다목적댐 운영을 위한
강수-유입량(10일) 빅데이터 분석 결과

□ 분석배경 및 목적

○ (배경) 수자원 관리를 위한 수많은 모형이 개발되어 사용되고 있으나 실측자료를 반영한 매개변수 최적화 및 토양수분, 융설 등 **여러 요인을 고려한 모델 구축**을 위해서는 **많은 노력이 필요함**

- 댐별로 매개변수 최적화 등의 작업이 필요
- 모델의 정확도 향상을 위해서는 토양수분, 융설 등을 입력 변수로 설정해야 되나 이와 관련된 정확한 데이터 부재시 오차 발생

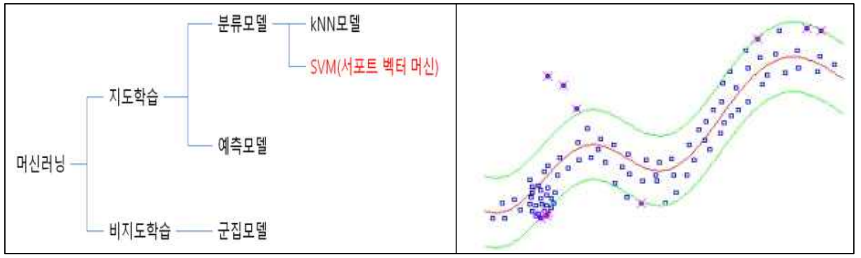
○ (목적) 방대한 자료의 축적과 분석기법의 발달로 새로운 수자원 관리 기법 필요

- (환경변화) 인공신경망, 딥러닝 등 자료를 활용하여 의미 있는 결과를 도출하는 기법이 다양해지고, 수자원 분야에서 **방대한 관측 자료가 축적되는** 등으로 예전과는 다른 분석환경이 조성
  * 댐(다목적댐)별로 최소 12년에서 최대 43년의 자료가 확보되어 있음
- (필요성) 기존의 경험식이나 물리모형이 아닌 데이터 자체를 분석하여 기존보다 간편하고 정확한 수자원 관리방안에 대한 **새로운 접근이 필요**
  * 데이터 기반의 분석은 댐별로 별도의 작업이 필요하지 않으며, 학습에 의해 결과를 도출함으로 새로운 자료의 반영이 간편함

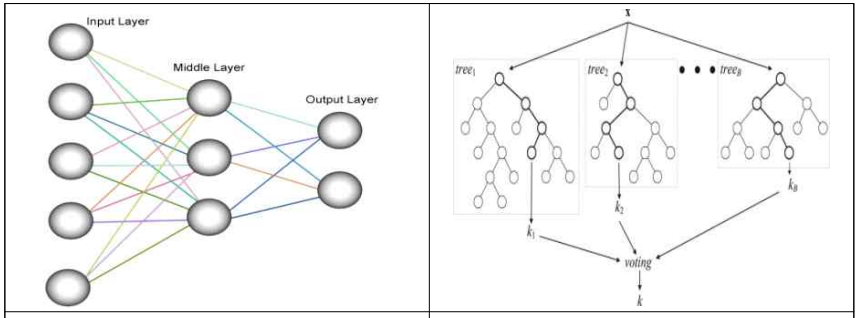
□ 분석방법

○ (분석기법) 기계학습의 방법으로 SVM, ANN, Random Forests 기법을 비교 검토하여 최종 분석 방법 선정

- SVM(Support Vector Machine) : 기계학습의 분류모델 중 하나로서 분류(Classification)뿐만 아니라 회귀(Regression)분석에도 적용되어 우수한 해결 능력을 보여주는 알고리즘
  * SVM(Support Vector Machine) 비선형 회귀모형 활용
  * SVM은 복잡한 비선형 의사결정 영역을 모형화 할 수 있기 때문에 매우 정확하며, 다른 모델들 보다 Over Fitting되는 경향이 적음



- ANN : 인공지능의 한 분야인 인공신경망(ANN)은 생물학(통상 인간)의 뇌 구조(신경망)를 모방하여 모델링한 수학적 모델로서, 병렬성이 뛰어나고 일정 수준의 오류에 강하며 주어진 환경에 대한 학습능력이 있는 알고리즘(알파고 등으로 최근에 많이 알려진 딥러닝은 인공신경망의 최신 버전이라 볼 수 있음)
- Random Forests : 여러개의 의사결정나무들로 구성된 모형으로 표본을 다수 생성하고 이를 적용하여 그 결과를 종합하는 앙상블 방법(의사결정나무모형들이 서로 독립일수록 예측오차가 작아지고 의사결정나무 수가 많아도 과적합하지 않는 장점이 있음)



ANN	Random Forests

○ (분석 Tool) 라이센스의 제약이 없고 다양한 패키지를 제공하여 분석의 편의성을 제공하는 R 패키지를 활용

○ (변수) 10개의 설명변수(영향인자 : 강우)를 입력 자료(일자료)로 사용하여 10일 평균 유입량을 예측하는 알고리즘 사용

- INPUT : 선행강우(1~3일), 누적강우(3일), 지체유입량(1~3일), 유효강우일수, 홍수기구분(5월~10월), 10일 평균 예측강우량

  * 10일 평균 예측강우량이 입력변수로 사용되므로 강우예측오차로 인해 정확도 영향(무강우시 영향 X)

- OUTPUT : 10일 평균 유입량

○ (대상지역) 전국의 다목적댐(15개)을 대상으로 분석실시

  * 2012년 이후 준공된 5개 다목적댐(군위댐, 김천부항댐, 보현산댐, 성덕댐, 영주댐) 제외(5년 이하의 자료 확보로 분석에서 제외)

댐명	유역면적 (㎢)	자료기간	댐명	유역면적 (㎢)	자료기간
소양강	2,703	'74~'16(43yr)	대청	3,204	'81~'16(36yr)
충주	6,648	'86~'16(31yr)	용담	930	'01~'16(16yr)
횡성	209	'00~'16(17yr)	주암(본)	1,010	'91~'16(26yr)
안동	1,584	'77~'16(40yr)	섬진강	763	'75~'16(42yr)
임하	1,361	'92~'16(25yr)	보령	164	'98~'16(19yr)
합천	925	'89~'16(28yr)	부안	59	'97~'16(20yr)
남강	2,285	'00~'16(17yr)	장흥	193	'05~'16(12yr)
밀양	95	'01~'16(16yr)	-	-	-

