Name. LI ZHI
Matriculation No. A0169388H

Assignment 4

Data Pre-process

1. Data trend

Because there is no trend due to the initial graph of the time series and thus no need to detrend these data.

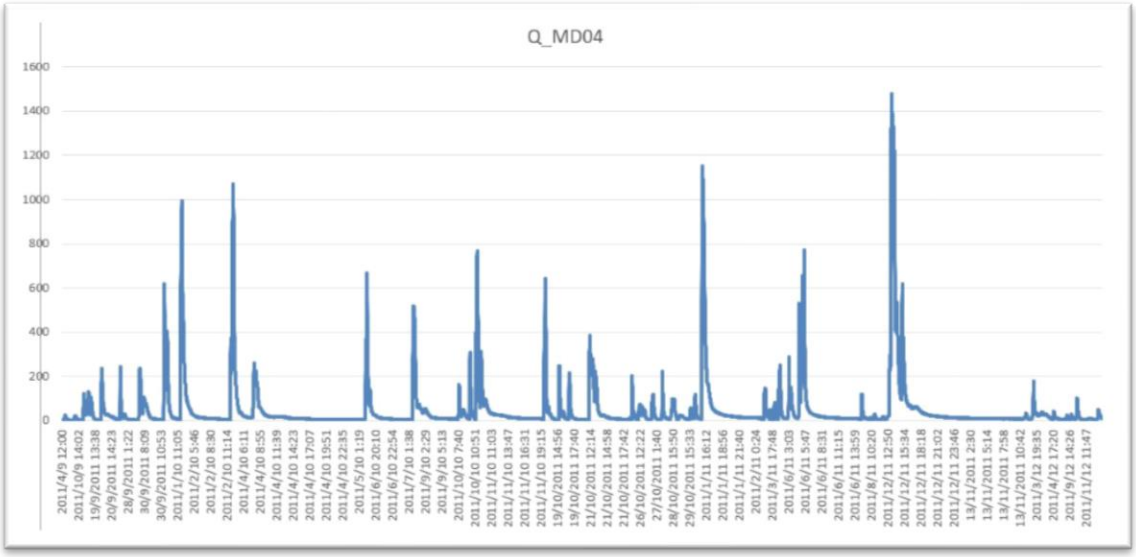

Fig.1 discharge of MD04

2. Data correlation

Determine which factor has the closest relationship with Q_MD04.

Tab.1 correlation with Q_MD04 among different stations

Station	rain	Q_MD01	Q_MD02	Q_CNTRLIB	Q_OPPRLINK	Raincumt
Corelationship	0.76	0.87	0.98	0.82	0.95	0

As we see, Q_MD02 gets highest relationship with Q_MD04 and then Q_OPP, Q_MD01, rain respectively. By considering rainfall is the source of run-offs, it becomes an important factor to be taken.

3. Data events

These data are split by different events. Typically we need to process them based on different events. Filtering them in EXCEL, there are 41 events other than 55 events mentioned in subtitle.

Another thing to be noted is that when shifting data, some events are less than the number which we want to shift. In doing this, we have to delete these events. For example, event 22 has only 7 data, it is impossible to predict the next 10 min or more.

Tab.2 data quantities of each event

events	1	2	7	8	12	13	14	15	16	17
number	99	82	174	217	183	251	162	501	1313	483
events	18	19	20	21	22	23	24	25	26	27

number	451	129	59	662	7	125	315	416	24	62
events	28	29	30	31	32	33	34	35	36	37
number	125	94	175	115	120	604	79	36	43	816
events	38	39	40	61	62	63	64	65	66	67
number	128	61	1403	87	230	135	38	58	56	141
events	75									
number	71									

4. data shift

   In order to predict 10 min, 20 min and 60 min later, we need to shift these data by 10, 20 or 60 based on different events. For example, to predict 10 min later, we shift Q_MD04 by 10 based on events and then to fit it using neural networks. In doing this, we use R to process this replicated work, and some basic codes are in appendix.

5. select dataset

   another significant issue is about selecting training data, cross-validation data and testing data. Generally we choose them by splitting into different portions and shuffle them. It is inappropriate when using time-lag recurrent algorithm because it stores previous sequential data.

   In this case, we are supposed to make training data experience different situations eg. high peaks, intermedia peaks and lower peaks in fig.1 due to events. In doing this, we split data manually by different events.

