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

**Clustering Algorithms**

Dataset( FileName, V1-V23 ( GoF patterns), V1-V34 (Real) and V1-V46(For Security))

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

Create dataset for results of Kmeans clustering

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

1. GOF

###########

setwd("D:/Experimentjournal/FinalDatasets/Datasetswithlabelname/GoF")

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

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

totalfiles<- length(temp)

Out <- matrix(NA, nrow= totalfiles, ncol=28)

for(k in 1:totalfiles)

{

Dataset<- read.csv(temp[k],header=TRUE)

Len <-length(Dataset)

Km1 <- kmeans(Dataset[,2:Len], 3)

# a4 as Vector of Predicted clusters

a1 <- as.character(Km1[1])

a2 <- strsplit(a1, "") ;

a3 <- unlist(a2) ;

a4 <- as.vector(as.numeric(a3))

a4 <- a4[!is.na(a4)]

Filename <- paste(substr(temp[k],1, nchar(temp[k])-4))

# Total Sum of Square

TotalSS <- Km1[3]

# Sum of Square between clusters

BetweenSS <- Km1[6]

# Sum of Square within clusters

WithinSS <- Km1[5]

Out[k,1:1] <- Filename

Out[k,2:2] <- toString(TotalSS)

Out[k,3:3] <- toString(WithinSS)

Out[k,4:4] <- toString(BetweenSS)

Out[k,5:5] <- toString(Km1[4])

Out[k,6:28] <- a4

}

write.csv(Out, ("Kmeans\_GoF.csv"))

RealTime

###########

setwd("D:/Experimentjournal/FinalDatasets/Datasetswithlabelname/RealTime")

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

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

totalfiles<- length(temp)

Out <- matrix(NA, nrow= totalfiles, ncol=39)

for(k in 1:totalfiles)

{

Dataset<- read.csv(temp[k],header=TRUE)

Len <-length(Dataset)

Km1 <- kmeans(Dataset[,2:Len], 5)

# a4 as Vector of Predicted clusters

a1 <- as.character(Km1[1])

a2 <- strsplit(a1, "") ;

a3 <- unlist(a2) ;

a4 <- as.vector(as.numeric(a3))

a4 <- a4[!is.na(a4)]

Filename <- paste(substr(temp[k],1, nchar(temp[k])-4))

# Total Sum of Square

TotalSS <- Km1[3]

# Sum of Square between clusters

BetweenSS <- Km1[6]

# Sum of Square within clusters

WithinSS <- Km1[5]

Out[k,1:1] <- Filename

Out[k,2:2] <- toString(TotalSS)

Out[k,3:3] <- toString(WithinSS)

Out[k,4:4] <- toString(BetweenSS)

Out[k,5:5] <- toString(Km1[4])

Out[k,6:39] <- a4

}

write.csv(Out, ("Kmeans\_Realtime.csv"))

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

Security

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

setwd("D:/Experimentjournal/FinalDatasets/Datasetswithlabelname/Security")

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

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

totalfiles<- length(temp)

Out <- matrix(NA, nrow= totalfiles, ncol=51)

for(k in 1:totalfiles)

{

Dataset<- read.csv(temp[k],header=TRUE)

Len <-length(Dataset)

Km1 <- kmeans(Dataset[,2:Len], 8)

# a4 as Vector of Predicted clusters

a1 <- as.character(Km1[1])

a2 <- strsplit(a1, "") ;

a3 <- unlist(a2) ;

a4 <- as.vector(as.numeric(a3))

a4 <- a4[!is.na(a4)]

Filename <- paste(substr(temp[k],1, nchar(temp[k])-4))

# Total Sum of Square

TotalSS <- Km1[3]

# Sum of Square between clusters

BetweenSS <- Km1[6]

# Sum of Square within clusters

WithinSS <- Km1[5]

Out[k,1:1] <- Filename

Out[k,2:2] <- toString(TotalSS)

Out[k,3:3] <- toString(WithinSS)

Out[k,4:4] <- toString(BetweenSS)

Out[k,5:5] <- toString(Km1[4])

Out[k,6:51] <- a4

}

write.csv(Out, ("Kmeans\_Security.csv"))

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

Create dataset for results of Hirarchical clustering (Eucladian distance)

