PML_Project_Report

Michael O'Dell

August 22, 2014

Executive Summary

This analysis uses the random forest algorithm to build three in-sample predictive models. The estimated out-of-sample error (Kappa) from those three models is the average of their in-sample errors: `0.8463`.

The out-of-sample error (Kappa) of the last model (using 52 variables) predicted on a one-time use test set (separate from the 20 test cases) is `0.889`.

Project Objective

This project is an exercise in predictive modeling. The goal is to predict the manner in which an exercise was performed for twenty observations in a test data set by creating a predictive model from a training data set and applying that model to the test data.

Process Overview

The prediction process consists of five steps:

1. Question: The goal of this project is to predict the manner in which subjects performed an exercise.
2. Identify Appropriate Data: The data for this project has been provided as part of the assignment.
3. Select/Create Features: Identify and/or create covariants that best explain the outcome variable.
4. Identify Algorithms: Choose the appropriate modeling algorithm given the features
5. Estimate Parameters: Estimate the prediction parameters based on the selected algorithm
6. Evaluate: Evaluate the algorithm on new data.

Question:

Background

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity.This data is often used to quantify how much of a particular activity they do, but rarely used quantify how well the activity is done.

This project examines data collected from participants correctly and incorrectly executing a specific exercise. Four axis inertial measurement units (IMUs) located on the belt, armband, glove, and dumbbell measured three-axis acceleration, gyroscopic, and magnetometer data while 6 participants conducted 10

repetitions using five standardized forms (one correct and four inccorrect) of the Unilateral Dumbbell Bicept Curl. More information is available from the website here: http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset).

Data

The data consists of four sets of 38 sensor variables (one set for each IMU) plus an additional 8 variables that include an observation index, three time stamps, a sliding time-series data sample window flag (does the observation start a new time-series of observations), a time-series data sample index, and for the training data, a class variable indicating which of the 5 forms in which the exercies was conducted.

Of the 38 sensor variables, * Nine are the raw three-axis (x,y,z) data from the three IMU sensors * Three more are the calculated Euler Angles from the raw data (pitch, roll, yaw) * 24 are eight features calculated on the Euler Angles over the all of observations in a given data sample (average, standard deviation, variance, maximum, minimum, amplitude, skewness, and kurtosis) * The remaining two variables are the total acceleration calculated for each observation from the three-axis accelerometer data and the total accerlation variance calculated over the time window.

Count of NAs:

```
##                   [,1]    [,2]    [,3]    [,4]    [,5]    [,6]    [,7]
## Obs_missing_data   "   0"  "9555"  "9556"  "9557"  "9558"  "9559"  "9574"
## Count_of_Variables "60"    "68"    " 2"    " 4"    " 3"    " 4"    " 2"
## Percent_NAs        "0.00%" "0.98%" "0.98%" "0.98%" "0.98%" "0.98%" "0.98%"
##                   [,8]    [,9]    [,10]   [,11]   [,12]   [,13]   [,14]
## Obs_missing_data   "9599"  "9600"  "9601"  "9602"  "9603"  "9604"  "9761"
## Count_of_Variables " 1"    " 4"    " 2"    " 1"    " 1"    " 2"    " 6"
## Percent_NAs        "0.98%" "0.98%" "0.98%" "0.98%" "0.98%" "0.98%" "1.00%"
```

Upon inspection, over 98% of the derived time-series observations are missing, and while they can be calculated since the observations have a time-series index (num_window), this poses a potential problem for the prediction of the test set part of the assignment.

