Article can be accessed online at http://www.publishingindia.com

Ensemble Approach for Zoonotic Disease Forecasting using Machine Learning Techniques

Vikash Chandra Sharma*, David Frankenfield, Anupam Gupta***, Rama Krishna Singh ******

Abstract

More than two-third of emerging infectious diseases in recent decades are zoonotic in origin. Timely prediction of these diseases which migrate from animals to humans and preventive measures to stop the loss in terms of morbidity and mortality is the requirement of healthcare industry. Avian Influenza is one of the zoonotic diseases that have created havoc in recent past especially in Asian subcontinent. In past, attempts have been made to predict influenza using traditional time-series techniques (AR, MA, ARMA, ARIMA etc.) as well as machine learning techniques to capture the cyclicity and seasonality of these virus strains. In current research an effort has been made to utilize the Empirical Mode Decomposition (EMD) to extract the Intrinsic Mode function (IMF) and then apply state of art Machine Learning (ML) techniques to predict the series. Several machine learning techniques like Random Forest (RF) along with Gradient Boosting Machine (GBM) and Support Vector Regression (SVR)have been applied on the decomposed series. Exogenous models showed variables like temperature, humidity and precipitation have been incorporated to improve upon the forecast. An ensemble approach of ML models showed significant improvement over the traditional models in terms of long term forecast accuracy.

Keywords: Random Forest, Gradient Boosting Machine, Support Vector Regression, Machine Learning, Avian Influenza

Introduction

In healthcare industry, application of time-series modeling and prediction of future outbreak of certain infectious diseases and disease events which occur in a cyclic or rhythmic pattern are very crucial. The forecasting of disease helps to predict the course of disease, warn healthcare experts and adopt control measures to prevent disease outbreaks. The US Agency for International Development launched its Emerging Pandemic Threats Programme in late 2009 to build an early warning system to detect and reduce the impacts of zoonotic diseases.

Zoonotic diseases are a group of infectious diseases naturally transmitted from animals to humans. Avian influenza (AI) under consideration is one of those zoonotic diseases which posea major threat to mankind in recent years. It refers to the disease caused by infection with avian (bird) influenza (flu) Type A viruses. Humans are affected by AI virus subtypes H5N1 and H9N2 and swine influenza virus subtypes H1N1 and H3N2. The AI (H5N1) virus subtype, a highly pathogenic AI virus, first infected humans in 1997 during a poultry outbreak in Hong Kong and China. Since its widespread re-emergence in 2003 and 2004, this avian virus has spread from Asia to Europe and Africa and has become entrenched in poultry in some countries, resulting in millions of poultry infections and many human deaths. The mortality and morbidity associated with this disease have devastated communities in some countries and led to global changes in public health. Countries not only suffered huge economic loss but in some instances closed down - global travel and trade networks. These vulnerabilities emphasize the need for a systematic, pre-emptive, advanced and improved predictive modeling approach to predict the emergence of such pandemics that could impact the health risk to susceptible human population.

AI also has some cyclic or repeating pattern to capture which, traditional time-series predictions are performed

* Senior Consultant at UnitedHealth Group, Noida, Uttar Pradesh, India. Email:vikash_c_sharma@uhc.com
** Director at United HealthCare, Greater Denver Area, United States. Email: david_l_frankenfield@uhc.com
*** Director at UnitedHealth Group, New Delhi, India. Email: anupam_gupta@uhc.com
**** Associate Director at UnitedHealth Group, Noida, Uttar Pradesh, India.Email: rama_k_singh@uhc.com

using the autoregressive integrated moving average (ARIMA) (Box & Jenkins, 1970 technique. ARIMA model attempts to filter out high frequency noise in the data to detect local trends based on linear dependence in observations in the series. ARIMA models even though widely applied incorporate lot many assumptions. First, it assumes linear relationship between independent and dependent variables. Real-world relationships are often non-linear and therefore more complex than the assumptions build into ARIMA model. As a result this model does not perform well when data structure is complex. Also, these models assume a constant standard deviation of errors over time. This assumption can be removed when ARIMA is used in conjunction with a Generalized Auto Regressive Conditional Heteroskedasticity (GARCH) (Engle, 1995) model. GARCH technique attempts to characterize model's non-constant standard deviations in a time-series but it comes with its own challenges and optimizing the parameters for GARCH is always a challenge.