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

GoF

#########

setwd("D:/Experimentjournal/FinalDatasets/Datasetswithlabelname/GoF")

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

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

totalfiles<- length(temp)

Out <- matrix(NA, nrow= totalfiles, ncol=24)

for(k in 1:totalfiles)

{

Dataset<- read.csv(temp[k],header=TRUE)

Len <-length(Dataset)

distance <- dist(Dataset[,2:Len], method="euclidean")

cluster <- hclust(distance, method="average")

Filename <- paste(substr(temp[k],1, nchar(temp[k])-4))

group.3 <- cutree(cluster, k = 3)

Out[k,1:1] <- Filename

Out[k,2:24] <- group.3

}

write.csv(Out, ("Hirarchical\_Euclidian\_GoF.csv"))

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

RealTime

#########

setwd("D:/Experimentjournal/FinalDatasets/Datasetswithlabelname/RealTime")

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

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

totalfiles<- length(temp)

Out <- matrix(NA, nrow= totalfiles, ncol=35)

for(k in 1:totalfiles)

{

Dataset<- read.csv(temp[k],header=TRUE)

Len <-length(Dataset)

distance <- dist(Dataset[,2:Len], method="euclidean")

cluster <- hclust(distance, method="average")

Filename <- paste(substr(temp[k],1, nchar(temp[k])-4))

group.3 <- cutree(cluster, k = 5)

Out[k,1:1] <- Filename

Out[k,2:35] <- group.3

}

write.csv(Out, ("Hirarchical\_Euclidian\_RealTime.csv"))

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

Security

#########

setwd("D:/Experimentjournal/FinalDatasets/Datasetswithlabelname/Security")

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

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

totalfiles<- length(temp)

Out <- matrix(NA, nrow= totalfiles, ncol=47)

for(k in 1:totalfiles)

{

Dataset<- read.csv(temp[k],header=TRUE)

Len <-length(Dataset)

distance <- dist(Dataset[,2:Len], method="euclidean")

cluster <- hclust(distance, method="average")

Filename <- paste(substr(temp[k],1, nchar(temp[k])-4))

group.3 <- cutree(cluster, k = 8)

Out[k,1:1] <- Filename

Out[k,2:47] <- group.3

}

write.csv(Out, ("Hirarchical\_Euclidian\_Security.csv"))

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

Create dataset for results of Hirarchical clustering (Eucladian distance)

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

GoF

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

setwd("D:/Experimentjournal/FinalDatasets/Datasetswithlabelname/GoF")

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

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

totalfiles<- length(temp)

Out <- matrix(NA, nrow= totalfiles, ncol=24)

for(k in 1:totalfiles)

{

Dataset<- read.csv(temp[k],header=TRUE)

Len <-length(Dataset)

distance <- dist(Dataset[,2:Len], method="manhattan")

cluster <- hclust(distance, method="average")

Filename <- paste(substr(temp[k],1, nchar(temp[k])-4))

group.3 <- cutree(cluster, k = 3)

Out[k,1:1] <- Filename

Out[k,2:24] <- group.3

}

write.csv(Out, ("Hirarchical\_manhattan\_GoF.csv"))

#######

RealTime

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

setwd("D:/Experimentjournal/FinalDatasets/Datasetswithlabelname/RealTime")

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

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

totalfiles<- length(temp)

Out <- matrix(NA, nrow= totalfiles, ncol=35)

for(k in 1:totalfiles)

{

Dataset<- read.csv(temp[k],header=TRUE)

Len <-length(Dataset)

distance <- dist(Dataset[,2:Len], method="manhattan")

cluster <- hclust(distance, method="average")

Filename <- paste(substr(temp[k],1, nchar(temp[k])-4))

group.3 <- cutree(cluster, k = 5)

Out[k,1:1] <- Filename

Out[k,2:35] <- group.3

}

write.csv(Out, ("Hirarchical\_manhattan\_RealTime.csv"))

#######

Security

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

setwd("D:/Experimentjournal/FinalDatasets/Datasetswithlabelname/Security")

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

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

totalfiles<- length(temp)

Out <- matrix(NA, nrow= totalfiles, ncol=47)

for(k in 1:totalfiles)

{

Dataset<- read.csv(temp[k],header=TRUE)

Len <-length(Dataset)

distance <- dist(Dataset[,2:Len], method="manhattan")

cluster <- hclust(distance, method="average")

