#####################################################################

**Design Problems Allocation (Cluster technique Reusult**

###############################################################################################

**8 GoF design problems**

###############################################################################################

setwd("D:/Experimentjournal/clusteringwithproblems/Results/GoF\_Result\_8\_Problems")

temp = list.files(pattern="\*.csv")

for (i in 1:length(temp)) assign(temp[i], read.csv(temp[i]))

totalfiles<- length(temp)

for(k in 1: totalfiles)

{

Temp\_Dataset<- read.csv(temp[k],header=TRUE)

Filename <- substr(temp[k],1, regexpr('\_', temp[k])-1)

dumy\_ds <- Temp\_Dataset

Rows <- nrow(dumy\_ds)

for( i in 1:Rows)

{

St <- toString(dumy\_ds[i,2])

Occuranc<-unlist(gregexpr(pattern ='\_',St))

New\_St <- substr(St,Occuranc[length(Occuranc)-1]+1, nchar(St))

DS <- as.matrix(dumy\_ds)

y<-which(dumy\_ds==St)

DS[y]<-toString(New\_St)

dumy\_ds<-data.frame(DS)

}

colnames(dumy\_ds)<- cbind("SNO","FileN","Problem","Actual","Predict")

Uniq <- as.vector(unique(dumy\_ds$FileN))

Len\_uniq <- length(Uniq)

Out <- matrix(NA, nrow= Len\_uniq, ncol=4)

for( i in 1: Len\_uniq)

{

DS <- dumy\_ds[dumy\_ds$FileN == Uniq[i],]

Vector<-ifelse(as.vector(DS$Actual) == as.vector(DS$Predict),1,0)

Total\_Correct\_Prediction <- length(which(Vector ==1))

Out[i,1:1] <- Filename

Out[i,2:2] <- Uniq[i]

Out[i,3:3] <- 8

Out[i,4:4] <- Total\_Correct\_Prediction

}

Filename1 <- paste(Filename,"\_", "GoF\_Problems\_8\_FinalResult" , ".csv", sep="")

write.csv(Out, Filename1)

}

###############################################################################################

**23 GoF design problems**

###############################################################################################

setwd("D:/Experimentjournal/clusteringwithproblems/Results/GoF\_Result\_23\_Problems")

temp = list.files(pattern="\*.csv")

for (i in 1:length(temp)) assign(temp[i], read.csv(temp[i]))

totalfiles<- length(temp)

for(k in 1: totalfiles)

{

Temp\_Dataset<- read.csv(temp[k],header=TRUE)

Filename <- substr(temp[k],1, regexpr('\_', temp[k])-1)

dumy\_ds <- Temp\_Dataset

Rows <- nrow(dumy\_ds)

for( i in 1:Rows)

{

St <- toString(dumy\_ds[i,2])

Occuranc<-unlist(gregexpr(pattern ='\_',St))

New\_St <- substr(St,Occuranc[length(Occuranc)-1]+1, nchar(St))

DS <- as.matrix(dumy\_ds)

y<-which(dumy\_ds==St)

DS[y]<-toString(New\_St)

dumy\_ds<-data.frame(DS)

}

colnames(dumy\_ds)<- cbind("SNO","FileN","Problem","Actual","Predict")

Uniq <- as.vector(unique(dumy\_ds$FileN))

Len\_uniq <- length(Uniq)

Out <- matrix(NA, nrow= Len\_uniq, ncol=4)

for( i in 1: Len\_uniq)

{

DS <- dumy\_ds[dumy\_ds$FileN == Uniq[i],]

Vector<-ifelse(as.vector(DS$Actual) == as.vector(DS$Predict),1,0)

Total\_Correct\_Prediction <- length(which(Vector ==1))

Out[i,1:1] <- Filename

Out[i,2:2] <- Uniq[i]

Out[i,3:3] <- 23

Out[i,4:4] <- Total\_Correct\_Prediction

}

Filename1 <- paste(Filename,"\_", "GoF\_Problems\_23\_FinalResult" , ".csv", sep="")

write.csv(Out, Filename1)