Another challenge in time series prediction is the prediction horizon. When the prediction horizon increases, the uncertainty of future trends also increases, rendering a more tough prediction problem. Researchers have always wanted to extract the maximum knowledge from the past values to better utilize them for long-term time series prediction. More recently, new classes of regression models along with machine learning techniques have been developed to address the challenges associated with classical methods. Literature suggests the usage of Random Forest (Kane, Price, Scotch & Rabinowitz, 2014) technique for superior prediction accuracy when compared to the classical models. This paper makes use of an ensemble of Random Forest with Gradient Boosting Machine and Support Vector Regression models to come up with a better forecast for longer duration which can then be used for future planning and taking preventive steps to contain the disease from spreading and transforming into epidemic.

Data

Data for avian influenza virus were collected from the online web-based application (EMPRES-i) (http://empres-i.fao.org/eipws3g/)which has been designed to support veterinary services by facilitating the organization and access to regional and global disease information. This platform is a global animal disease information system including emergent zoonoses and other high impact animal diseases. In this research, data were considered for a period of Jan'06 to Jul'14 for some Asian countries (China, India, Nepal, Bangladesh, Vietnam, and South Korea) because these countries historically seemed to witness majority of outbreak cases of AI. Fig. 1 demonstrates the number of outbreak cases in Asian countries considered in the research work.

Fig. 2 shows the monthly time plot for avian influenza outbreak cases in nations under consideration.

Avian influenza was forecasted taking into consideration few exogenous meteorological variables like daily temperature (minimum and maximum temperature), relative humidity and precipitation. The data were obtained from http://globalweather.tamu.edu/ for the Asian countries in the study for a period of Jan'06 to Jul'14. The data from different sources were merged into a single data set and was then brought to monthly level taking the number of outbreak cases, average minimum temperature, average maximum temperature, average humidity, and average precipitation.

Methodology

Preliminary Analysis

The model was fitted considering data for a period of Jun'06 to Dec'12 and the forecast was validated on a period of Jan'13 to Jul'14. This period was selected considering the availability of data and to maintain the consistency of training and validation period across all modeling methods. Univariate time series analysis was performed on the data using few of the traditional forecasting techniques like Holt Winter's and ARIMA models to get a benchmark estimate. Improvement was observed with application of models like AR-GARCH model over traditional forecasting models. Actual vs fitted records are highlighted to show the performance of traditional models in Fig. 3. A detailed R syntax is shared in Appendix A for reference.

However, real world forecasting processes involve complex nonlinear series having large number of components. In the study for zoonotic diseases, occurrences are driven by lot many factors like weather, humidity, cleanliness, income group, etc. It is difficult to analyse such disease as