Neural solutions

Predict 10 min later

1. Algorithm

   there are multiple algorithms inside neural networks such as MLP(multi-layer perceptron), time-lag recurrent, radial basis, self-organizing maps and so on.

   As in hydrologic analysis, we are more concerned about supervised network and in many cases, sequential time data are required to model and do prediction. Based on that, we choose time-lag recurrent to fit our data.

   Time-lag recurrent algorithm can store previous sequential data and value it by adding different weights.

2. Dataset

Tab.3 selection of different datasets

dataset	training	CV	test
event	1	13	2
	7	17	8
	12	21	15
	14	32	19
	16	36	23
	18	39	25
	20	63	27
	24	-	30

	26	-	34
	28	-	38
	29	-	65
	31	-	66
	33	-	-
	35	-	-
	37	-	-
	40	-	-
	61	-	-
	62	-	-
	64	-	-
	67	-	-
	75	-	-
Total number	6351	1692	1920
percentage	64%	17%	19%

3. Neural architecture

we simply set Q_MD04 as input and Q_MD04_shifted10 as output and choose 1 hidden layer with 8 neurons inside.

Tab.4 results of different trails using 1-8-1

trail	MSE(training)	MSE(CV)	R
1	0.010	0.022	0.714
2	0.007	0.019	0.383
3	0.020	0.027	0.637

As we see, even though correlation could achieve 0.7, different trails with same setting contribute to little big difference. That may interpret our model is not stable enough.

Next, improve neural architecture to 2 hidden layers with 8 neurons inside each to testify our model,

Tab.5 results of different trails using 1-8-8-1

trail	MSE(training)	MSE(CV)	R
1	0.007	0.011	0.787
2	0.018	0.028	0.340
3	0.007	0.010	0.868

In this case, r could reach 0.868 which is high enough for prediction while r is still varying greatly according to different trails.

Then, we try to use Rainfall and Q_MD04 as inputs, Q_MD04_shifted10 as output as well. Choose 1 hidden layer and 8 neurons.

Tab. 6 results of different trails using 2-8-1

trail	MSE(training)	MSE(CV)	R
1	0.008	0.010	0.741
2	0.009	0.023	0.674

3	0.007	0.012	0.912

Within this, r is much more stable than before and it even reach as high as 0.912.
Last, we vary hidden layers to 2 and 8 neurons inside.

Tab.7 results of different trails using 2-8-8-1

trail	MSE(training)	MSE(CV)	R
1	0.006	0.007	0.654
2	0.008	0.015	0.718
3	0.010	0.027	0.709

As shown above, this case gets the most stable model than before while r is not as much high as we expect.
By comparing these cases, we finally chose Rainfall, Q_MD04 as input, Q_MD04_shifted10 as output and 2 hidden layers with 8 neurons inside model.

4. outcome

based on the 2-8-1 model, we can roughly make prediction about 10 min later.