○ (기계학습) 전체 자료 중 2016년 자료를 검증 데이터로 사용하고 나머지 데이터는 학습에 사용

- 2015년까지 자료를 Training 자료로 사용하여 알고리즘을 학습
- Training기간 : ~'15 (11~42yr), Test기간 : '16 (1yr)

□ 분석결과

○ 댐유입량 예측을 위한 알고리즘으로 SVM(R패키지/e1071) 채택

- 분석방법(SVM, ANN, random forests)별 통계적 상관성은 거의 유사한 결과를 나타냄
- SVM으로 분석한 결과가 미미하지만 가장 좋은 것으로 나타나 분석 알고리즘으로 채택

  * SVM-Type : eps-regression, SVM-Kernel : radial

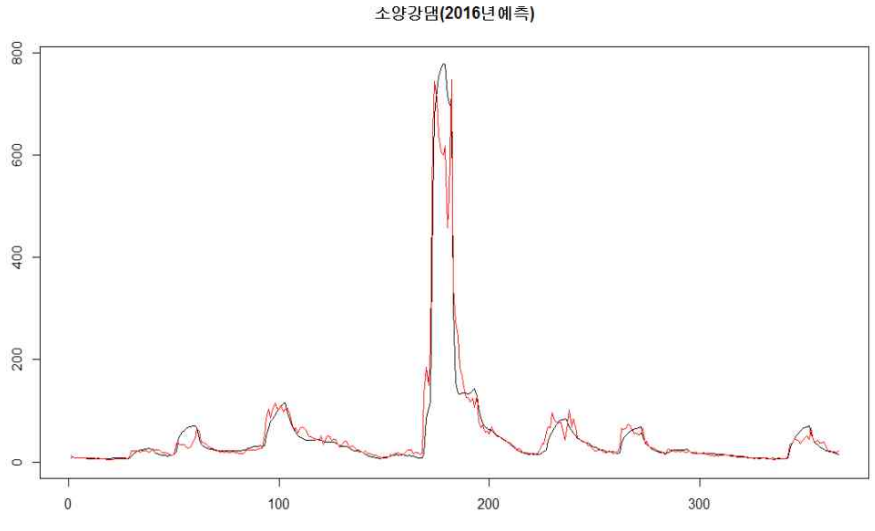
  * ANN(nnet패키지 사용)은 size 최적화 이전 결과

구분	SVM	ANN	random forests
상관계수(R)	0.94	0.94	0.92
결정계수(R^2)	0.87	0.85	0.82
RMSE(㎥/s)	15.18	16.46	19.21

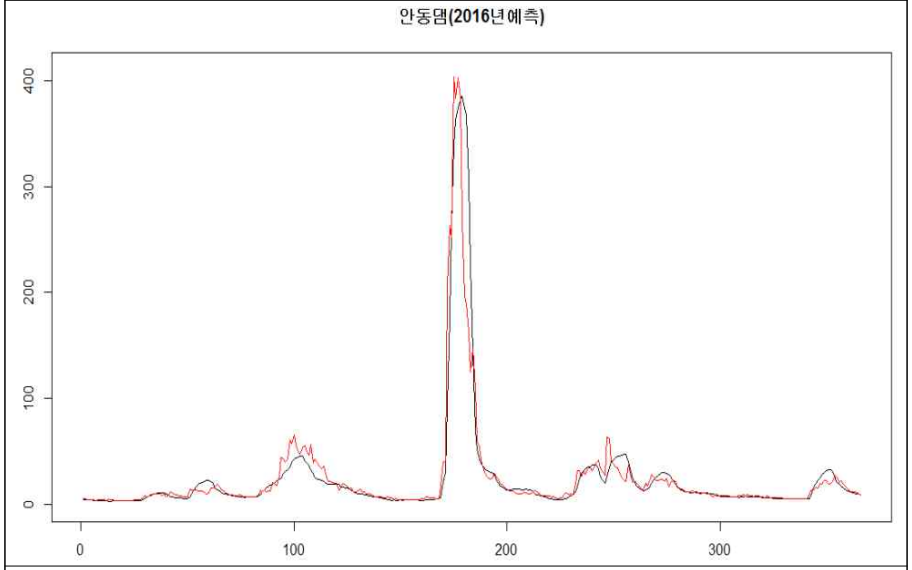
○ 10일 평균유입량 예측결과 상관계수(R) 0.94 결정계수(R2) 0.87 RMSE 15.2㎥/s로 분석되어 댐운영시 예측 모형으로 적용가능

  * 유역면적이 큰 다목적댐(소양강, 충주, 대청 등)에서 홍수기 유량차이로 인해 RMSE 값이 다소 높게 산정되나 대부분의 다목적댐은 10(㎥/s) 이하로 나타남