Fig. 1: Avian Influenza Outbreaks for a Period of Jan'06 to Jul'14

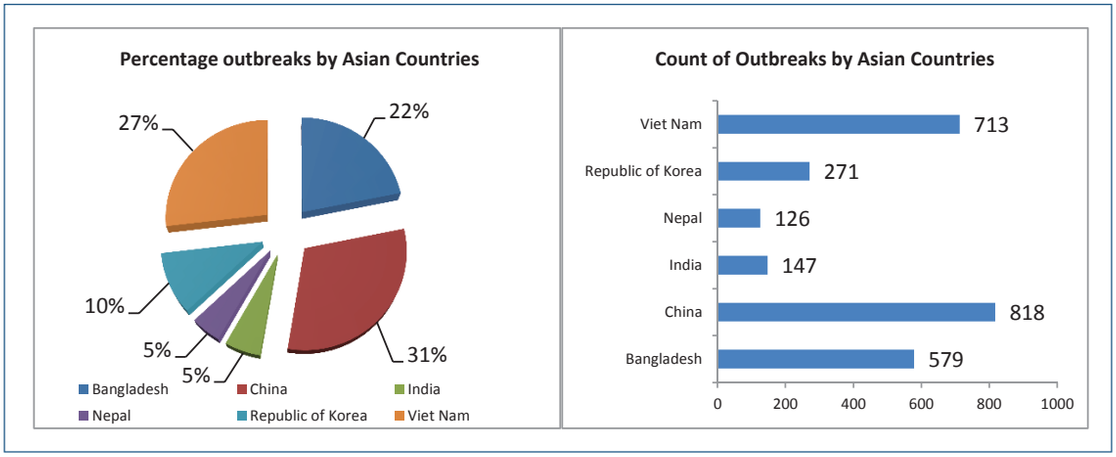


Fig. 2: Time Plot for Avian Influenza Outbreaks in Asia

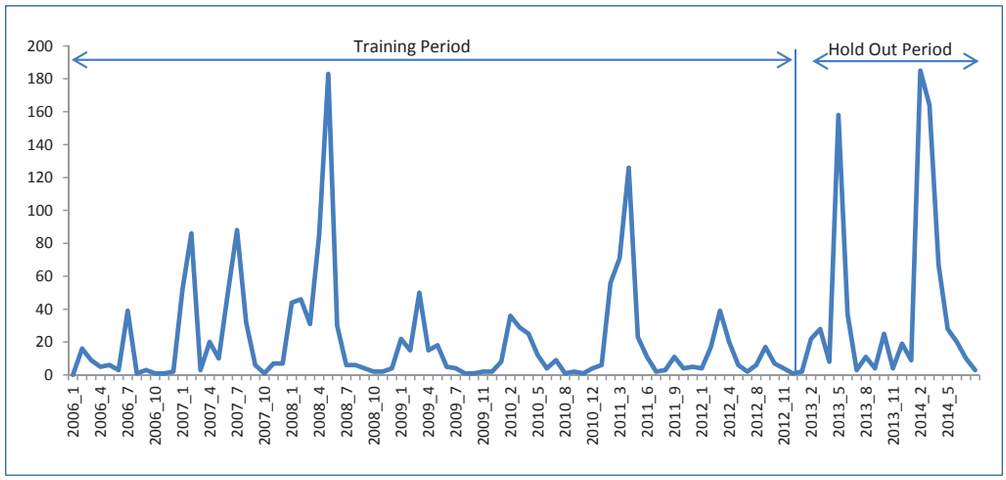
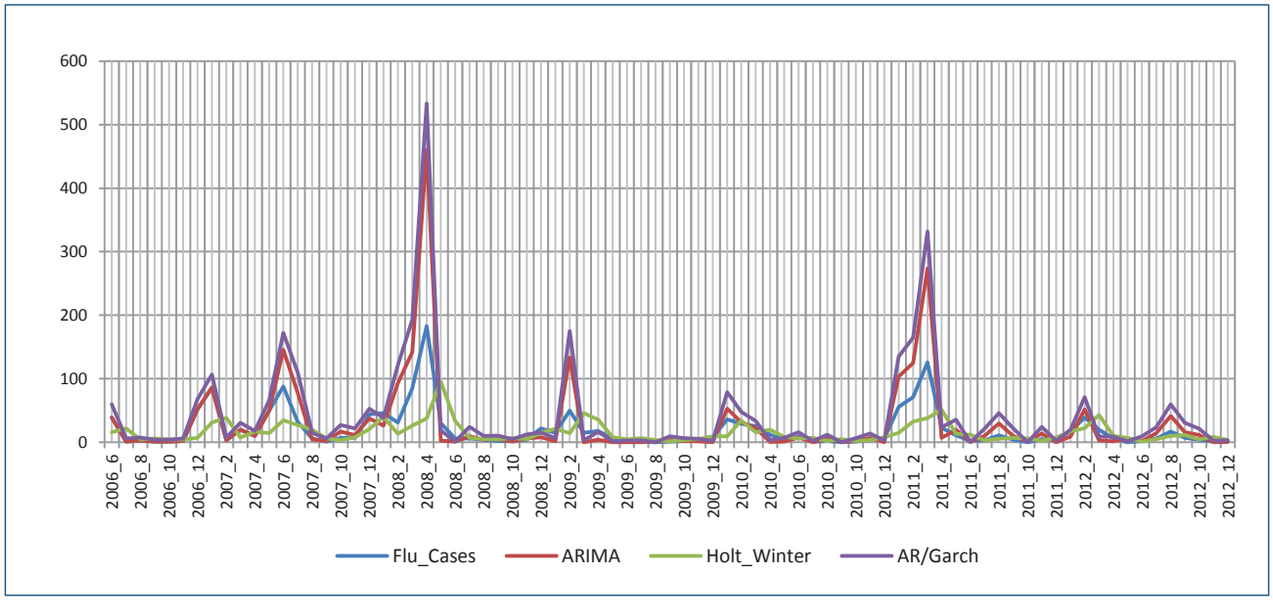