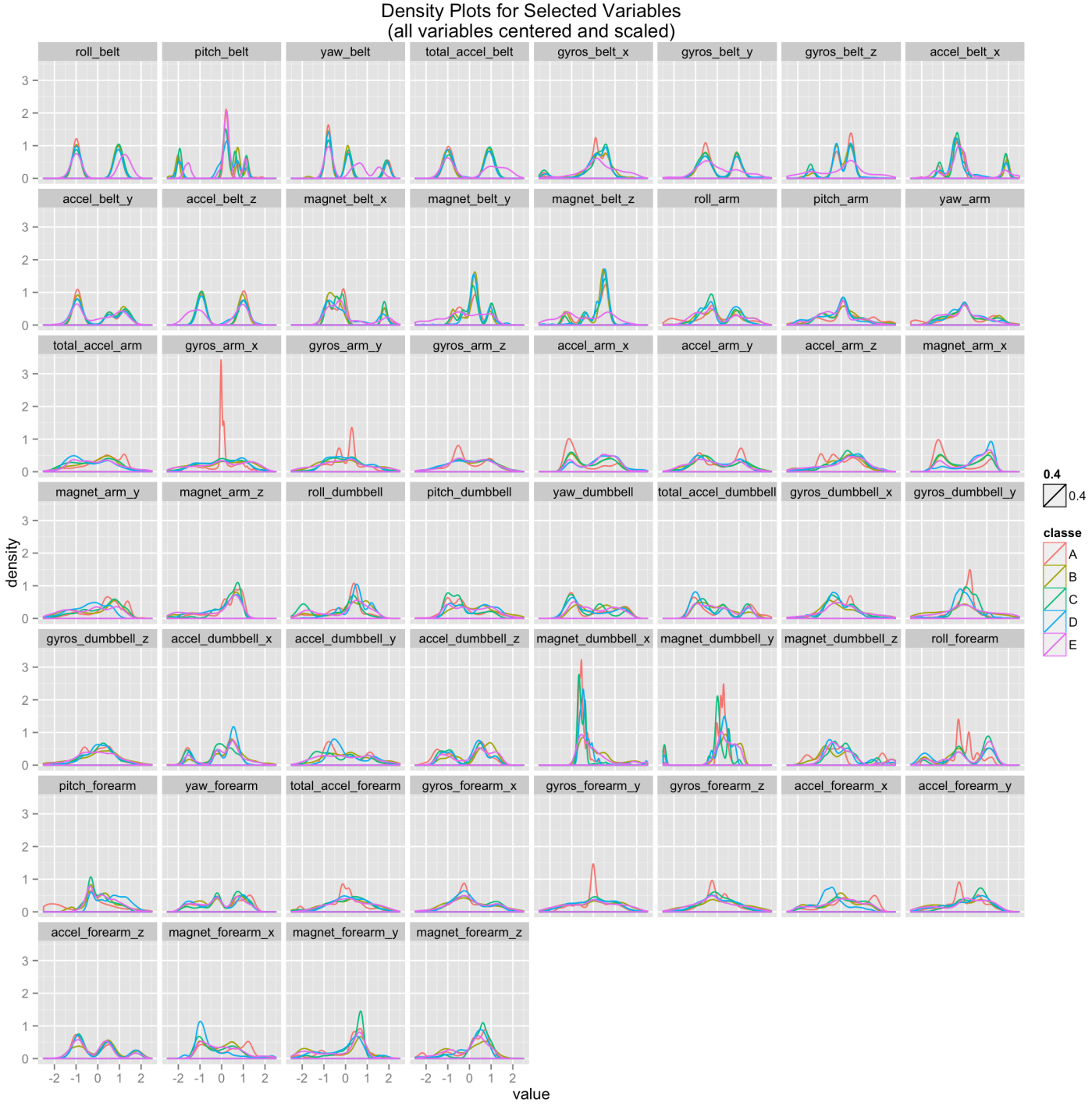
Per the assingment instructions, the test set consists of twenty observations and the assignment is to predict the class of each observation–thus implying that some, if not all, of the test observations are not part of a time-series and thus for which time-series variables cannot be calculated.

To account for both possiblities, isolated observations and observations from a single time-series, models will be tested with and without those variables.

To do this, the provided training set is split into a training and test set by sampling complete time-series window indexes (keeping series together). The test set is then split into two sub-test sets (test1 for validating a models without time-series features and test2 to validate a model with time-series features). Test1 is randomly subsetted to create a test (test1a) and validation (test1b) set, thus allowing for out of sample error to be calculated for both a model with and a model without time-series features.

Exploratory Plots

Plotting density (histogram) plots for the 52 non time-series variables (9 three-axis, 3 Euler Angles + total acceleartion for four IMUs) data for a single user reveals a few variables with distributions close to normal, but no clear mean separation between outcomes for any variables. Including all users shows all variables are multi-modal suggesting that parametric modeling will not be effective.