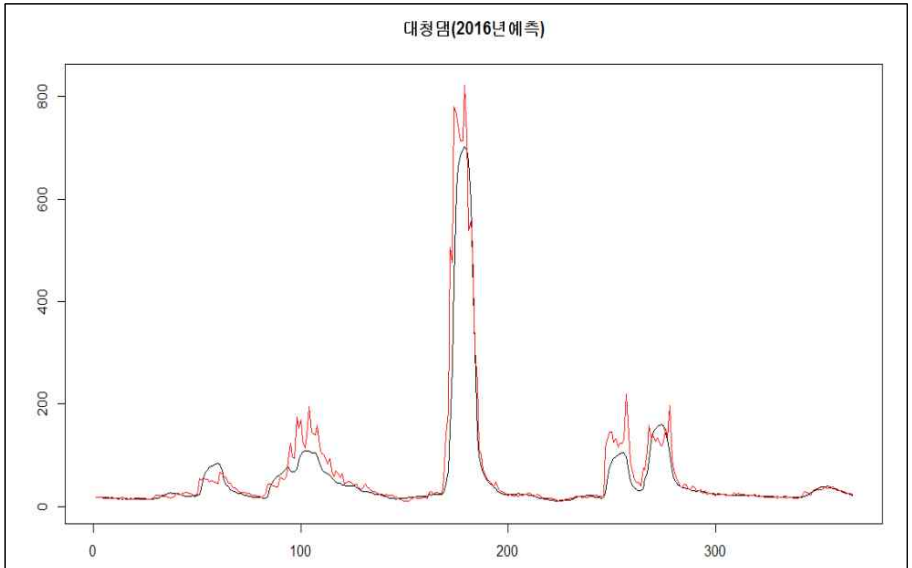
댐명	상관계수 (R)	RMSE (㎥/s)	결정계수 (R^2)	댐명	상관계수 (R)	RMSE (㎥/s)	결정계수 (R^2)
대청	0.96	37.6	0.87	소양강	0.97	30.0	0.93
용담	0.95	8.4	0.89	충주	0.94	67.7	0.86
주암(본)	0.93	8.9	0.85	횡성	0.95	1.4	0.90
섬진강	0.92	8.1	0.81	안동	0.93	20.2	0.87
보령	0.95	1.6	0.85	임하	0.95	10.1	0.86
부안	0.90	0.7	0.73	합천	0.96	6.1	0.91
장흥	0.95	1.8	0.91	남강	0.94	23.8	0.86
				밀양	0.95	1.3	0.89
평균	0.94	15.2	0.87				