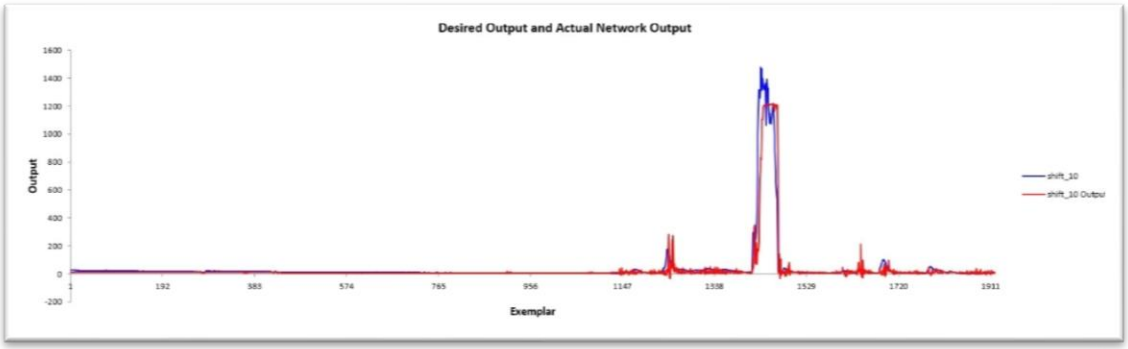

Fig.2 comparison of desired and output testing da

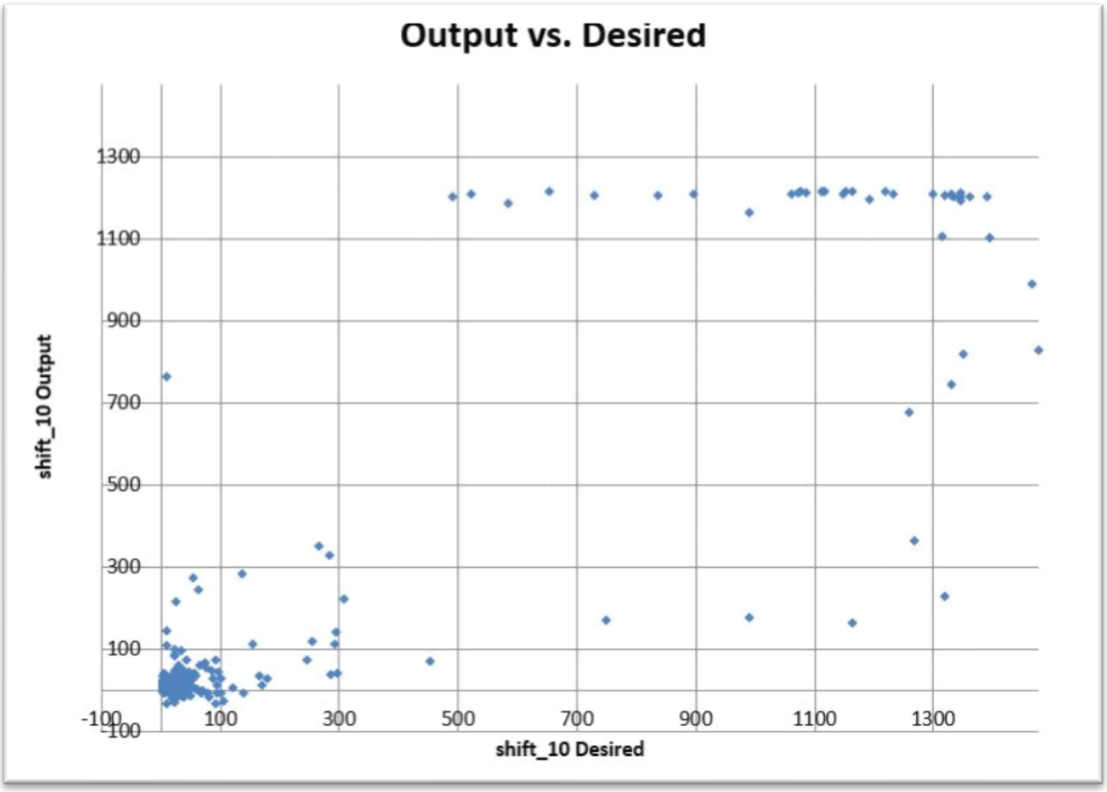

Fig.3 scatter plot of 2-8-1 model

Here shows some prediction data.

Tab.8 part of results of testing data

Rainfall	Q_MD04	Desired	Output
0	9.973	9.634	8.9517924
0	10.141	10.479	8.94347774
0	10.91	10.311	8.93393485
0	10.311	10.694	8.94463858
0	10.141	10.141	9.06518609
0	11.123	10.141	9.08102568
0	9.973	10.91	8.99180474
0	9.973	9.973	9.08316475
0	9.973	10.141	9.01618729
0	9.634	9.973	8.91471825
0	10.479	9.973	8.90084062
0	10.311	9.802	8.86076255
0	10.694	10.141	8.96009636
0	10.141	10.311	9.01526631
0	10.141	9.802	9.06975199

Predict 20 min later

1. algorithm

   algorithm is the same as before.

2. Dataset

Tab.9 selection of different datasets

dataset	training	CV	test
event	1	13	2
	7	17	8
	12	21	15
	14	32	19
	16	36	23
	18	39	25
	20	63	27
	24	-	30
	26	-	34
	28	-	38
	29	-	65
	31	-	66
	33	-	-
	35	-	-
	37	-	-
	40	-	-
	61	-	-
	62	-	-
	64	-	-
	67	-	-
	75	-	-
Total number	6161	1622	1800
percentage	64%	17%	19%

3. Architecture

Based on an optimal model that we chose before, we test is using the same setting.