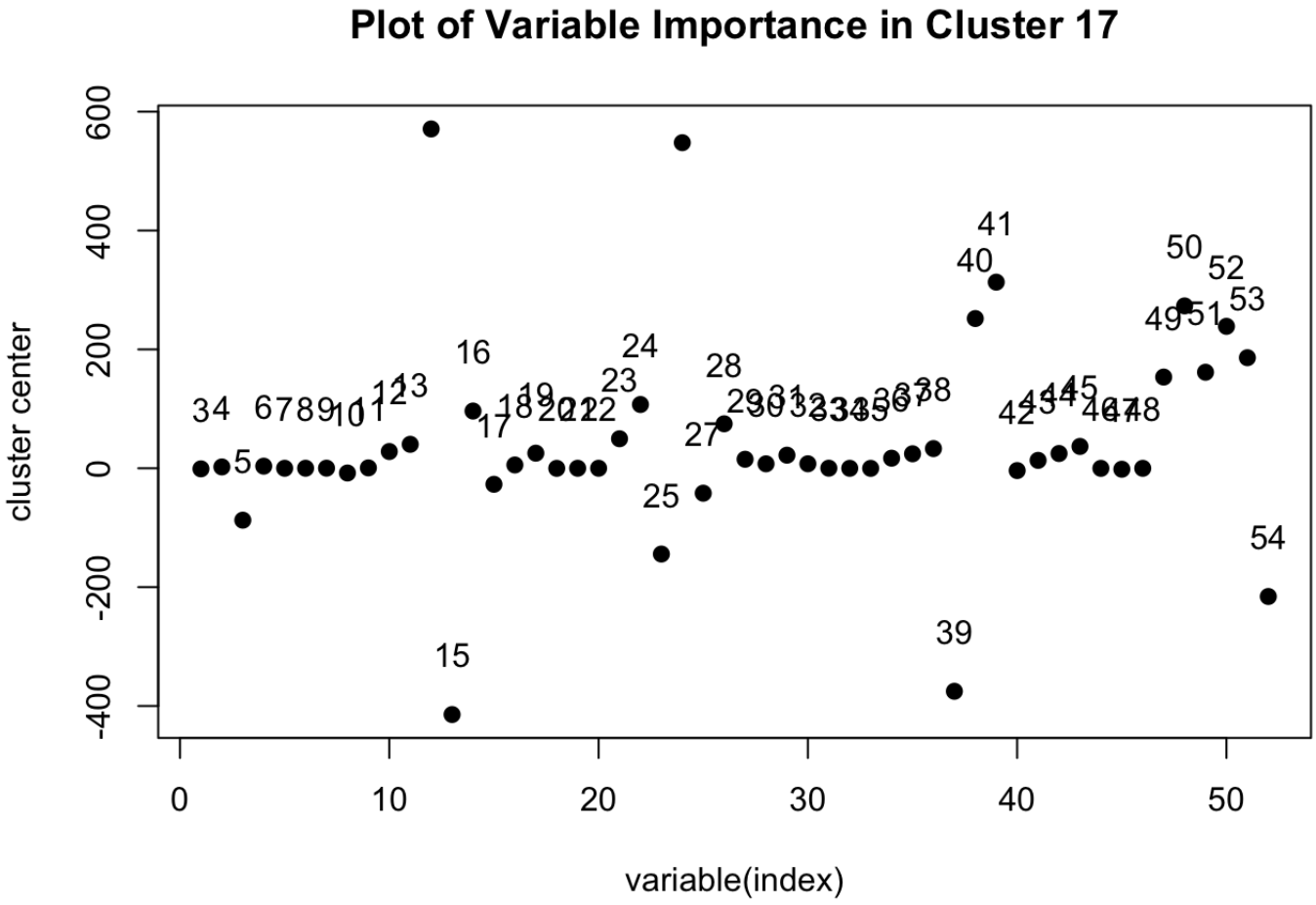
Imputing Data and/or New Covariants

Given that the 52 sensor variables have no missing values, there is no need to imput data for them. However, if these 52 variables do not yeild a set of factors sufficient to build a good predictive model, new covarients (such total gyros and total magnet–similar to the included total acceleration) will need to be calculated.