Fig. 3: Actual vs Fitted for Traditional Models



its components, when interacting with each other, mask and distort the regularities which need to be identified. This gives rise to the requirement to break down the process under consideration into individual components and analyse each and every component separately. Analysis of individual component and consideration of contribution they make into the process at hand helps us understand the process better as well as increases forecast reliability.

In the research work, decomposing the monthly influenza outbreaks using Empirical Mode Decomposition (EMD) (Wu & Hu, 2006) technique was done. Exogenous variables like temperature, humidity, and precipitation were used and a set of machine learning techniques like Gradient Boosting Machine, Support Vector Regression, and Random Forest were applied on EMD decomposed data to come up with the forecasted value. Models were compared on basis of mean absolute percentage error (MAPE) on the hold out validation period. For all modeling exercise, R-Studio programming environment was used and various packages were considered from CRAN (http://CRAN.R-project.org).

Empirical Mode Decomposition

Empirical Mode Decomposition is a decomposition technique which was proposed as the fundamental part of the Hilbert-Huang transform (HHT) (Huang, Shen, Long, Wu, Shin, Zheng, Yen, Tung & Liu, 1998). In contrast to other decomposition techniques, the EMD decomposes any given data into intrinsic mode functions (IMF) that are not set analytically and are determined by analysed sequence only. The basic functions are determined directly from the input data. An IMF resulting from the EMD shall satisfy the following requirements:

1. The number of IMF extrema (the sum of the maxima and minima) and the number of zero-crossings must either be equal or differ at most by one;
2. At any point of an IMF, the mean value of the envelopes defined by local maxima and local minima shall be zero.

Decomposition contains a family of frequency ordered IMF components. Each successive IMF contains lower frequency oscillations than the preceding one.

Fig. 4 depicts the analysed sequence of the thin blue line which is the actual series under consideration. The envelopes are shown in green. Mean is calculated based on the two envelopes and then subtracted from the initial sequence. To obtain the final IMF, new maxima and minima shall be identified and all the above steps repeated until stoppage criteria are met. This recursive processof subtracting the mean of envelopes from the initial sequence is called sifting. Fig. 5 shows the sifting process applied on the monthly Avian Flu cases recorded for Asian countries. The sifting process continues until the mean value of the minima and maxima envelopes becomes zero and that is the first IMF extract.

