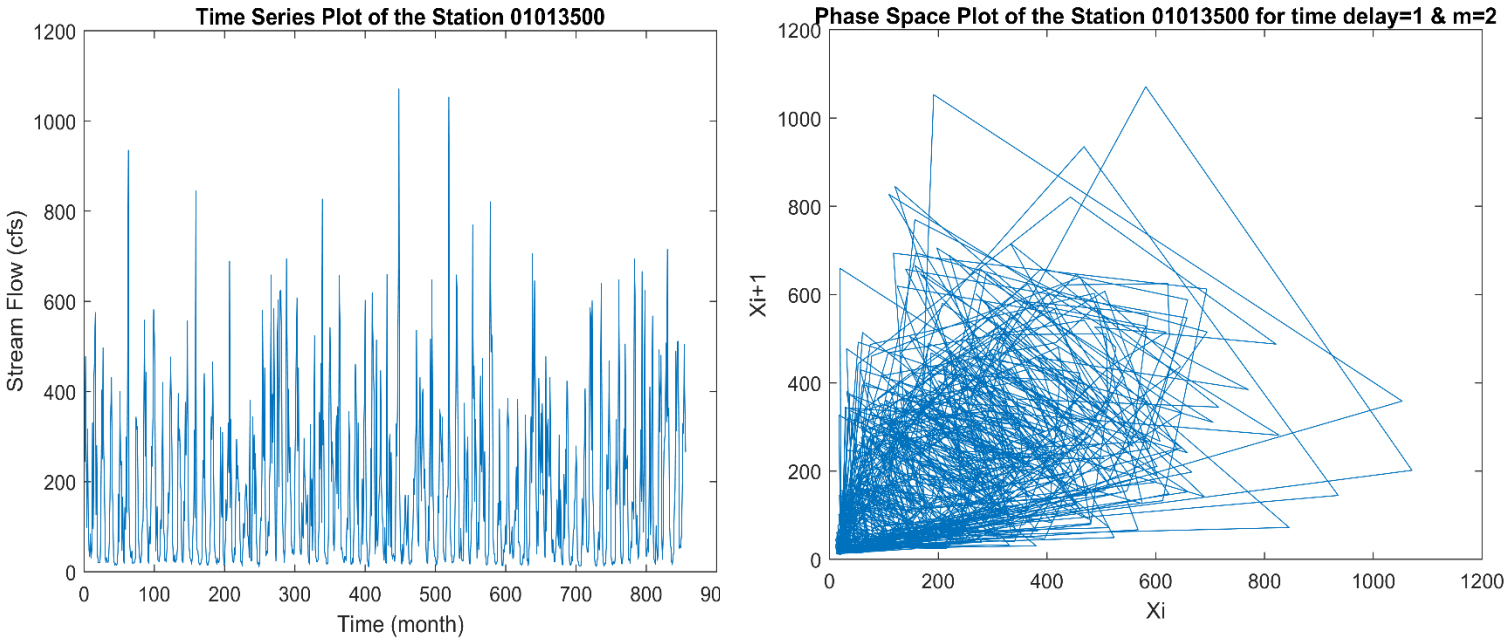

Figure 3. Time series and phase space plot of the data (t=1, m=2)

6. Procedure of work

The data used in this report is downloaded from USGS website. Stations starting with 01 is being studied in this report. The data for these stations is downloaded from the website, then by using MATLAB some modifications are applied for that data.

As all the stations don't have data for the same period of time as other stations, so a fixed period of time is being considered. Stations which have data shorter than that period is not taken under study and only stations with that period of data are being chosen. From chosen stations there may be some stations which may have missing data, so by applying some code in MATLAB those stations are also identified and being deleted. The final stations which are remained, the study is applied on those stations.

Procedure in MATLAB after downloading the data can be as follow:

1. **Codes for opening the series of stations data at once**
2. Opening the file (01stations) which contains the name of stations file data
3. Opening all stations using the name provided by (01stations) file
4. Changing the table format to array format so work can be done easily on it
5. Setting a fixed start date
6. Checking if the data contains the year of starting date
7. Using IF condition, for finding the stations which has data for years before than 1950 and also 1950
8. Storing the name of stations with data for that period of time.

