Final Paper Results

library(tidyverse, quietly = TRUE)

## ── Attaching packages ──────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 3.0.0 ✔ purrr 0.2.5  
## ✔ tibble 2.0.1 ✔ dplyr 0.7.8  
## ✔ tidyr 0.8.2 ✔ stringr 1.3.1  
## ✔ readr 1.3.1 ✔ forcats 0.3.0

## Warning: package 'tibble' was built under R version 3.5.2

## ── Conflicts ─────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(purrr, quietly = TRUE)  
library(stringr, quietly = TRUE)  
library(furrr, quietly = TRUE)

## Warning: package 'future' was built under R version 3.5.2

library(lubridate, quietly = TRUE)

##   
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':  
##   
## date

library(broom, quietly = TRUE)  
library(knitr, quietly = TRUE)  
library(DAAG, quietly = TRUE)  
library(ROSE, quietly = TRUE)

## Loaded ROSE 0.0-3

library(caret, quietly = TRUE)

##   
## Attaching package: 'caret'

## The following object is masked from 'package:future':  
##   
## cluster

## The following object is masked from 'package:purrr':  
##   
## lift

library(e1071, quietly = TRUE)  
library(readxl, quietly = TRUE)

This document summarises the important results from the IBI analysis.

# Part 1: Time Domain

## Data Set

Note: In this section, I couldn’t reproduce the same numbers for table() as in the code that was sent

mydat = read.csv("../DataOutputs/data\_out.csv")%>%  
 filter(threshold == 250 & winds == 2)%>%  
 select(-winds, - threshold, -index)  
dim(mydat)

## [1] 2153 15

mydat1 <- mydat[ complete.cases(mydat) , ]  
rmind <- which((mydat1$Stress>=5) & (mydat1$Y == "Control"))  
mydat2 <- mydat1[-rmind, ]  
dim(mydat2)

## [1] 129 15

#142  
str(mydat2)

## 'data.frame': 129 obs. of 15 variables:  
## $ ID : int 201 201 201 201 201 202 202 207 207 207 ...  
## $ when : Factor w/ 2115 levels "2016-12-22 23:29:35",..: 65 74 110 114 146 302 314 417 418 428 ...  
## $ Event : Factor w/ 2116 levels "2016-12-22 23:29:35",..: 65 74 110 114 146 302 313 417 418 428 ...  
## $ Stress : int 2 1 3 2 2 2 1 1 1 4 ...  
## $ Y : Factor w/ 2 levels "Control","Episode": 1 1 2 1 2 1 2 1 1 2 ...  
## $ SDNN : num 56.7 27.5 47.8 55.2 47 ...  
## $ SDANN : num 30.82 9.6 13.64 5.58 19.47 ...  
## $ SDNNIDX: num 45.9 29 40.1 46.8 43.6 ...  
## $ pNN50 : num 19.4 11.4 12.1 19.3 13.6 ...  
## $ SDSD : num 48.8 38 38 55.9 39 ...  
## $ rMSSD : num 48.8 38 38 55.9 39 ...  
## $ IRRR : num 78.1 31.3 62.5 78.1 62.5 ...  
## $ MADRR : num 31.3 15.6 15.6 31.3 15.6 ...  
## $ TINN : num 143 63.8 109 134.5 109.1 ...  
## $ HRVi : num 9.15 4.09 6.98 8.61 6.98 ...

table(mydat2$Y)

##   
## Control Episode   
## 110 19

# Control Episode   
# 0 110 19   
  
#creat within subject z scores  
mydat3 <- mydat2  
temp = unique(mydat2$ID)  
m = length(temp)  
for (i in 1:m){  
tempdat <- mydat2[which(mydat2$ID == temp[i]), ]   
mydat3[which(mydat2$ID == temp[i]), 6:15] <- scale(tempdat[ , 6:15])   
}  
mydat3 <- mydat3[complete.cases(mydat3), ]  
table(mydat3$Y)

##   
## Control Episode   
## 107 19

# Control Episode   
# 0 107 19

## Summary Statistics

The statistics for the controls:

mydat3 %>%  
 filter(Y == 'Control') %>%  
 select(-c(Y, when, Event, Stress)) %>%  
 psych::describe(quant=c(.25,.75))%>%  
 rownames\_to\_column()%>%  
 as\_tibble()%>%  
 rename(Variable = rowname)%>%  
 filter(Variable != "ID") %>%  
 select(one\_of(c("Variable", "n", "mean", "sd"))) %>%  
 kable()