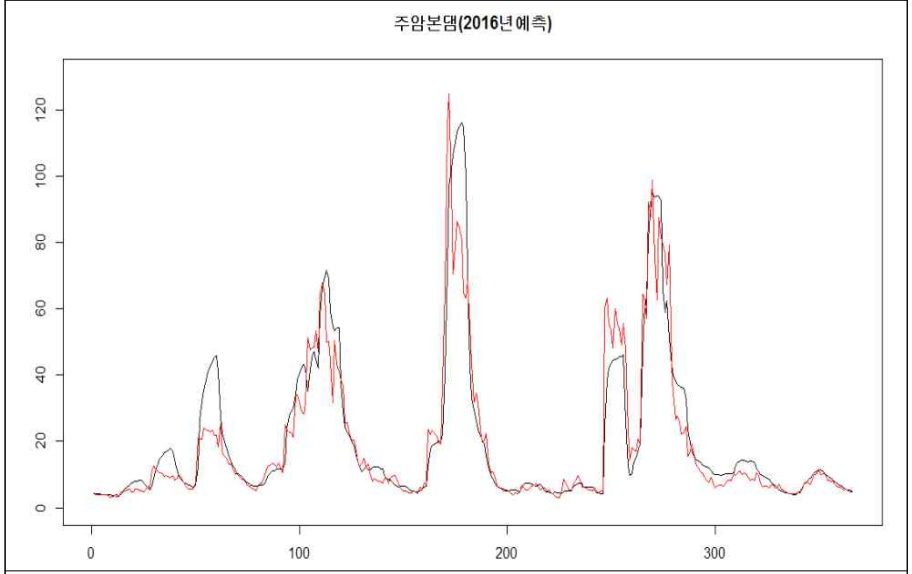
소양강댐(한강) 유입량(10일 평균) 예측결과



안동댐(낙동강) 유입량(10일 평균) 예측결과

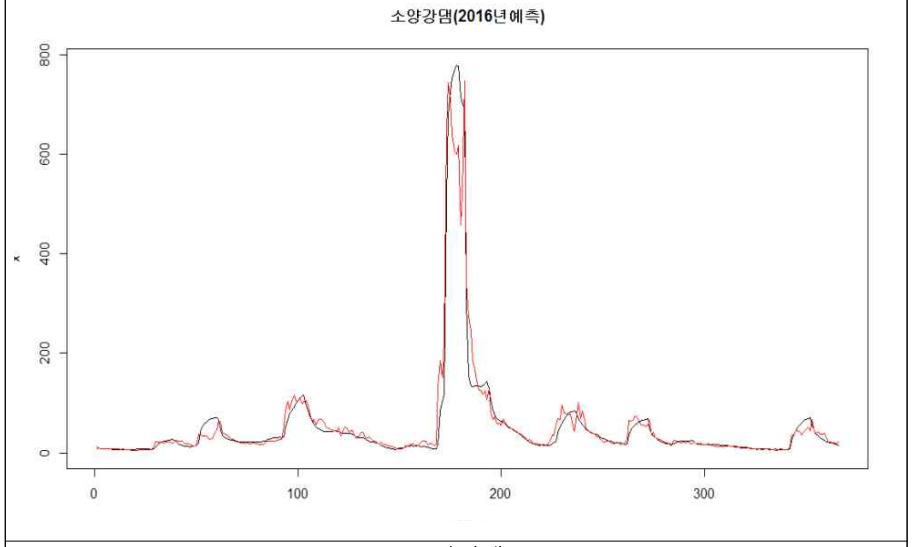


대청댐(금강) 유입량(10일 평균) 예측결과

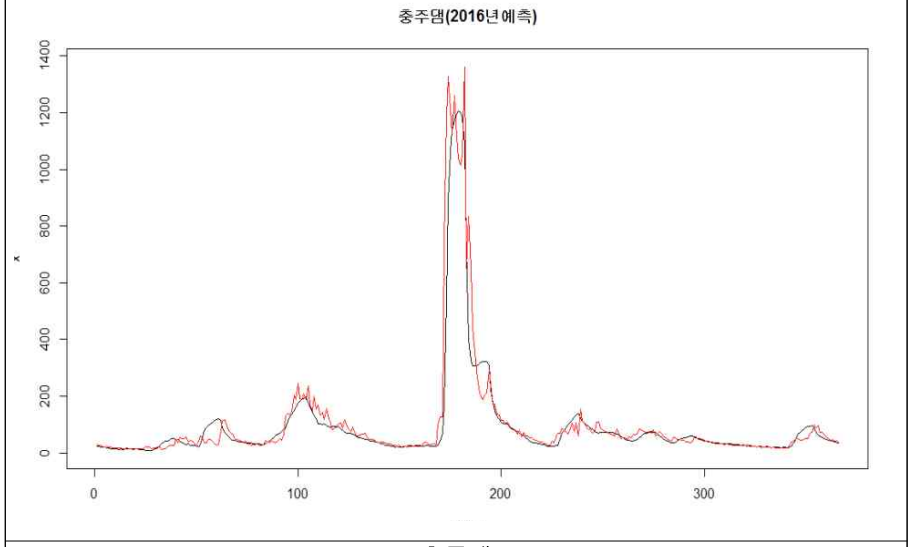


주암댐(섬진강) 유입량(10일 평균) 예측결과

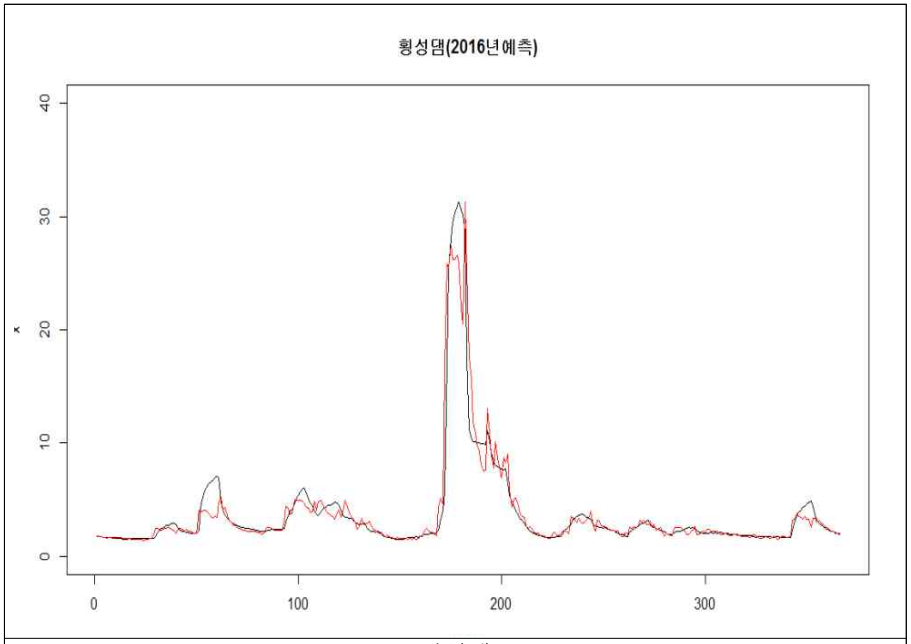
붙임1 **10일 평균 유입량 예측 결과**



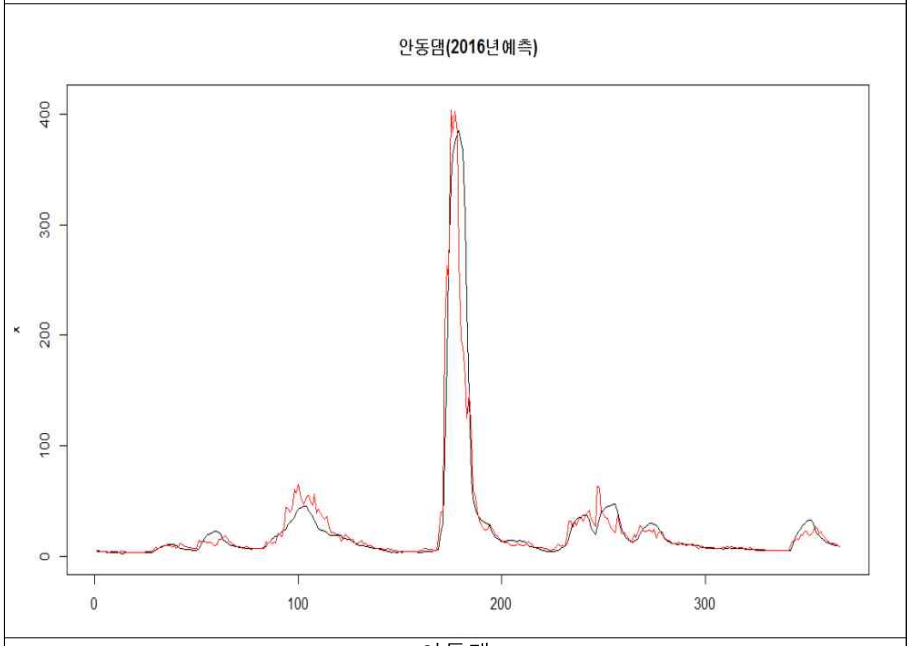
소양강댐



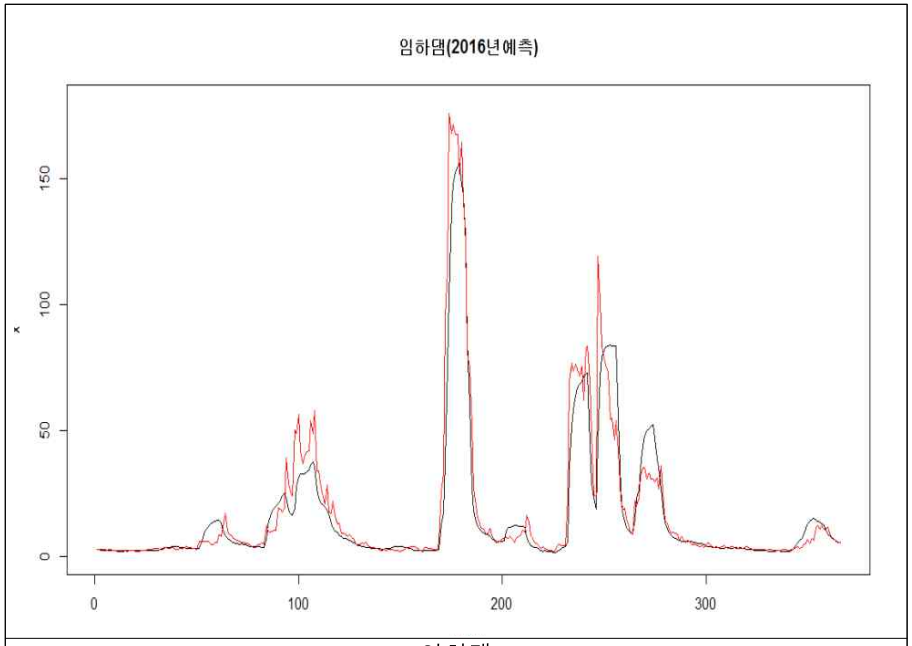
충주댐



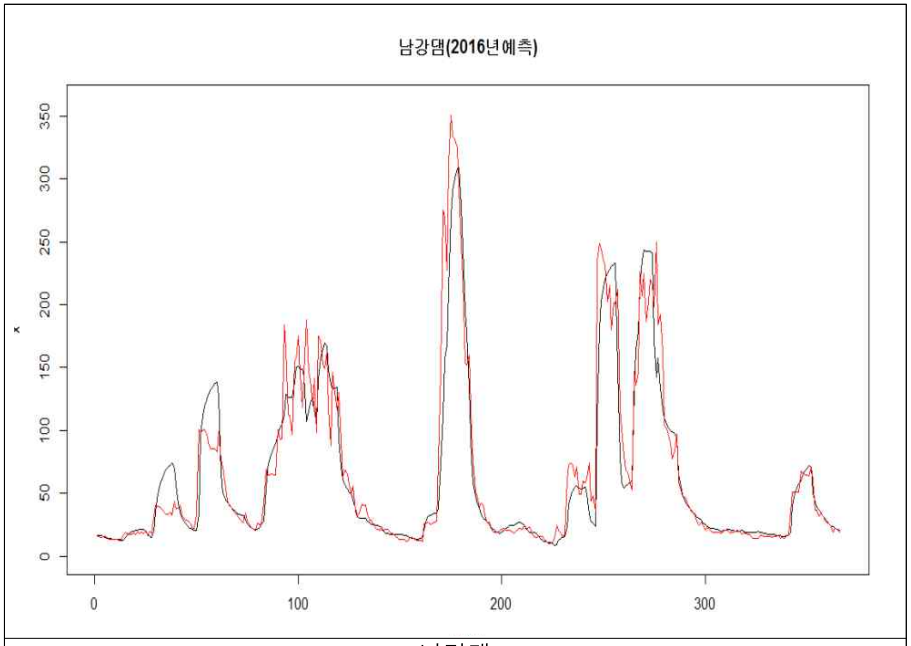
횡성댐



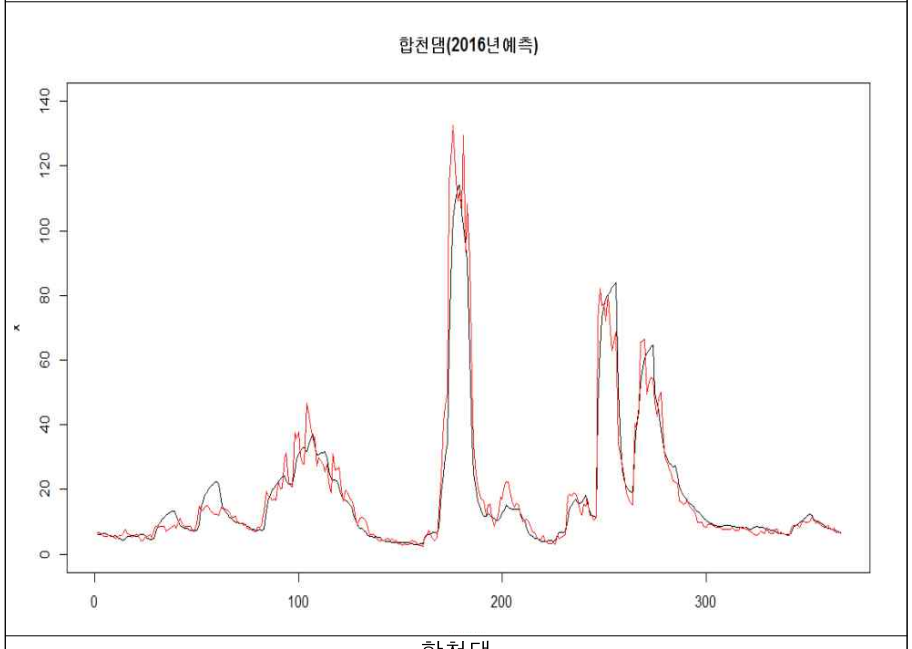
안동댐



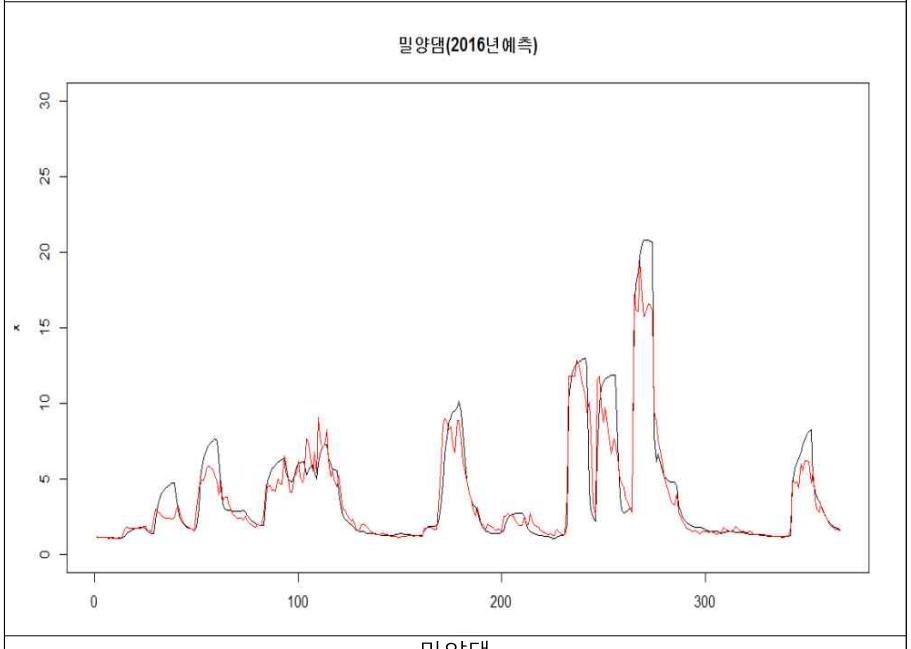
임하댐



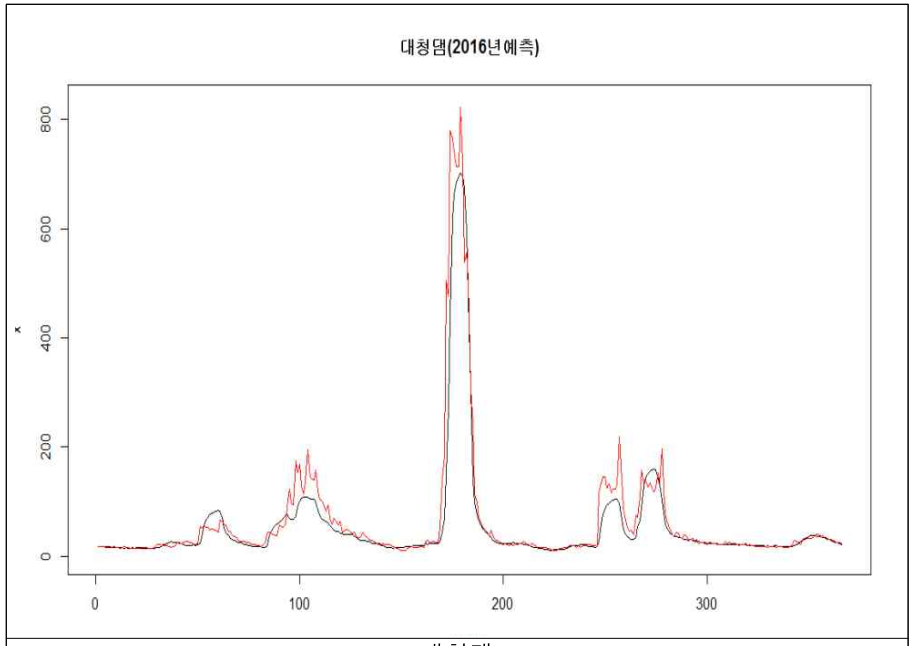
남강댐



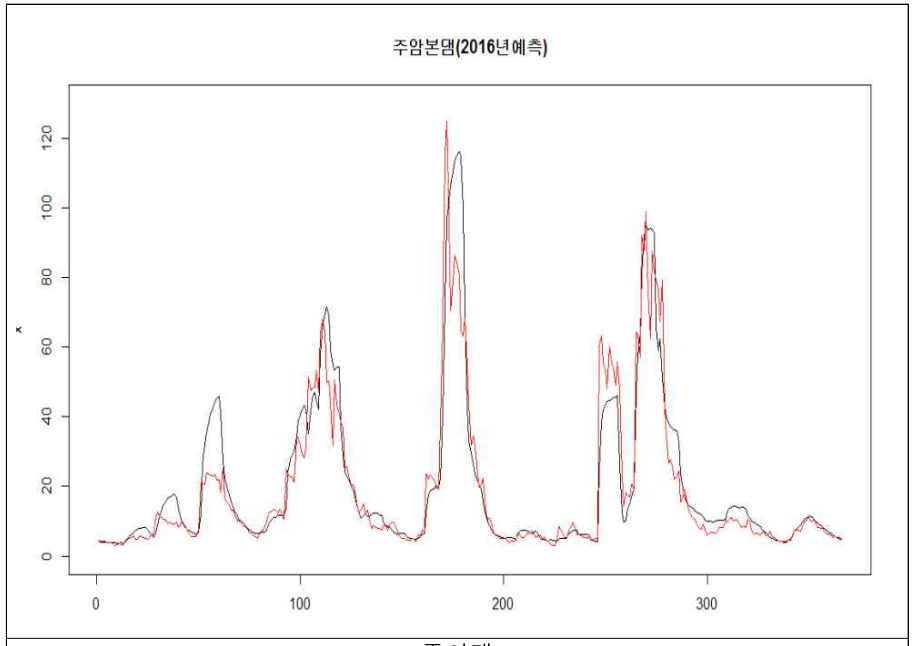
합천댐



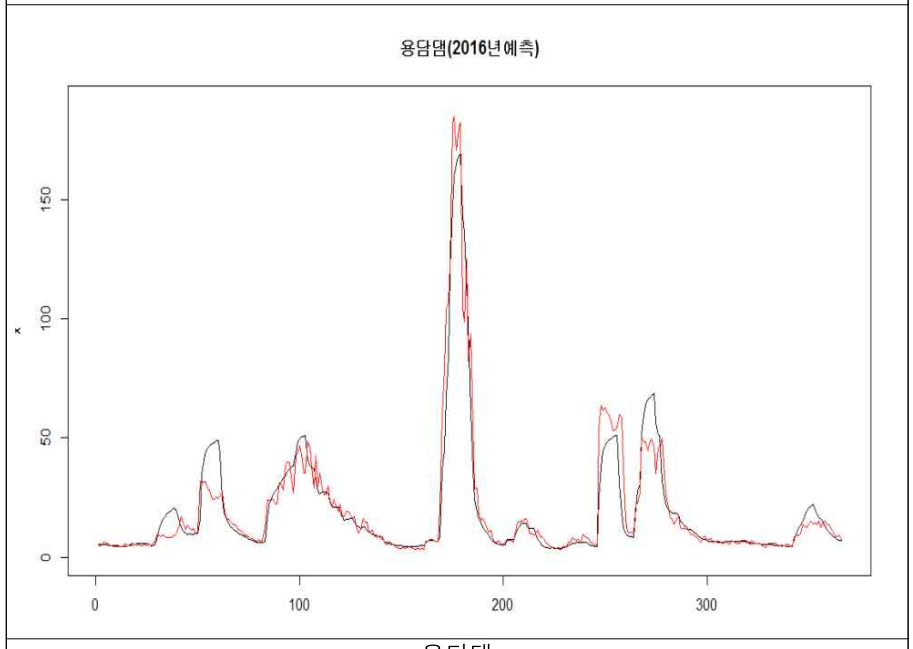