Monthly avian influenza outbreak data were first tested for various transformations and then square root transformation was considered to reduce on the variance

Fig. 4: Plotting the Envelopes and their Mean

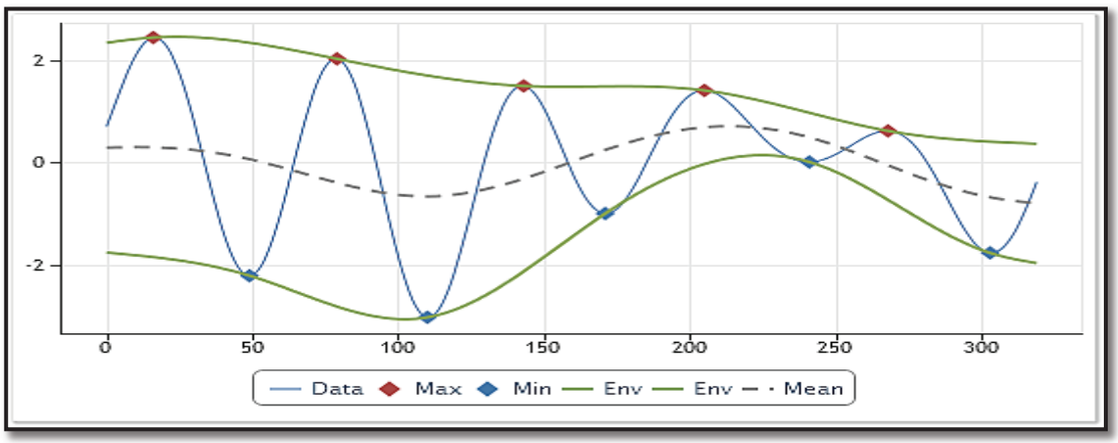


Fig. 5: Sifting Simulations on Monthly Avian Flu cases in Asia

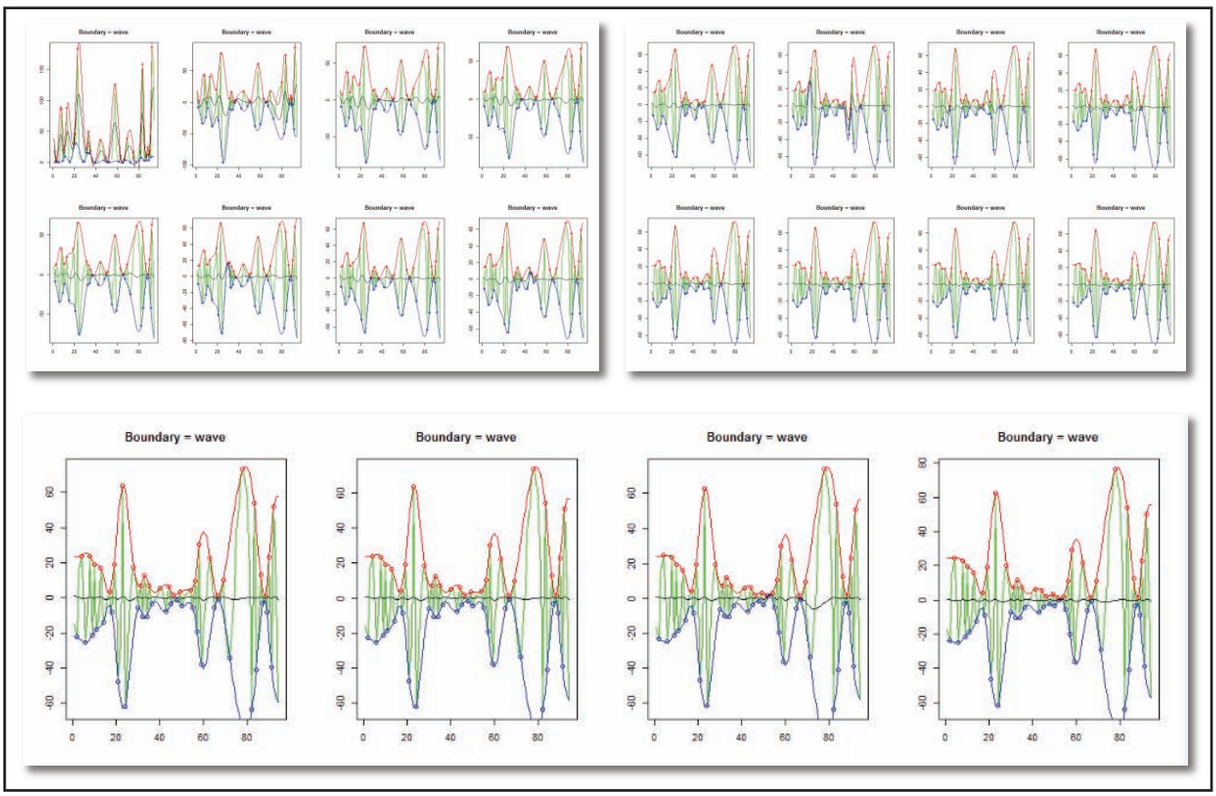
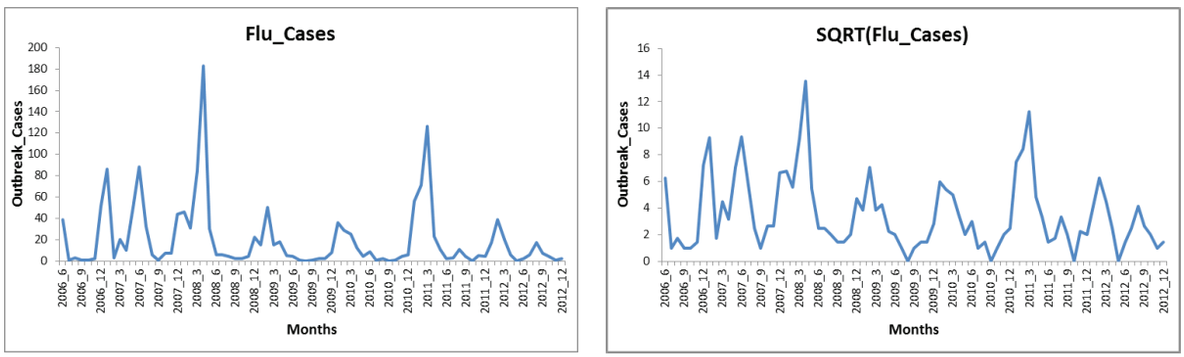