Filename <- paste(substr(temp[k],1, nchar(temp[k])-4))

group.3 <- cutree(cluster, k = 8)

Out[k,1:1] <- Filename

Out[k,2:47] <- group.3

}

write.csv(Out, ("Hirarchical\_manhattan\_security.csv"))

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

Create dataset for results of Fuzzy Cmeans clustering

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

GoF

#########3

setwd("D:/Experimentjournal/FinalDatasets/Datasetswithlabelname/GoF")

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

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

totalfiles<- length(temp)

Out <- matrix(NA, nrow= totalfiles, ncol=24)

for(k in 1:totalfiles)

{

Dataset<- read.csv(temp[k],header=TRUE)

Len <-length(Dataset)

result <- cmeans(Dataset[,2:Len], centers=3, iter.max=100, m=2, method="cmeans")

Filename <- paste(substr(temp[k],1, nchar(temp[k])-4))

a1 <- as.character(result[3])

a2 <- strsplit(a1, "") ;

a3 <- unlist(a2) ;

a4 <- as.vector(as.numeric(a3))

a4 <- a4[!is.na(a4)]

Out[k,1:1] <- Filename

Out[k,2:24] <- a4

}

write.csv(Out, ("cmeans\_GoF.csv"))

##########

RealTime

#########3

setwd("D:/Experimentjournal/FinalDatasets/Datasetswithlabelname/RealTime")

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

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

totalfiles<- length(temp)

Out <- matrix(NA, nrow= totalfiles, ncol=35)

for(k in 1:totalfiles)

{

Dataset<- read.csv(temp[k],header=TRUE)

Len <-length(Dataset)

result <- cmeans(Dataset[,2:Len], centers=5, iter.max=100, m=2, method="cmeans")

Filename <- paste(substr(temp[k],1, nchar(temp[k])-4))

a1 <- as.character(result[3])

a2 <- strsplit(a1, "") ;

a3 <- unlist(a2) ;

a4 <- as.vector(as.numeric(a3))

a4 <- a4[!is.na(a4)]

Out[k,1:1] <- Filename

Out[k,2:35] <- a4

}

write.csv(Out, ("cmeans\_RealTime.csv"))

##########

Security

#########3

setwd("D:/Experimentjournal/FinalDatasets/Datasetswithlabelname/Security")

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

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

totalfiles<- length(temp)

Out <- matrix(NA, nrow= totalfiles, ncol=47)

for(k in 1:totalfiles)

{

Dataset<- read.csv(temp[k],header=TRUE)

Len <-length(Dataset)

result <- cmeans(Dataset[,2:Len], centers=8, iter.max=100, m=2, method="cmeans")

Filename <- paste(substr(temp[k],1, nchar(temp[k])-4))

a1 <- as.character(result[3])

a2 <- strsplit(a1, "") ;

a3 <- unlist(a2) ;

a4 <- as.vector(as.numeric(a3))

a4 <- a4[!is.na(a4)]

Out[k,1:1] <- Filename

Out[k,2:47] <- a4

}

write.csv(Out, ("cmeans\_security.csv"))

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

Create dataset for results of PAM clustering (euclidenc)

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

GoF

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

setwd("D:/Experimentjournal/FinalDatasets/Datasetswithlabelname/GoF")

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

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

totalfiles<- length(temp)

Out <- matrix(NA, nrow= totalfiles, ncol=24)

for(k in 1:totalfiles)

{

Dataset<- read.csv(temp[k],header=TRUE)

Len <-length(Dataset)

result <- pam(Dataset[,2:Len], 3, FALSE, "euclidean")

Filename <- paste(substr(temp[k],1, nchar(temp[k])-4))

a1 <- as.character(result[3])

a2 <- strsplit(a1, "") ;

a3 <- unlist(a2) ;

a4 <- as.vector(as.numeric(a3))

a4 <- a4[!is.na(a4)]

Out[k,1:1] <- Filename

Out[k,2:24] <- a4

}

write.csv(Out, ("PAM\_Euclidian\_GoF.csv"))

########

RealTime

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

setwd("D:/Experimentjournal/FinalDatasets/Datasetswithlabelname/RealTime")

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

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

totalfiles<- length(temp)

Out <- matrix(NA, nrow= totalfiles, ncol=35)

for(k in 1:totalfiles)