Tab.10 results of different trails using 2-8-1

trail	MSE(training)	MSE(CV)	R
1	0.010	0.008	0.503
2	0.008	0.004	0.497
3	0.008	0.004	0.292

Tab.11 results of different trails using 2-8-8-1

trail	MSE(training)	MSE(CV)	R
1	0.012	0.024	0.402
2	0.009	0.004	0.485
3	0.014	0.011	0.350

Tab.12 results of different trails using 1-8-1

trail	MSE(training)	MSE(CV)	R

1	0.009	0.005	0.448
2	0.010	0.010	0.302
3	0.010	0.008	0.400

As shown above, we choose 2 inputs and 1 hidden layer which are the same as optimal model finally.

4. outcome

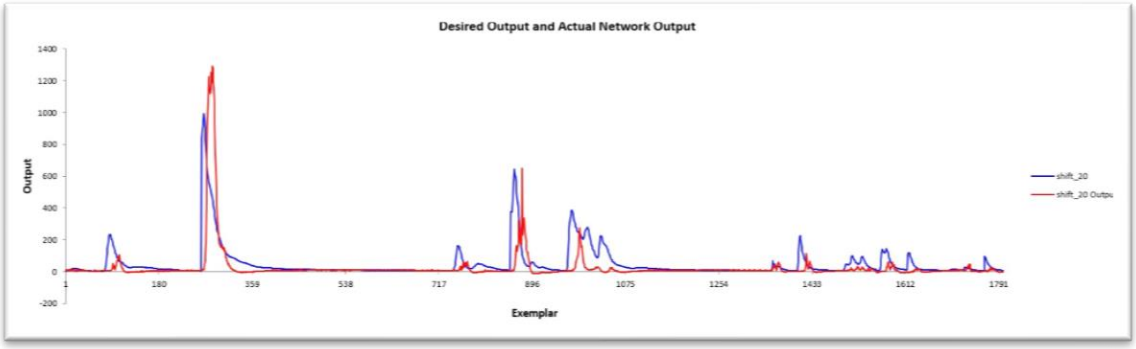

Fig.4 comparison of desired and output testing data

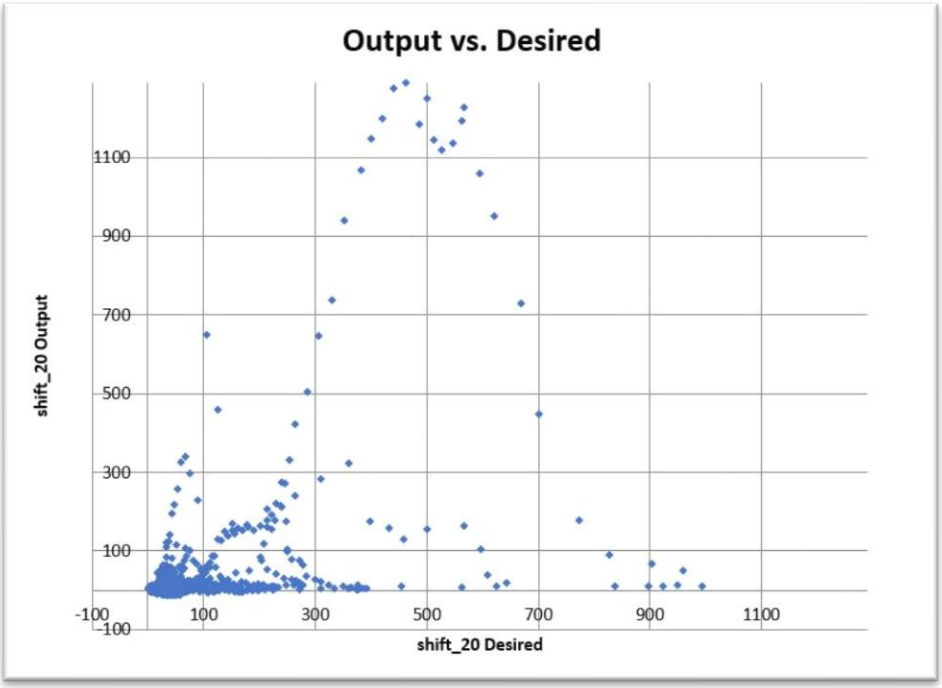

Fig.5 scatter plot of 2-8-1 model

Predict 60 min later

1. algorithm

the same as before

2. dataset

Tab.13 selection of different datasets

dataset	trainning	CV	Testing
events	1	13	2
	7	17	8

	12	21	15
	14	32	19
	16	39	23
	18	63	25
	24		27
	28		30
	29		34
	31		38
	33		
	37		
	40		
	61		
	62		
	67		
	75		
Total number	5380	1358	1347
percentage	67%	17%	16%

3. architecture

select optimal neural architecture firstly as well which is 2 inputs(Rainfall, Q_MD04), 1 output(Q_MD04_shifted60) 1 hidden layer with 8 neurons.