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | n | mean | sd |
| SDNN | 107 | -0.0199529 | 0.9760171 |
| SDANN | 107 | 0.0083916 | 0.9483852 |
| SDNNIDX | 107 | -0.0469653 | 0.9328950 |
| pNN50 | 107 | -0.0003382 | 0.9542373 |
| SDSD | 107 | 0.0115435 | 0.9598725 |
| rMSSD | 107 | 0.0115467 | 0.9598776 |
| IRRR | 107 | -0.0376267 | 0.9463951 |
| MADRR | 107 | 0.0017730 | 0.9635628 |
| TINN | 107 | -0.0432215 | 0.9493561 |
| HRVi | 107 | -0.0432215 | 0.9493561 |

And for episodes:

mydat3 %>%  
 filter(Y != 'Control') %>%  
 select(-c(Y, when, Event, Stress)) %>%  
 psych::describe(quant=c(.25,.75))%>%  
 rownames\_to\_column()%>%  
 as\_tibble()%>%  
 rename(Variable = rowname)%>%  
 filter(Variable != "ID") %>%  
 select(one\_of(c("Variable", "n", "mean", "sd")))%>%  
 kable()

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | n | mean | sd |
| SDNN | 19 | 0.1123663 | 0.7356372 |
| SDANN | 19 | -0.0472578 | 0.9312470 |
| SDNNIDX | 19 | 0.2644888 | 0.9770645 |
| pNN50 | 19 | 0.0019047 | 0.8968964 |
| SDSD | 19 | -0.0650084 | 0.8577036 |
| rMSSD | 19 | -0.0650263 | 0.8576682 |
| IRRR | 19 | 0.2118977 | 0.9145446 |
| MADRR | 19 | -0.0099848 | 0.8360532 |
| TINN | 19 | 0.2434052 | 0.8862870 |
| HRVi | 19 | 0.2434052 | 0.8862870 |

## Between Groups Test

None of the features are significantly different using the permutation T test.

mydat3 %>%  
 select(Y:HRVi)%>%  
 summarise\_at(funs(twotPermutation(.[Y == "Control"],  
 .[Y != "Control"],   
 plotit = F)),  
 .vars = vars(SDNN:HRVi))%>%  
 kable()

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SDNN | SDANN | SDNNIDX | pNN50 | SDSD | rMSSD | IRRR | MADRR | TINN | HRVi |
| 0.588 | 0.804 | 0.202 | 0.99 | 0.75 | 0.744 | 0.274 | 0.958 | 0.218 | 0.231 |

## Machine Learning

# library(e1071)  
# # make predictions  
# library(ROSE)  
# library(caret)  
modelfit <- function(data, mytest){  
  
 # mytrain <- dataset[-folder, ]  
 # mytest <- dataset[folder, ]   
 ##Generate balanced dataset  
 train.m.bal<- ovun.sample(Y ~ ., data=data, method="both",p=0.6, seed=1342)$data  
 #names(train.m.bal)  
 #table(train.m.bal$Y)  
 x.train <- train.m.bal[, 6:15]  
 x.train <- as.matrix(x.train)  
 y.train <- train.m.bal$Y  
 #y.train <- as.numeric(train.m.bal$Y)  
 #y.train[which(train.m.bal$Y == "Control")] <- 0  
 #y.train[which(train.m.bal$Y == "Episode")] <- 1  
 #table(y.train) # 0 is control, 1 is episode  
 data.train <- cbind(y.train, x.train)  
 fit<- train(as.factor(y.train) ~ ., data= data.train,method="svmPoly")  
 x.test <- mytest[, 6:15]  
 x.test <- as.matrix(x.test)  
 ytest <- as.factor(as.numeric(mytest$Y))  
 #table(ytest)  
 yhat6 = predict(fit, x.test)  
 temp <- confusionMatrix(yhat6,ytest)  
   
 result <- c(temp$overall[1], temp$byClass[1:2]) #<-can change threshold if you want  
 return(result)  
}  
  
#make predictions  
#involve random sampling, need to set the seed  
#set.seed(9)  
#nfolds=4  
#subdata<-createFolds(mydat2$Y, nfolds)  
#t1 <- modelfit(mydat2, subdata$Fold1)  
#t1  
#t2 <- modelfit(mydat2, subdata$Fold2)  
#t2  
#t3 <- modelfit(mydat2, subdata$Fold3)  
#t3  
#t4 <- modelfit(mydat2, subdata$Fold4)  
#t4  
#colMeans(rbind(t1,t2,t3,t4))  
  
set.seed(9)  
nfolds=4  
subdata<-createFolds(mydat3$Y, nfolds)  
t1 <- modelfit(mydat3[-subdata$Fold1,], mydat3[subdata$Fold1,])  
#t1  
t2 <- modelfit(mydat3[-subdata$Fold2,], mydat3[subdata$Fold2,])  
#t2  
t3 <- modelfit(mydat3[-subdata$Fold3,], mydat3[subdata$Fold3,])  
#t3  
t4 <- modelfit(mydat3[-subdata$Fold4,], mydat3[subdata$Fold4,])  
#t4  
#colMeans(rbind(t1,t2,t3,t4))