밀양댐



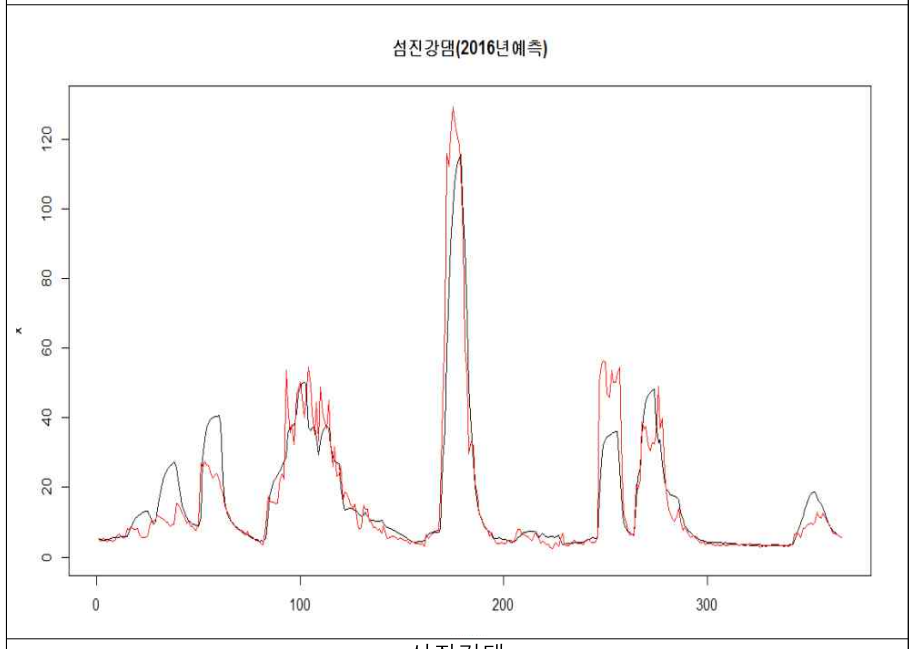
대청댐



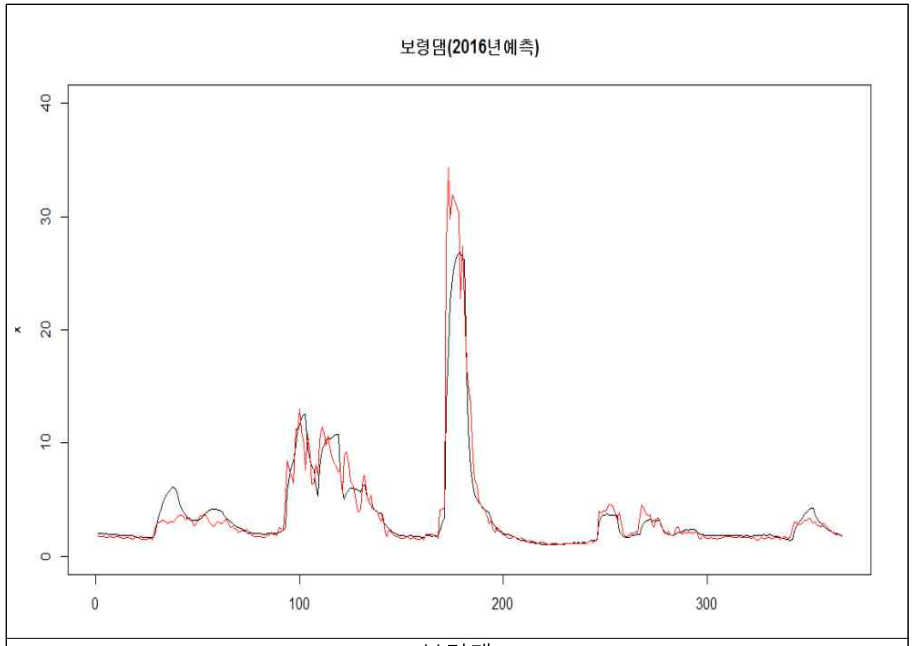
주암댐



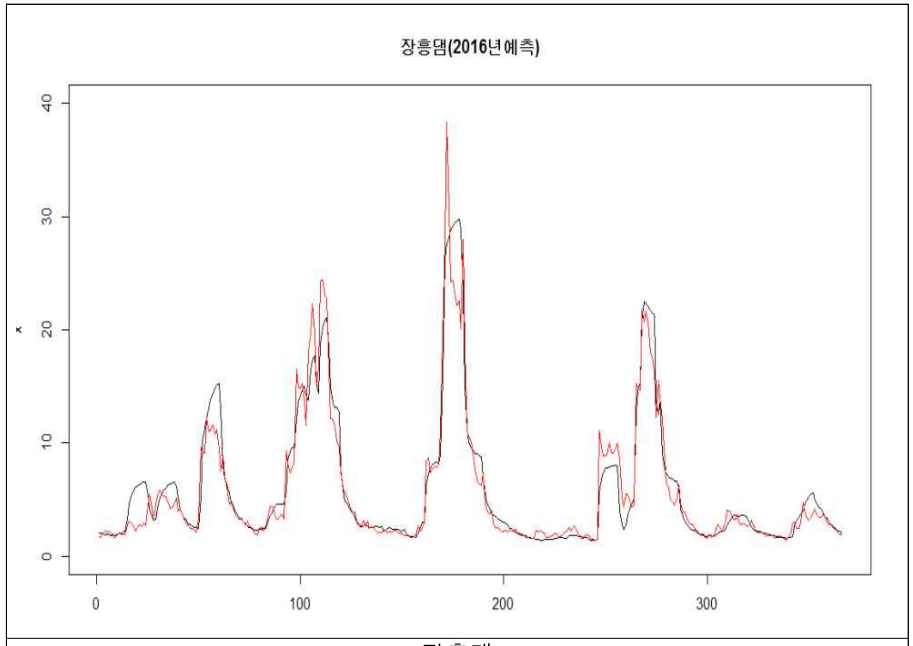
용담댐



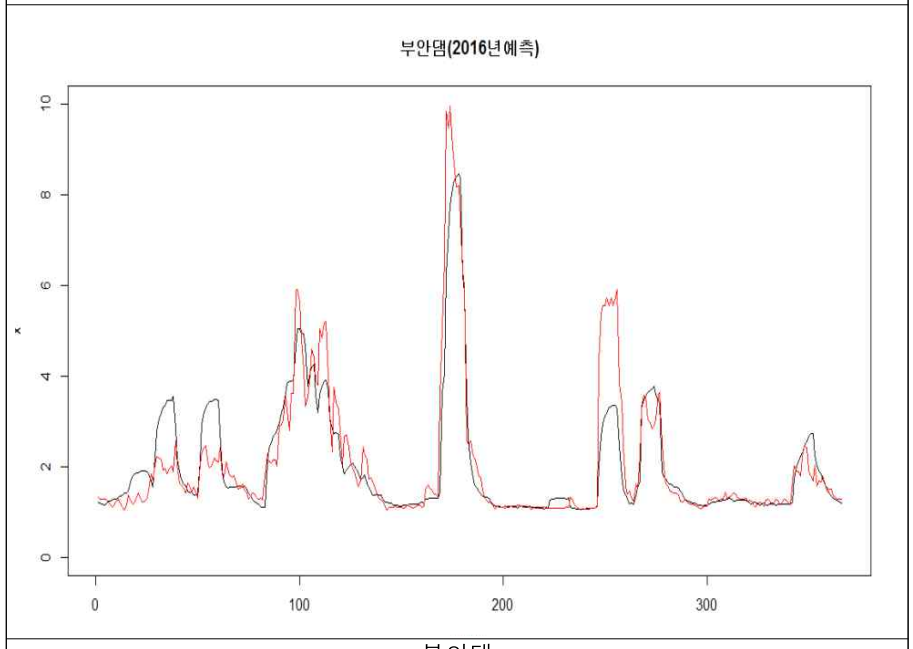
섬진강댐


보령댐(2016년예측)
x
0
10
20
30
40
0
100
200
300

보령댐


장흥댐(2016년예측)
x
0
10
20
30
40
0
100
200
300

장흥댐


부안댐(2016년예측)
x
0
2
4
6
8
10
0
100
200
300

부안댐

붙임2 분석 코드

```
##SVM 패키지 로드
install.packages("e1071")
library(e1071)

##경로설정
setwd("D:/test/1024_dam/result1107_1")
getwd()
remove(list = ls())

##입력자료 업로드(CSV파일, 동일 폴더에 분석대상 15개 댐 자료 배치)
temp = list.files(pattern="*.csv")
li=length(temp)
for(i in 1:li){
    a=temp[i]
    b=nchar(a)
    df0=read.csv(paste(a, sep=""),header = TRUE)