}

#####################################

###############################################################################################

**23 GoF design Examples**

###############################################################################################

setwd("D:/Experimentjournal/clusteringwithproblems/Results/GoF\_Result\_23\_Examples")

temp = list.files(pattern="\*.csv")

for (i in 1:length(temp)) assign(temp[i], read.csv(temp[i]))

totalfiles<- length(temp)

for(k in 1: totalfiles)

{

Temp\_Dataset<- read.csv(temp[k],header=TRUE)

Filename <- substr(temp[k],1, regexpr('\_', temp[k])-1)

dumy\_ds <- Temp\_Dataset

Rows <- nrow(dumy\_ds)

for( i in 1:Rows)

{

St <- toString(dumy\_ds[i,2])

Occuranc<-unlist(gregexpr(pattern ='\_',St))

New\_St <- substr(St,Occuranc[length(Occuranc)-1]+1, nchar(St))

DS <- as.matrix(dumy\_ds)

y<-which(dumy\_ds==St)

DS[y]<-toString(New\_St)

dumy\_ds<-data.frame(DS)

}

colnames(dumy\_ds)<- cbind("SNO","FileN","Problem","Actual","Predict")

Uniq <- as.vector(unique(dumy\_ds$FileN))

Len\_uniq <- length(Uniq)

Out <- matrix(NA, nrow= Len\_uniq, ncol=4)

for( i in 1: Len\_uniq)

{

DS <- dumy\_ds[dumy\_ds$FileN == Uniq[i],]

Vector<-ifelse(as.vector(DS$Actual) == as.vector(DS$Predict),1,0)

Total\_Correct\_Prediction <- length(which(Vector ==1))

Out[i,1:1] <- Filename

Out[i,2:2] <- Uniq[i]

Out[i,3:3] <- 23

Out[i,4:4] <- Total\_Correct\_Prediction

}

Filename1 <- paste(Filename,"\_", "GoF\_Examples\_23\_FinalResult" , ".csv", sep="")

write.csv(Out, Filename1)

}

#####################################

###############################################################################################

**49 Secuity design problems**

###############################################################################################

setwd("D:/Experimentjournal/clusteringwithproblems/Results/Security\_Result\_49\_Problems")

temp = list.files(pattern="\*.csv")

for (i in 1:length(temp)) assign(temp[i], read.csv(temp[i]))

totalfiles<- length(temp)

for(k in 1: totalfiles)

{

Temp\_Dataset<- read.csv(temp[k],header=TRUE)

Filename <- substr(temp[k],1, regexpr('\_', temp[k])-1)

dumy\_ds <- Temp\_Dataset

Rows <- nrow(dumy\_ds)

for( i in 1:Rows)

{

St <- toString(dumy\_ds[i,2])

Occuranc<-unlist(gregexpr(pattern ='\_',St))

New\_St <- substr(St,Occuranc[length(Occuranc)-1]+1, nchar(St))

DS <- as.matrix(dumy\_ds)

y<-which(dumy\_ds==St)

DS[y]<-toString(New\_St)

dumy\_ds<-data.frame(DS)

}

colnames(dumy\_ds)<- cbind("SNO","FileN","Problem","Actual","Predict")

Uniq <- as.vector(unique(dumy\_ds$FileN))

Len\_uniq <- length(Uniq)

Out <- matrix(NA, nrow= Len\_uniq, ncol=4)

for( i in 1: Len\_uniq)

{

DS <- dumy\_ds[dumy\_ds$FileN == Uniq[i],]

Vector<-ifelse(as.vector(DS$Actual) == as.vector(DS$Predict),1,0)

Total\_Correct\_Prediction <- length(which(Vector ==1))

Out[i,1:1] <- Filename

Out[i,2:2] <- Uniq[i]

Out[i,3:3] <- 49

Out[i,4:4] <- Total\_Correct\_Prediction

}

Filename1 <- paste(Filename,"\_", "Security\_Examples\_23\_FinalResult" , ".csv", sep="")

write.csv(Out, Filename1)