{

Dataset<- read.csv(temp[k],header=TRUE)

Len <-length(Dataset)

result <- pam(Dataset[,2:Len], 5, FALSE, "euclidean")

Filename <- paste(substr(temp[k],1, nchar(temp[k])-4))

a1 <- as.character(result[3])

a2 <- strsplit(a1, "") ;

a3 <- unlist(a2) ;

a4 <- as.vector(as.numeric(a3))

a4 <- a4[!is.na(a4)]

Out[k,1:1] <- Filename

Out[k,2:35] <- a4

}

write.csv(Out, ("PAM\_Euclidian\_RealTime.csv"))

########

Security

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

setwd("D:/Experimentjournal/FinalDatasets/Datasetswithlabelname/Security")

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

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

totalfiles<- length(temp)

Out <- matrix(NA, nrow= totalfiles, ncol=47)

for(k in 1:totalfiles)

{

Dataset<- read.csv(temp[k],header=TRUE)

Len <-length(Dataset)

result <- pam(Dataset[,2:Len], 8, FALSE, "euclidean")

Filename <- paste(substr(temp[k],1, nchar(temp[k])-4))

a1 <- as.character(result[3])

a2 <- strsplit(a1, "") ;

a3 <- unlist(a2) ;

a4 <- as.vector(as.numeric(a3))

a4 <- a4[!is.na(a4)]

Out[k,1:1] <- Filename

Out[k,2:47] <- a4

}

write.csv(Out, ("PAM\_Euclidiansecurity.csv"))

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

Create dataset for results of PAM clustering (manhattan)

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

GoF

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

setwd("D:/Experimentjournal/FinalDatasets/Datasetswithlabelname/GoF")

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

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

totalfiles<- length(temp)

Out <- matrix(NA, nrow= totalfiles, ncol=24)

for(k in 1:totalfiles)

{

Dataset<- read.csv(temp[k],header=TRUE)

Len <-length(Dataset)

result <- pam(Dataset[,2:Len], 3, FALSE, "manhattan")

Filename <- paste(substr(temp[k],1, nchar(temp[k])-4))

a1 <- as.character(result[3])

a2 <- strsplit(a1, "") ;

a3 <- unlist(a2) ;

a4 <- as.vector(as.numeric(a3))

a4 <- a4[!is.na(a4)]

Out[k,1:1] <- Filename

Out[k,2:24] <- a4

}

write.csv(Out, ("PAM\_Manhattan\_GoF.csv"))

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

RealTime

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

setwd("D:/Experimentjournal/FinalDatasets/Datasetswithlabelname/RealTime")

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

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

totalfiles<- length(temp)

Out <- matrix(NA, nrow= totalfiles, ncol=35)

for(k in 1:totalfiles)

{

Dataset<- read.csv(temp[k],header=TRUE)

Len <-length(Dataset)

result <- pam(Dataset[,2:Len], 5, FALSE, "manhattan")

Filename <- paste(substr(temp[k],1, nchar(temp[k])-4))

a1 <- as.character(result[3])

a2 <- strsplit(a1, "") ;

a3 <- unlist(a2) ;

a4 <- as.vector(as.numeric(a3))

a4 <- a4[!is.na(a4)]

Out[k,1:1] <- Filename

Out[k,2:35] <- a4

}

write.csv(Out, ("PAM\_Manhattan\_RealTime.csv"))

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

Security

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

setwd("D:/Experimentjournal/FinalDatasets/Datasetswithlabelname/Security")

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

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

totalfiles<- length(temp)

Out <- matrix(NA, nrow= totalfiles, ncol=47)

for(k in 1:totalfiles)

{

Dataset<- read.csv(temp[k],header=TRUE)

Len <-length(Dataset)

result <- pam(Dataset[,2:Len], 8, FALSE, "manhattan")

Filename <- paste(substr(temp[k],1, nchar(temp[k])-4))

a1 <- as.character(result[3])

a2 <- strsplit(a1, "") ;

a3 <- unlist(a2) ;

a4 <- as.vector(as.numeric(a3))

a4 <- a4[!is.na(a4)]

Out[k,1:1] <- Filename

Out[k,2:47] <- a4

}

write.csv(Out, ("PAM\_Manhattan\_Securitye.csv"))