    na1=substr(a,1,(b-4))
na2=which(df0[,1]=="20161231")

##데이터전처리
df0[,5]=ifelse(((9-as.numeric(substr(as.character(df0[,1]),6,6)))%%9)>4,1,0)
sum(is.na(df0))
name=c("q_m","r_m","q1","q2","q3","r1","r2","r3","rs","d","season")
q=df0[,4]
rf=df0[,3]
rf_d=ifelse(rf<5,0,1)
rf=ifelse(rf<5,0,rf)
n=length(q)-5
df=as.data.frame(matrix(ncol = 11,nrow = n-5))
q_m1=as.data.frame(matrix(ncol = 10,nrow = n-5))
rf_m1=as.data.frame(matrix(ncol = 10,nrow = n-5))
for(i in 1:10){
    q_m1[,i]=q[c((5+i):(n+i-1))]
    rf_m1[,i]=rf[c((5+i):(n+i-1))]
}
q_m=apply(q_m1,1,mean)
r_m=apply(rf_m1,1,mean)
df[,1]=log(q_m+1)
df[,2]=r_m
for(i in 2:4){
    df[,i+1]=log(q[c((6-i+1):(n-i+1))]+1)
    df[,i+4]=rf[c((6-i+1):(n-i+1))]
}
df[,9]=df[,6]+df[,7]+df[,8]
df[,10]=rf_d[c(5:(n-1))]+rf_d[c(4:(n-2))]+rf_d[c(3:(n-3))]
df[,11]=df0[c(6:n),5]

df=df[c(1:na2),]
m=mean(df[,1])
v=sd(df[,1])
df_z=as.data.frame(scale(df))
colnames(df_z)=name
df_z1=df_z

##SVM분석
samp=c(1:(na2-366))
train.df1=df_z1[samp,]
test.df1=df_z1[-samp,]
svm.dam=svm(q_m~.,data=train.df1,cost=1)


##결과
summary(svm.dam)
cor(svm.dam$fitted,train.df1$q_m)
rmse(svm.dam$fitted,train.df1$q_m)
mae(svm.dam$fitted,train.df1$q_m)

df.pred1=predict(svm.dam, test.df1, type="class")
```

```
test=cbind(df.pred1,test.df1)
x=exp((test.df1$q_m*v)+m)
y=exp((df.pred1*v)+m)
y1=((max(max(x),max(y))%/%10)+1)*10
plot(x, main=paste(na1,"댐(2016년예측)",sep=""),type="l",ylim=c(0,y1))
lines(y,col="red")
result=as.data.frame(matrix(ncol = 4))
colnames(result)=c("cor","rmse","mae","r2")
result[,1]=cor(x,y)
R2 <- 1 - (sum((x-y)^2)/sum((x-mean(x))^2))
result[,2]=rmse(x,y)
result[,3]=mae(x,y)
result[,4]=R2
write.csv(result,file=paste("result_",na1,".csv",sep = ""),row.names=FALSE)
}

temp2 = list.files(pattern="result*")
li1=length(temp2)
a1=temp2[1]
b1=read.csv(paste(a1, sep=""),header = TRUE)
resul1=cbind(substr(a1,8,9),b1)
for(i in 2:li1){
    a1=temp2[i]
    b1=read.csv(paste(a1, sep=""),header = TRUE)
    c=cbind(substr(a1,8,9),b1)
    resul1=rbind(resul1,c)
}
write.csv(resul1,file = "result.csv",row.names = FALSE)
```