Here are the mean accuracy scores for the machine learning approach

colMeans(rbind(t1,t2,t3,t4)) %>%  
 kable()

|  |  |
| --- | --- |
|  | x |
| Accuracy | 0.6348286 |
| Sensitivity | 0.6267806 |
| Specificity | 0.7000000 |

# Part 2: Frequency Domain

## Data Set

Note: I couldn’t reproduce the same numbers for table() as in the code that was sent

mydat <- read.csv("../DataOutputs/data\_out\_20181113.csv")%>%  
 filter(threshold == 100 & winds == 2)%>%  
 select(-winds, - threshold, -index)  
  
ind <- c(1, 4, 5, 16:37) # Why did we use these indeces?  
mydat\_freq <- mydat[, ind]  
#mydat2 <- mydat[ complete.cases(mydat$SDNN) , ]  
  
mydat1 <- mydat\_freq[ complete.cases(mydat\_freq) , ]  
rmind <- which((mydat1$Stress>=5) & (mydat1$Y == "Control"))  
mydat2 <- mydat1[-rmind, ]  
dim(mydat2)

## [1] 101 25

#142  
str(mydat2)

## 'data.frame': 101 obs. of 25 variables:  
## $ ID : int 201 201 201 201 201 201 201 202 202 202 ...  
## $ Stress : int 2 1 2 2 2 2 2 3 2 1 ...  
## $ Y : Factor w/ 2 levels "Control","Episode": 1 1 1 1 1 2 2 1 1 2 ...  
## $ Avg\_niHR : num 80.7 104.8 78.2 74.7 74.8 ...  
## $ Start\_niHR: num 83.5 120 83.5 61.9 69.8 ...  
## $ End\_niHR : num 93.7 103.8 80 75.3 76.8 ...  
## $ Avg\_HR : num 80.1 105 78.3 74.1 74.2 ...  
## $ HF\_1 : num 198 137 219 376 369 ...  
## $ HF\_2 : num 284.2 49.1 207 264 252.8 ...  
## $ HF\_3 : num 216.3 36.8 212.7 250.3 256.8 ...  
## $ HF\_4 : num 247.2 23.4 306.7 283.5 281.2 ...  
## $ HF\_5 : num 278.1 70.9 176.6 300.5 301.3 ...  
## $ HF\_6 : num 251 95 153 198 197 ...  
## $ LF\_1 : num 730 575 1072 605 564 ...  
## $ LF\_2 : num 1200 196 2966 165 170 ...  
## $ LF\_3 : num 249 144 805 135 137 ...  
## $ LF\_4 : num 928.5 76.1 601.8 714.3 705.2 ...  
## $ LF\_5 : num 860 343 997 453 453 ...  
## $ LF\_6 : num 1563 390 647 697 698 ...  
## $ LFHF\_1 : num 3.64 5.53 4.57 1.51 1.42 ...  
## $ LFHF\_2 : num 4.392 6.609 14.359 0.641 0.687 ...  
## $ LFHF\_3 : num 1.273 4.335 4.561 0.545 0.537 ...  
## $ LFHF\_4 : num 3.69 2.79 1.87 2.57 2.56 ...  
## $ LFHF\_5 : num 3.07 5.05 6.97 1.52 1.51 ...  
## $ LFHF\_6 : num 5.38 5.92 4.17 3.33 3.35 ...

table(mydat2$Y)

##   
## Control Episode   
## 86 15

# Control Episode   
# 0 110 19   
  
#creat within subject z scores  
mydat3 <- mydat2  
temp = unique(mydat2$ID)  
m = length(temp)  
for (i in 1:m){  
 tempdat <- mydat2[which(mydat2$ID == temp[i]), ]   
 mydat3[which(mydat2$ID == temp[i]), 4:25] <- scale(tempdat[ , 4:25])   
}  
mydat3 <- mydat3[complete.cases(mydat3), ]  
table(mydat3$Y)