Factor selection

Since none of the variables have clear separation of outcome means, K-means clustering may be helpful. Calculating clusters for 30 centers and multiple restarts produces no clear distinction between classes with the exception of class A for which variables: magnet_belt_y, magnet_arm_z, magnet_dumbbell_y, and magnet_forearem_y look to play a major role.

```
##
##        A   B   C   D   E
##   1   40  24   0  36  28
##   2  201   9   9  13  20
##   3  216   5  24  16  15
##   4  126  77   8  98  92
##   5  157  53  39   5  89
##   6  543  13  44   0  10
##   7   92 149 116  77  58
##   8   52  57 118  21  18
##   9  120  59  28  62  89
##   10   3 191  83   7 123
##   11  79  73  91 103  53
##   12   0  87   0   0   0
##   13  65  39  51 162  56
##   14  78  49  72 202 103
##   15  21  90  25 138 105
##   16  33  58  43  96   6
##   17  17  26  23  36  47
##   18  35  70   5   3  22
##   19  49   6  18  42   0
##   20  44  39   5  34  45
##   21  99 112 122   0  74
##   22 144 114  62  41  31
##   23  14 108   0  53 104
##   24 313  13  13   6  14
##   25 257  71  94  31 107
##   26  20  50  86  17  29
##   27  60 110 106 149 136
##   28 102  37  65  69  68
##   29   6 216  34   5  47
##   30  72  12  99  67  25
```



Adding additional covarients calculable for single observations, (total acceleration (dropping the existing total_accel variables since they are an order of magnitude smaller than the actual total acceleration), total gyro, and total magnet vector magnitudes calculated using the formula:

$$Total_{vectormag} = \sqrt{x^2 + y^2 + z^2}$$

for all IMUs yields 12 additional variables, but produce no better results when clustering.

Noting the four variables identified for class A through clusting and visual inspection of density distributions suggests a first cut set of feature variables:

```
f20 <- c(
"magnet_belt_y",
"magnet_arm_z",
"magnet_dumbbell_y",
"magnet_forearm_y",
"magnet_forearm_z",
"yaw_belt",
"total_accel_belt",
"gyros_arm_z",
"gyros_dumbbell_y",
"accel_dumbbell_x",
"roll_dumbbell",
"yaw_dumbbell",
"magnet_dumbbell_z",
"accel_forearm_x",
"accel_dumbbell_z",
"magnet_belt_x",
"pitch_belt",
"magnet_forearm_x",
"accel_dumbbell_x",
"accel_dumbbell_y"
)
```

Algorithm

Given the non-normal distributions of the data, the non-parametric Random Forest may yield the best results. Random Forests do not require pre-processing other than removal of NAs and the data set of 52 variables is free of NAs.

```
## Loading required package: randomForest
## randomForest 4.6-7
## Type rfNews() to see new features/changes/bug fixes.
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##      8.842e-01      8.543e-01      8.710e-01      8.965e-01      2.376e-01
## AccuracyPValue  McnemarPValue
##      0.000e+00      5.495e-10
```

Given that the model must correctly predict 20 observations, this model will likely miss 2 (or more likely 3 given that in-sample error is optimistic). A 2nd feature set of just the three-axis data (36 variables) is worse.