Tab.14 results of different trails using 2-8-1

trail	MSE(training)	MSE(CV)	R
1	0.012	0.001	-0.022
2	0.010	0.010	0.302
3	0.011	0.001	0.294

Tab.15 results of different trails using 2-8-8-1

trail	MSE(training)	MSE(CV)	R
1	0.010	0.003	0.318
2	0.010	0.001	0.017
3	0.011	0.001	0.244

Then choose Q_MD04 as single input and 1 hidden layer with 8 neurons.

Tab.16 results of different trails using 1-8-1

trail	MSE(training)	MSE(CV)	R
1	0.038	0.009	0.115
2	0.012	0.001	-0.057
3	0.013	0.001	0.082

Based on these results, we choose optimal model as well.

4. outcome

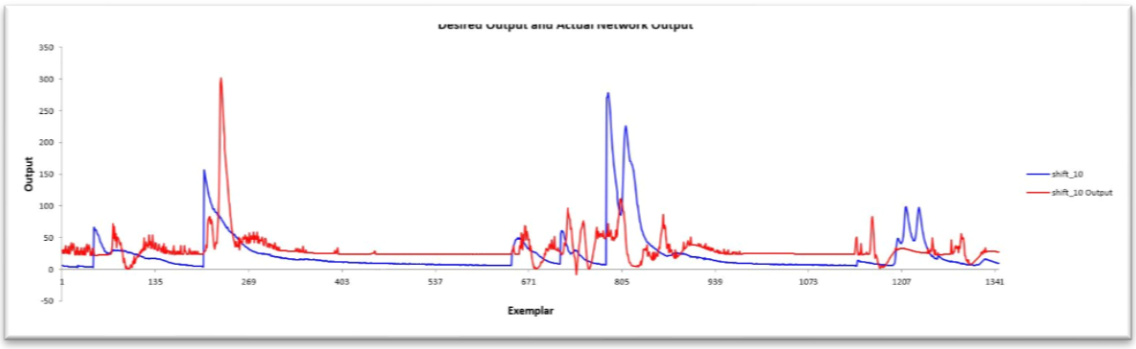

Fig.6 comparison of desired and output testing data

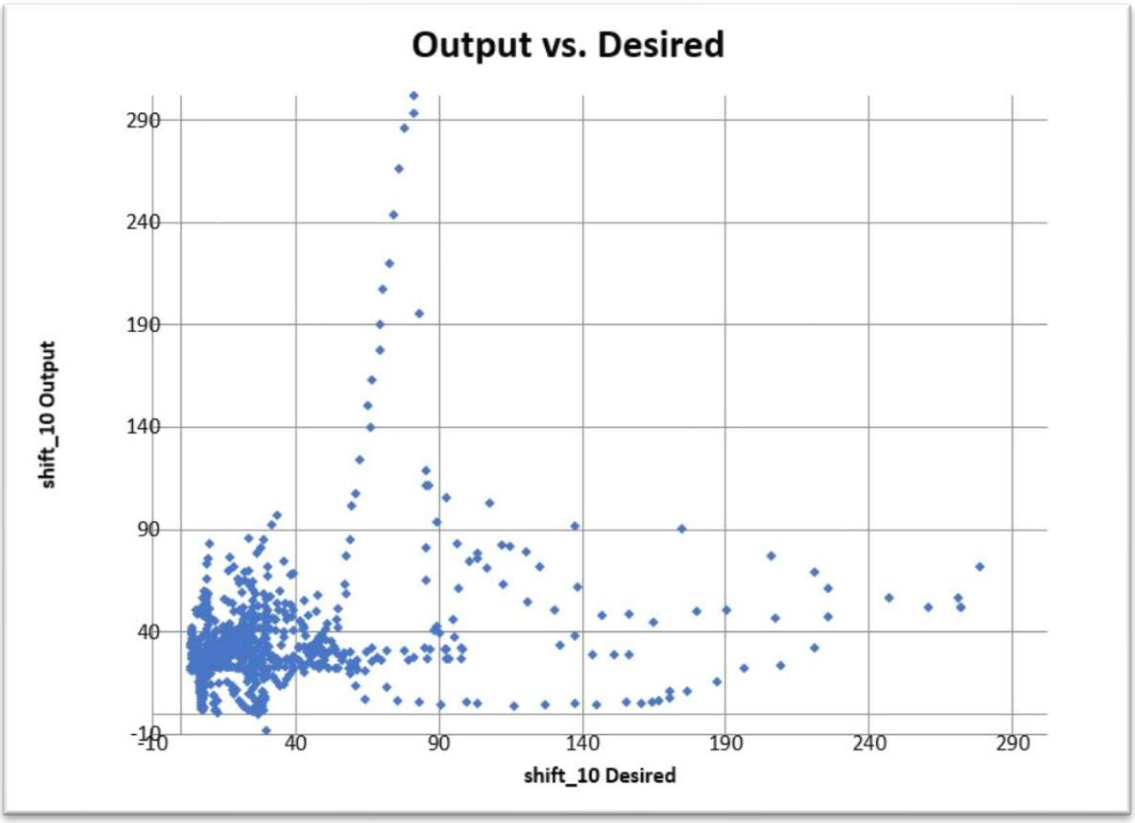

Fig.7 scatter plot of 2-8-1 model

At last, we choose 2 inputs(Rainfall, Q_MD04) 1 output and 1 hidden layer with 8 neurons as our optimal model, activation function as tanh function because it has lager region within interval [-1,1].

Analysis

1. Stability

   We define stability as when we try using the same setting, the result could be slightly different than greatly.

   when predict 10 min later, as we see in Tab.6 results of correlation become reasonably stable. That proves our selection of dataset satisfies this model to some extend.

   When predict 20 min later, because we choose the same model, shown in Tab.10, R varies from 0.5 to 0.2, it starts becoming a bit unstable.

At last, when predict 60 min later, shown in Tab.14, R even becomes negative in some cases. That shows this model is unstable by choosing this combination of data.

2. Accuracy

Generally speaking, forecasting run-offs within short period could be more accurate. Among this process, we can see this trend clearly.