##   
## Control Episode   
## 80 15

# Control Episode   
# 0 107 19

## Summary Statistics

The statistics for the controls:

mydat3 %>%  
 filter(Y == 'Control') %>%  
 psych::describe(quant=c(.25,.75))%>%  
 rownames\_to\_column()%>%  
 as\_tibble()%>%  
 rename(Variable = rowname)%>%  
 filter(Variable != "ID") %>%  
 select(one\_of(c("Variable", "n", "mean", "sd"))) %>%  
 kable()

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | n | mean | sd |
| Stress | 80 | 2.2875000 | 0.9027910 |
| Y\* | 80 | 1.0000000 | 0.0000000 |
| Avg\_niHR | 80 | 0.0154878 | 0.9761098 |
| Start\_niHR | 80 | -0.0590838 | 0.9560958 |
| End\_niHR | 80 | -0.0507360 | 0.9622082 |
| Avg\_HR | 80 | 0.0124995 | 0.9768782 |
| HF\_1 | 80 | 0.0820805 | 0.9629468 |
| HF\_2 | 80 | 0.0463964 | 0.9628535 |
| HF\_3 | 80 | -0.0512680 | 0.8700588 |
| HF\_4 | 80 | 0.0263691 | 0.8593367 |
| HF\_5 | 80 | 0.0276100 | 0.9305916 |
| HF\_6 | 80 | -0.0390963 | 0.9416274 |
| LF\_1 | 80 | 0.0573686 | 0.9875923 |
| LF\_2 | 80 | 0.0687041 | 0.9995445 |
| LF\_3 | 80 | 0.0052999 | 0.8982046 |
| LF\_4 | 80 | 0.0139359 | 0.9229206 |
| LF\_5 | 80 | 0.0641006 | 0.9494280 |
| LF\_6 | 80 | 0.0362489 | 0.9919129 |
| LFHF\_1 | 80 | 0.0347172 | 0.9997116 |
| LFHF\_2 | 80 | 0.0395405 | 0.9907805 |
| LFHF\_3 | 80 | -0.0245226 | 0.9125398 |
| LFHF\_4 | 80 | -0.0067690 | 0.9762075 |
| LFHF\_5 | 80 | 0.0210187 | 0.9283743 |
| LFHF\_6 | 80 | 0.0502157 | 0.9647115 |

And for episodes:

mydat3 %>%  
 filter(Y != 'Control') %>%  
 psych::describe(quant=c(.25,.75))%>%  
 rownames\_to\_column()%>%  
 as\_tibble()%>%  
 rename(Variable = rowname)%>%  
 filter(Variable != "ID") %>%  
 select(one\_of(c("Variable", "n", "mean", "sd"))) %>%  
 kable()

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | n | mean | sd |
| Stress | 15 | 2.6666667 | 1.1126973 |
| Y\* | 15 | 2.0000000 | 0.0000000 |
| Avg\_niHR | 15 | -0.0826015 | 0.7371774 |
| Start\_niHR | 15 | 0.3151138 | 0.8024897 |
| End\_niHR | 15 | 0.2705922 | 0.7816671 |
| Avg\_HR | 15 | -0.0666642 | 0.7334775 |
| HF\_1 | 15 | -0.4377626 | 0.6725447 |
| HF\_2 | 15 | -0.2474476 | 0.7869229 |
| HF\_3 | 15 | 0.2734294 | 1.2497167 |
| HF\_4 | 15 | -0.1406352 | 1.3177196 |
| HF\_5 | 15 | -0.1472535 | 1.0071068 |
| HF\_6 | 15 | 0.2085135 | 0.9327088 |
| LF\_1 | 15 | -0.3059656 | 0.5529616 |
| LF\_2 | 15 | -0.3664218 | 0.3464467 |
| LF\_3 | 15 | -0.0282664 | 1.1726291 |
| LF\_4 | 15 | -0.0743251 | 1.0559623 |
| LF\_5 | 15 | -0.3418699 | 0.8326587 |
| LF\_6 | 15 | -0.1933276 | 0.5736372 |
| LFHF\_1 | 15 | -0.1851583 | 0.4953266 |
| LFHF\_2 | 15 | -0.2108829 | 0.5768020 |
| LFHF\_3 | 15 | 0.1307875 | 1.0990169 |
| LFHF\_4 | 15 | 0.0361014 | 0.7411999 |
| LFHF\_5 | 15 | -0.1120998 | 1.0242672 |
| LFHF\_6 | 15 | -0.2678170 | 0.7653008 |

## Between Groups Test

None of the features are significantly different using the permutation T test.

mydat3 %>%  
 select(Y:LFHF\_6)%>%  
 summarise\_at(funs(twotPermutation(.[Y == "Control"],  
 .[Y != "Control"],   
 plotit = F)),  
 .vars = vars(Avg\_niHR:LFHF\_6)) %>%  
 kable()