Codes for the procedure mentioned above:

```
%%This code is written for reading the series of .txt files of stations
% and finding the stations with data starting earlier than 1950
%For opening the file which contains the name of sites.
sites=readtable('01sites.txt');
%For finding the length of the data
n=length(table2array(sites));
%Changing the number format to string format
sitesName=num2str(sites.Var1,'%08d');
x=table2array(sites);
for i=1:n
m            = readtable(sitesName(i,1:8));
data         = table2array(m(:,2:7));
startingdate = 1950; %fixing the starting date
%Checking that if the stations data has data for year 1950 or not.
rownumber    = find(data(:,4)==startingdate);
%For checking that if "rownumber" has value or is empty
tf           = isempty(rownumber);
%If condition for finding the stations which has data earlier than 1950 and
%also has data for year 1950
if data(1,4)< startingdate(1,1) && tf~=1
    x(i,2)   = 1;
else
    x(i,2)   = 0;
end
end
%%Storing the selected stations with proper span of data
row              = find(x(:,2)==1);
selectedstations = x(row,1);
```

2. Second code for choosing stations with no missing data

From first part of code, we find the stations with specific range of data, now in this part of code the stations are selected which has not any missing data, and for that we follow the following procedure of coding:

1. Change the format of array (selectedstations) from number to string
2. Find the number of stations
3. Declare an array for storing the name of stations which has no missing data
4. Start a for loop which is from 1 to number of stations
5. Read the data of stations
6. Store the data as array in different array name
7. Set the fixed starting date
8. Find the row which intersects with the starting date
9. Find the ending date in the data
10. Count the theoretical length of data
11. Count the actual length of data
12. If both counting's are same, then there is no missing data, if it doesn't match then the stations has missing value.
13. Store the name of stations which has complete number of data

Codes for the procedure mentioned above

```
%%For finding the stations with no missing data
%For changing the extension of selected sites from number to string
selectedsitesName=num2str(selectedstations,'%08d');
numberofstations=length(selectedsitesName); %Finding Number of stations
%For finding the stations with no missing data
Finalstations=zeros(numberofstations,2);
Finalstations(1:numberofstations,1)=selectedstations;
for i=1:numberofstations
stationstable = readtable(selectedsitesName(i,1:8));
data          = table2array(stationstable(:,2:7));
startingdate  = 1950;
row           = find(data(:,4)==startingdate(1,1));
finaldata     = data(row(1,1):length(data),:);
firstdate     = finaldata(1,4:5);
lastdate      = finaldata(size(finaldata,1),4:5);
count         =((lastdate(1,1)-firstdate(1,1))*12)-
firstdate(1,2)+lastdate(1,2)+1;
if count == length(finaldata)
    Finalstations(i,2)=1;
else
   Finalstations(i,2)=0;
end
end
```

3. For storing all stations data in one .mat file for further using

1. Change the format of array which contains the name of stations, from number to string
2. Find the length of the data
3. Declare an array for storing the data
4. Using for loop take the data from 1950 to the latest date
5. As some of the stations latest date is different, find the stations with shortest period and save all the data from all other stations according to that period of time.
6. Finally save the file as .mat file for later using.

```
%%This code is for storing the data of all stations in one .mat file
Data          = num2str(Finalstations(:,1),'%08d');
numstations   = length(Data);
Finaldata01   = zeros(72*12,numstations);
for i=1:numstations
    read      = readtable(Data(i,1:8));
    store     = table2array(read(:,2:7));
    rowN      = find(store(:,4)==1950);
    Finaldata01(1:length(store)-rowN+1,i)=store(rowN:length(store),6);
end
rowno=find(Finaldata01(:,:)==0);
Finaldata01=Finaldata01(1:(rowno(1,1)-1),:);
%Saving the data in .mat file for later using
file='flowdata.mat';
save (file,'Finaldata01');
```

4. Function for prediction

1. First the function for one value and one step prediction is coded which can be used for looping to perform the prediction for further values
2. Declare the output of function
3. Declare the input of function, which is the data which the prediction will be performed on, time delay and embedding which are two important parameters for using PSR prediction.
4. Reconstruct the phase space for the data
5. Declare a zero matrix for distance
6. Using a for loop find the distance between the vectors
7. Find the nearest neighbor to the last vector
8. Find the predicted value using the nearest neighbor

```
%%Function code for predicting one step ahead, using phase space
%reconstruction
function [predictedvalue,nearestneighbor,pred,linearindices,psr] =
predict(x,timedelay,embeddingdim)
%% For phase space reconstruction
psr=phaseSpaceReconstruction(x,timedelay,embeddingdim);

%for finding the distance between the vectors
distance=zeros(size(psr,1),size(psr,1));
for i=1:size(psr,1)
  for   j=1:size(psr,1)
    distance(i,j)=sqrt(sum((psr(i,:)-psr(j,:)).^2));
  end
end
index=1:size(distance,1)+1:size(distance,1)*size(distance,1);
distance(index)=nan;
%For finding the nearest neighbour
nearestneighbor=min(distance(size(distance,1),:));

%%For predicting the values
linearindices=find (distance(size(distance,1),:)==nearestneighbor(1,1));
pred=(psr((linearindices(1,1)+1),size(psr,2)));
%For taking the average of the predicted values
predictedvalue=pred;
      end
```