Tab.17 comparison of r using optimal model

prediction	10 min	20 min	60 min
r	0.912	0.503	0.302

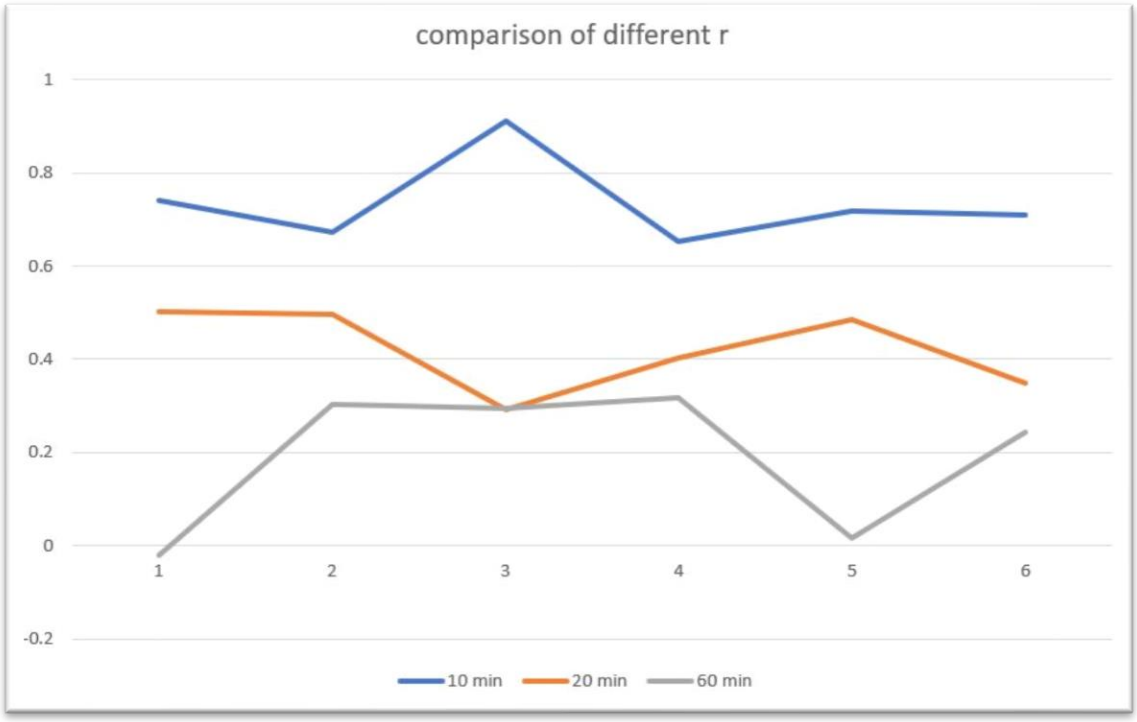

Fig.8 accuracy of different forecast

One reason to this is because the reduce of datasets. As we see in the table below, data is reducing when predict more time later. That is called “Achilles heel” in data science. We desire to get enough data so that it can help us improve accuracy. While it is not always good to get big enough data, some data can deceive us and we are supposed to detect them and spend more time in processing.

Tab.18 total data number in prediction

prediction	10 min	20min	60 min
Total data number	9963	9583	8085
Raw data number	10330		

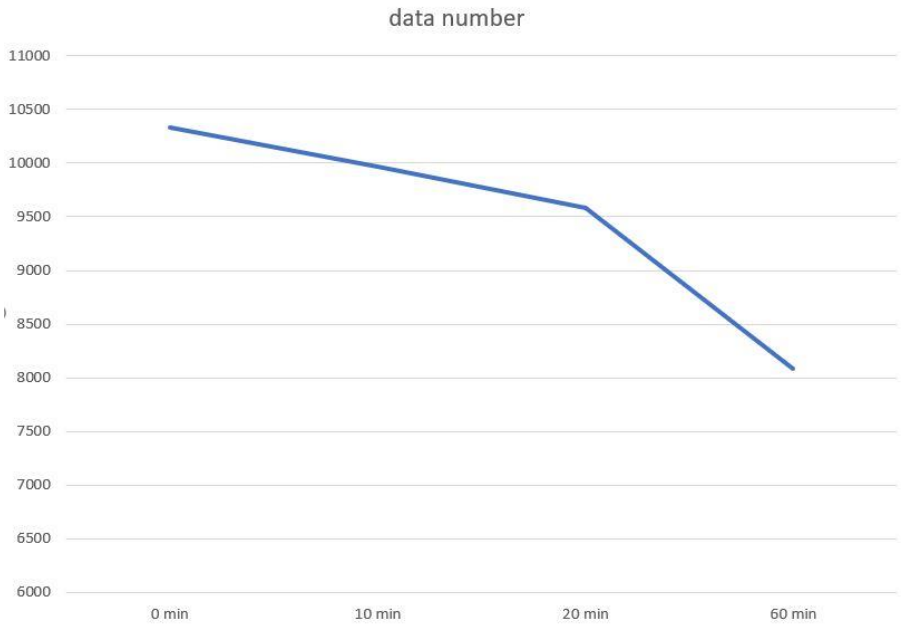

Fig.9 total data number

Another reason may be the selection of dataset. These results are only based on our optimal model which was chosen in prediction 10 min. If we want to get more accurate result in 20 min or 60 min prediction, we may change dataset and use other architecture to fit this.

3. Neural architecture

During this training and testing process, we see neural architecture of 1 hidden layer with 8 neurons is suitable enough to fit our data. If we put more hidden layers, that may result in instability of our model and less accuracy.