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Avg\_niHR | Start\_niHR | End\_niHR | Avg\_HR | HF\_1 | HF\_2 | HF\_3 | HF\_4 | HF\_5 | HF\_6 | LF\_1 | LF\_2 | LF\_3 | LF\_4 | LF\_5 | LF\_6 | LFHF\_1 | LFHF\_2 | LFHF\_3 | LFHF\_4 | LFHF\_5 | LFHF\_6 |
| 0.722 | 0.166 | 0.216 | 0.776 | 0.038 | 0.258 | 0.218 | 0.509 | 0.494 | 0.346 | 0.172 | 0.0975 | 0.906 | 0.744 | 0.134 | 0.394 | 0.413 | 0.362 | 0.57 | 0.861 | 0.621 | 0.221 |

## Machine Learning

# library(e1071)  
# # make predictions  
# library(ROSE)  
# library(caret)  
#mydat2, subdata$Fold1  
modelfit <- function(data, mytest){  
   
 data=mydat3[-subdata$Fold1,]  
 mytest = mydat3[subdata$Fold1,]  
 ##Generate balanced dataset  
 train.m.bal<-ovun.sample(Y ~ ., data=data, method="both",p=0.6, seed=1342)$data  
 #names(train.m.bal)  
 #table(train.m.bal$Y)  
 # 6:15  
 #[1] "ID" "Stress" "Y" "Avg\_niHR" "Start\_niHR"  
 #[6] "End\_niHR" "Avg\_HR" "HF\_1" "HF\_2" "HF\_3"   
 #[11] "HF\_4" "HF\_5" "HF\_6" "LF\_1" "LF\_2"   
 #[16] "LF\_3" "LF\_4" "LF\_5" "LF\_6" "LFHF\_1"   
 #[21] "LFHF\_2" "LFHF\_3" "LFHF\_4" "LFHF\_5" "LFHF\_6"   
 # just take column 4 to column 19   
 x.train <- train.m.bal[, 4:19]  
 x.train <- as.matrix(x.train)  
 y.train <- train.m.bal$Y  
 #y.train <- as.numeric(train.m.bal$Y)  
 #y.train[which(train.m.bal$Y == "Control")] <- 0  
 #y.train[which(train.m.bal$Y == "Episode")] <- 1  
 #table(y.train) # 0 is control, 1 is episode  
 data.train <- cbind(y.train, x.train)  
 fit<- train(as.factor(y.train) ~ ., data= data.train,method="svmPoly")  
 x.test <- mytest[, 4:19]  
 #x.test <- as.matrix(x.test)  
 ytest <- as.numeric(mytest$Y)  
 #table(ytest)  
 yhat6 = predict(fit, x.test)  
 #yhat6 <-as.numeric(yhat6)  
 #yhat6 <- as.factor(yhat6)  
 ytest <- as.factor(ytest)  
 temp <- confusionMatrix(yhat6,ytest)  
   
 result <- c(temp$overall[1], temp$byClass[1:2]) #<-can change threshold if you want  
 return(result)  
}  
  
#make predictions  
#involve random sampling, need to set the seed  
# set.seed(9)  
# nfolds=4  
# subdata<-createFolds(mydat2$Y, nfolds)  
# t1 <- modelfit(mydat2, subdata$Fold1)  
# t1  
# t2 <- modelfit(mydat2, subdata$Fold2)  
# t2  
# t3 <- modelfit(mydat2, subdata$Fold3)  
# t3  
# t4 <- modelfit(mydat2, subdata$Fold4)  
# t4  
# colMeans(rbind(t1,t2,t3,t4))  
  
set.seed(9)  
nfolds=4  
subdata<-createFolds(mydat3$Y, nfolds)  
t1 <- modelfit(mydat3[-subdata$Fold1,], mydat3[subdata$Fold1,])  
#t1  
t2 <- modelfit(mydat3[-subdata$Fold2,], mydat3[subdata$Fold2,])  
#t2  
t3 <- modelfit(mydat3[-subdata$Fold3,], mydat3[subdata$Fold3,])  
#t3  
t4 <- modelfit(mydat3[-subdata$Fold4,], mydat3[subdata$Fold4,])  
#t4  
#colMeans(rbind(t1,t2,t3,t4))

Here are the mean accuracy scores for the machine learning approach:

colMeans(rbind(t1,t2,t3,t4)) %>%  
 kable()

|  |  |
| --- | --- |
|  | x |
| Accuracy | 0.8333333 |
| Sensitivity | 0.8500000 |
| Specificity | 0.7500000 |