5. Commands for using the prediction function

1. Loading the .mat file which contains the data of several stations
2. Select the station you want to perform the prediction on it
3. Set the training data length
4. Set the embedding dimension and time delay you want to perform prediction on.
5. Using for loop run the function which predict the values simultaneously.
6. Find the correlation and RMSE between the predicted and observed values
7. Store the correlation and RMSE results in an array.

```
%%For loading the data and declaring the training set with an array of
%%Containing the 75% of the actual data
load('flowdata.mat')
```

```
data=flowdata(:,1);
lengthofdata=length(data);
Ltraining=round(0.75*lengthofdata) %For finding the length of 75% of the data
x=data(1:Ltraining,1);
embdim=3;  %put the embeding dimension you want to perform the prediction for
tdelay=2;  %put the delay time you want to perform the prediction for
for i=1:(lengthofdata-Ltraining)
x(Ltraining+i-1,1)= data(Ltraining+i-1,1);
y(i,1)=predict(x,tdelay,embdim);
end
%% For model accuracy check
R(embdim*2-1:embdim*2,:)=corrcoef(y,data(Ltraining+1:lengthofdata,1))
i=1:(lengthofdata-Ltraining);
RMSE(embdim,1)=sqrt(sum((data(Ltraining +i,1)-y(i,1)).^2)/size(y,1))
%% For storing the output of correlation and RMSE in one table;
%First coulumn is for correlation and second column is for RMSE
for embdim=2:10
results(embdim,tdelay*2-1)=R(embdim*2-1,2);
results(embdim,tdelay*2)=RMSE(embdim,1);
end
```

7. Results and discussion

The code is ran for different embedding dimension and different time delays for finding an optimum embedding dimension and best accuracy, the result is give in table 1. From table 1 we can see that the accuracy is changing as the value of m and time delay is changing. The change in accuracy due to change in m shows a chaos presence in data. From the left side of the table the optimum CC value is for m=3. The right part of the table is performed the prediction for different time delays and fixed embedding dimension and the final optimal result is found for time delay=2 and m=3.

	Time delay=1			Embedding dim=3	
Embedding dimension	correlation coefficient	RMSE	Time delay	correlation coefficient	RMSE
2	0.319214382	244.7291	1	0.352033	230.771
3	0.352032812	230.771	2	0.419871	231.0902
4	0.322928404	222.5168	3	0.311724	220.651
5	0.310948226	233.9888	4	0.391479	211.8461
6	0.310728675	246.7683	5	0.375037	220.3737
7	0.332866149	231.1243	6	0.2743	238.0027
8	0.270586266	218.1673	7	0.38415	233.7157
9	0.205393848	231.6742	8	0.289704	221.9872
10	0.314080827	225.9456	12	0.346181	232.0533

A comparison between the predicted and observed values is plotted in the figure 4. The predicted values follow the path of observed values and some of the peak and lowest points are predicted reasonably.