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##      8.534e-01      8.154e-01      8.390e-01      8.671e-01      2.376e-01
## AccuracyPValue  McnemarPValue
##      0.000e+00      8.803e-19
```

Absent the including time-series variables such as minimum, maximum, variance, mean, etc. the kitchen sink feature set of all 52 non time-series variables is the best bet although risks over fitting the data.

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##      8.966e-01      8.698e-01      8.840e-01      9.082e-01      2.376e-01
## AccuracyPValue  McnemarPValue
##      0.000e+00      2.950e-20
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   A   B   C   D   E
##          A 554  31  13  36   1
##          B  27 426  50   2   1
##          C   5  11 437  20  26
##          D   7   0  27 310  14
##          E   2   2  12   3 487
##
## Overall Statistics
##
##                Accuracy : 0.884
##                  95% CI : (0.871, 0.896)
##     No Information Rate : 0.238
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.854
##  Mcnemar's Test P-Value : 5.49e-10
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.931    0.906    0.811    0.836    0.921
## Specificity             0.958    0.961    0.968    0.977    0.990
## Pos Pred Value          0.872    0.842    0.876    0.866    0.962
## Neg Pred Value          0.978    0.978    0.949    0.972    0.979
## Prevalence              0.238    0.188    0.215    0.148    0.211
## Detection Rate          0.221    0.170    0.175    0.124    0.194
## Detection Prevalence    0.254    0.202    0.199    0.143    0.202
## Balanced Accuracy       0.944    0.934    0.890    0.907    0.955
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   A   B   C   D   E
##          A 558  48  13  35  12
##          B  16 398  67   6  23
##          C  12  15 433  34   8
##          D   3   3  10 285  23
##          E   6   6  16  11 463
##
## Overall Statistics
##
##                Accuracy : 0.853
##                  95% CI : (0.839, 0.867)
##     No Information Rate : 0.238
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.815
##  Mcnemar's Test P-Value : <2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.938    0.847    0.803    0.768    0.875
## Specificity             0.943    0.945    0.965    0.982    0.980
## Pos Pred Value          0.838    0.780    0.863    0.880    0.922
## Neg Pred Value          0.980    0.964    0.947    0.961    0.967
## Prevalence              0.238    0.188    0.215    0.148    0.211
## Detection Rate          0.223    0.159    0.173    0.114    0.185
## Detection Prevalence    0.266    0.204    0.200    0.129    0.200
## Balanced Accuracy       0.941    0.896    0.884    0.875    0.928
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   A   B   C   D   E
##          A 570  31   4  15   6
##          B   9 428  69  11  22
##          C   7   8 446  29   7
##          D   2   1   5 311   4
##          E   7   2  15   5 490
##
## Overall Statistics
##
##                Accuracy : 0.897
##                  95% CI : (0.884, 0.908)
##     No Information Rate : 0.238
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.87
##  Mcnemar's Test P-Value : <2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.958    0.911    0.827    0.838    0.926
## Specificity             0.971    0.945    0.974    0.994    0.985
## Pos Pred Value          0.911    0.794    0.897    0.963    0.944
## Neg Pred Value          0.987    0.979    0.954    0.972    0.980
## Prevalence              0.238    0.188    0.215    0.148    0.211
## Detection Rate          0.228    0.171    0.178    0.124    0.196
## Detection Prevalence    0.250    0.215    0.198    0.129    0.207
## Balanced Accuracy       0.964    0.928    0.901    0.916    0.956
```

The estimated out-of-sample error using the in-sample errors of the three cross validated models is the average of their indvidual errors. In this case, using the concordance error measure Kappa, the estimated out-of-sample error is 0.8463.

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##      9.117e-01      8.888e-01      8.999e-01      9.225e-01      2.377e-01
## AccuracyPValue  McnemarPValue
##      0.000e+00      8.822e-14
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   A   B   C   D   E
##          A 581  30   4  19   5
##          B   5 424  44   7  24
##          C   3  12 475  28   7
##          D   3   0   7 315   5
##          E   3   4   9   2 487
##
## Overall Statistics
##
##                Accuracy : 0.912
##                  95% CI : (0.9, 0.923)
##     No Information Rate : 0.238
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.889
##  Mcnemar's Test P-Value : 8.82e-14
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.976    0.902    0.881    0.849    0.922
## Specificity             0.970    0.961    0.975    0.993    0.991
## Pos Pred Value          0.909    0.841    0.905    0.955    0.964
## Neg Pred Value          0.992    0.977    0.968    0.974    0.979
## Prevalence              0.238    0.188    0.215    0.148    0.211
## Detection Rate          0.232    0.169    0.190    0.126    0.195
## Detection Prevalence    0.255    0.201    0.210    0.132    0.202
## Balanced Accuracy       0.973    0.931    0.928    0.921    0.957
```

The out of sample error for the final model using 52 variables is `0.889`.

Time-series Features

Given that the data observations are time series, there is significant information in the sequence of observations as defined by the `num_window` variable. While the 20 test cases for the single observations and cannot be pre-processed to create variables such as mean, variance, standard deviation, maximum, minimum, when initially sampling test sets from the training data, I sampled a test set by `num_window`. With this test set, one can test the effect of including some time series variables.

And while modeling with these time-series variables is outside the scope of this assignment, the power of such variables can be seen in the fact that the researchers who collected this data, created a random forest predictive model with only 17 time-series variables with a much higher sensitivity.