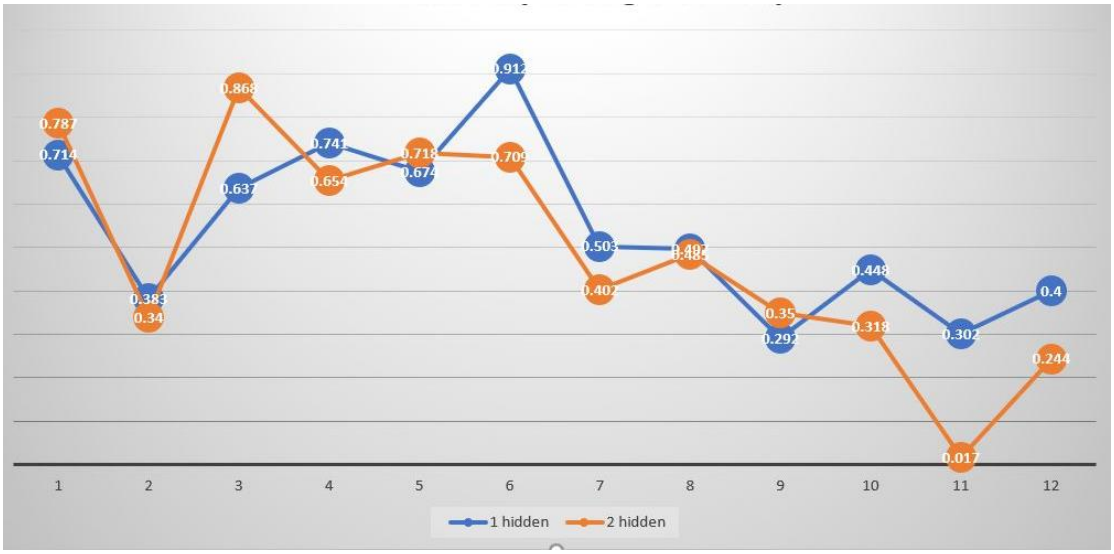

Fig.10 accuracy of different hidden layer

Improvement

If time is allowed, I expect to do such to improve my model.

1. try different combination of input and find out the best way to predict by covering all stations.
2. Using different parameters to understand Time-lag recurrent algorithm.
3. To normalize data and see difference between my model and normalized one.
4. To explain whether outliers matter in this prediction process.
5. To see whether there is a way to detect optimal neural architecture.

Appendix

R codes:

```
# data pre-process
library(xlsx)
my_data<-read.xlsx('KentRidgeRRData-55events.xlsx',2)
# normalize these data
norm<-function(x){
   y<-(x-min(x,na.rm = T))/(max(x,na.rm = T)+min(x,na.rm = T))
   return(y)
}
#-----------------------------------------------------
# shift by 60
MD_04<-subset(my_data,my_data$ev==1)
q<-MD_04$Q_MD04[seq(60,length(MD_04$Q_MD04))]
MD_04<-MD_04[seq(1,length(MD_04$Q_MD04)-59),]
MD_04$shift_10<-q
for(i in seq(1,67)){
   MD<-subset(my_data,my_data$ev==i)
   q<-MD$Q_MD04[seq(60,length(MD$Q_MD04))]
   MD<-MD[seq(1,length(MD$Q_MD04)-59),]
   MD$shift_10<-q
   MD_04<-rbind(MD,MD_04)
}
MD<-subset(my_data,my_data$ev==75)
q<-MD$Q_MD04[seq(60,length(MD$Q_MD04))]
MD<-MD[seq(1,length(MD$Q_MD04)-59),]
MD$shift_10<-q
MD_04<-rbind(MD,MD_04)
write.csv(MD_04,'shift60.csv')
#shift by 20
MD_04<-subset(my_data,my_data$ev==1)
q<-MD_04$Q_MD04[seq(20,length(MD_04$Q_MD04))]
MD_04<-MD_04[seq(1,length(MD_04$Q_MD04)-19),]
MD_04$shift_10<-q
```

```
for(i in seq(1,67)){
   MD<-subset(my_data,my_data$ev==i)
   q<-MD$Q_MD04[seq(20,length(MD$Q_MD04))]
   MD<-MD[seq(1,length(MD$Q_MD04)-19),]
   MD$shift_10<-q
   MD_04<-rbind(MD,MD_04)
}
MD<-subset(my_data,my_data$ev==75)
q<-MD$Q_MD04[seq(20,length(MD$Q_MD04))]
MD<-MD[seq(1,length(MD$Q_MD04)-19),]
MD$shift_10<-q
MD_04<-rbind(MD,MD_04)
write.csv(MD_04,'preprocess_shift20.csv')
#shift by 10
MD_04<-subset(my_data,my_data$ev==1)
q<-MD_04$Q_MD04[seq(10,length(MD_04$Q_MD04))]
MD_04<-MD_04[seq(1,length(MD_04$Q_MD04)-9),]
MD_04$shift_10<-q
for(i in seq(1,67)){
   MD<-subset(my_data,my_data$ev==i)
   q<-MD$Q_MD04[seq(10,length(MD$Q_MD04))]
   MD<-MD[seq(1,length(MD$Q_MD04)-9),]
   MD$shift_10<-q
   MD_04<-rbind(MD,MD_04)
}
```