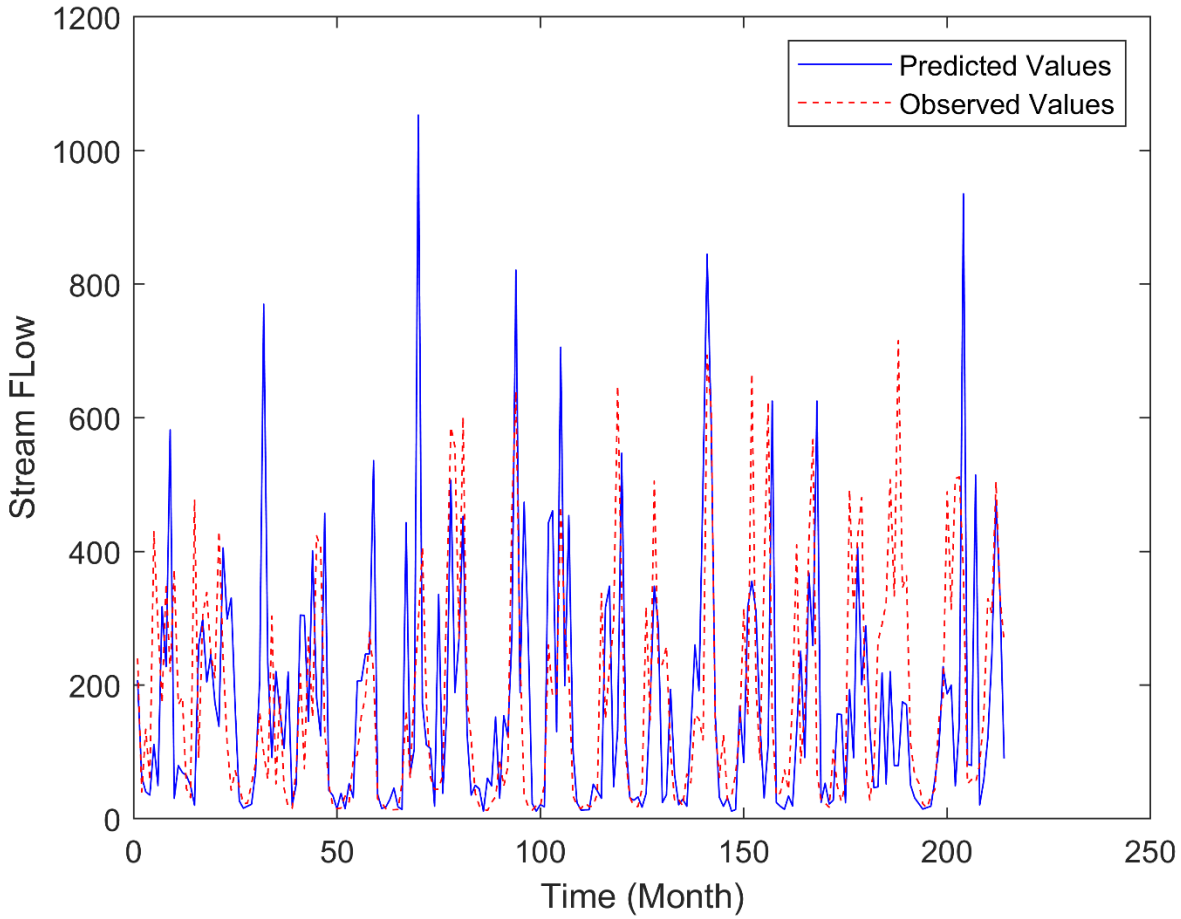

Figure 4. a comparison of the predicted and observed values for m=3 and time delay=2

Codes for figure 4 plot:

```
predicted=plot(1:length(y),y,'b');
hold on
observed=plot(1:length(y),data(Ltraining+1:lengthofdata),'--r');
hold off
legend([predicted, observed], 'Predicted Values', 'Observed Values')
xlabel('Time (Month)')
ylabel('Stream Flow')
```

8. Conclusion

In recent decades the nonlinear prediction methods have gain lots of attention and is applied for different streamflow data from different catchments. The results are so satisfying which encourages the researchers to improve nonlinear prediction methods. PSR is one of the best and suitable method for predicting the stream flow (Sivakumar, 2002) compared to another nonlinear methods like artificial neural network (ANN). The results

from (Sivakumar, 2003) paper and the results obtained from this report encourage to go further with nonlinear methods for more accurate and easier using method. The change in accuracy due to change in embedding dimension shows a presence of chaos in stream flow data and a chaotic method is best suited for stream flow prediction rather than stochastic based methods.

MATLAB was very useful for applying this method as this method requires doing a long procedure for more than 200 or 300 times, so doing manually is always very time consuming and full of errors, but fortunately MATLAB can do it only in seconds with zero percent of error in calculations.

9. References

1. Sivakumar, B. (2003). Forecasting monthly streamflow dynamics in the western United States: A nonlinear dynamical approach. *Environmental Modelling and Software*, *18*(8–9), 721–728. https://doi.org/10.1016/S1364-8152(03)00074-4
2. Sivakumar, B., Jayawardena, A. W., & Fernando, T. M. K. G. (2002). *River flow forecasting : use of phase-space reconstruction and artificial neural networks approaches*. *265*, 225–245.
3. Sivakumar, B., Berndtsson, R., & Persson, M. (2001). *Monthly runoff prediction using phase space reconstruction*. *46*(December).
4. Liu, Q., Islam, S., Rodriguez-Iturbe, I., & Le, Y. (1998). Phase-space analysis of daily streamflow: characterization and prediction. *Advances in Water Resources*, *21*(6), 463–475. https://doi.org/10.1016/S0309-1708(97)00013-4
5. Cai, W. D., Qin, Y. Q., & Yang, B. R. (2008). Determination of phase-space reconstruction parameters of chaotic time series. *Kybernetika*, *44*(4), 557–570.