Fig. 6: Modified Series Using Square Root Transformation



along with handling months which had no outbreak cases as shown in Fig. 6.

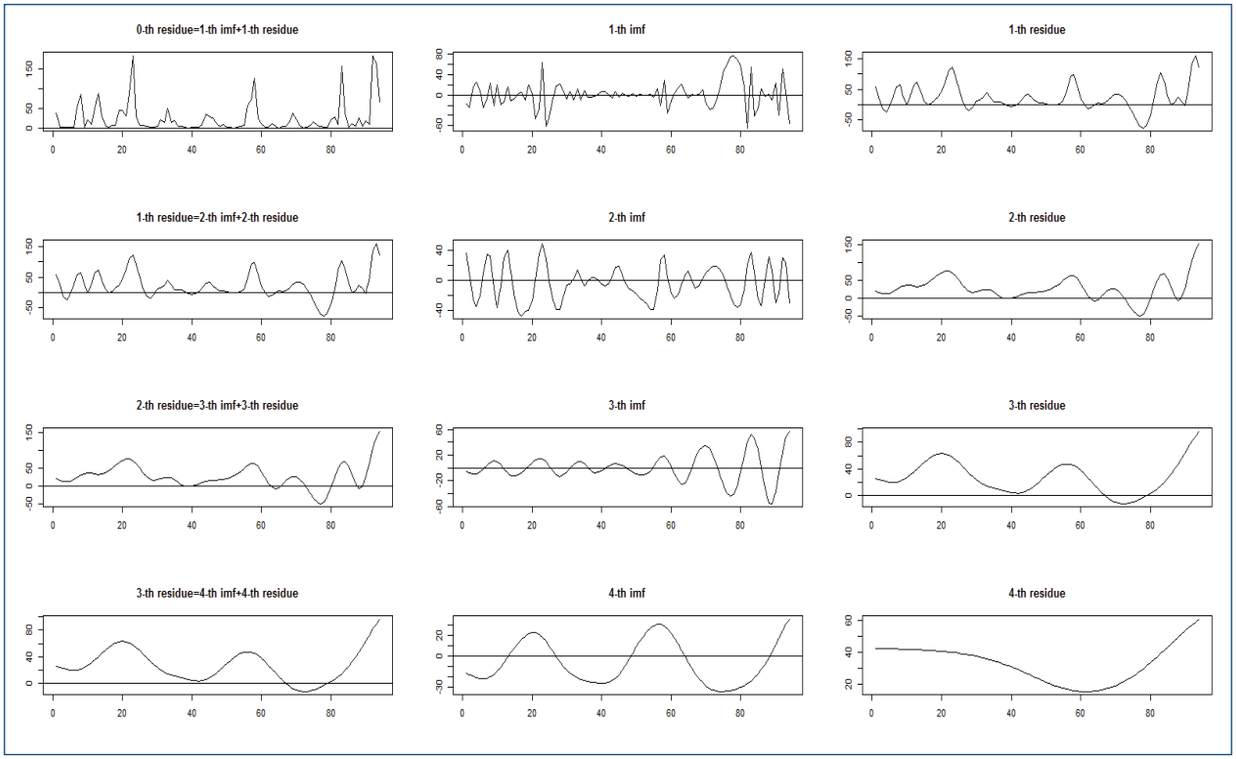
This transformed series was then decomposed into various IMFs using the EMD algorithm. Fig. 7 demonstrates the decomposition of the monthly series into varying frequency IMF series.

A maximum of 4 IMF series was extracted using the EMD algorithm for the given flu data. Along with the above 4 IMF series, the final residue series was also to be forecasted for a longer horizon.These extracted series were then forecasted using various machine learning techniques. All of these individual forecasts were summed up to obtain the final forecasted series.

Application of Machine Learning

Comparative study was performed on the given dataset with a combination of machine learning models random forest, support vector machine, and gradient boosting

Fig. 7: EMD Decomposition of Monthly Avian Flu Cases



machine.

Random forest (RF) is a typical machine learning technique which starts by creating decision trees in a recursive fashion. It selects a subset of available features and recursively partitions the data in the regression space until the amount of variation in the subspace is small. Random forest as a technique is greedy and as a result, does not necessarily converge to the global optimal solution. In order to avoid such indecisive convergence, a collection or ensemble of locally optimal trees is done which is termed as bagging. The ensemble of such trees is known as forest. Variables considered were the lagged values of temperature, humidity, precipitation, and seasonality indicators. All of these variables were scaled and centered. The model used 1000 trees with a grid search approach to sample the efficient number of features to be selected to build the final model with least root mean square error.

Another machine learning technique Support Vector Regression (SVR) (Scholkopf, 1997)is applied for forecasting in regression framework by introducing an alternative loss function. The loss function is modified to include a distance measure. It employs a rich class of non-linear modeling functions via kernels. For the current research, svmPoly kernel was used to decipher the support vectors.This kernel takes in three parameters namely degree, scale, and cost. A grid search was performed to choose these parameters automatically. Root mean square error was the metric considered to select the efficient parameters for every model.

Finally, a class of machine learning models Gradient Boosting Machine (GBM) (Friedman, 2001) which is again a tree-based model involving a recursive addition to the initial learning from the residuals was applied. It fits a tree-based model on the residuals using the specified list of variables at hand and explains the variance in the residuals. Total number of trees specified for model building was 500 with interaction depth as 5 and learning weight ofiteration was 0.1.

Fig. 8 shows the fit for the class of machine learning models discussed. Detailed syntax used for developing these machine learning models along with IMF extraction is appended in Appendix B for reference.

Fig. 8: Fit using Random Forest, Support Vector and Gradient Boosting

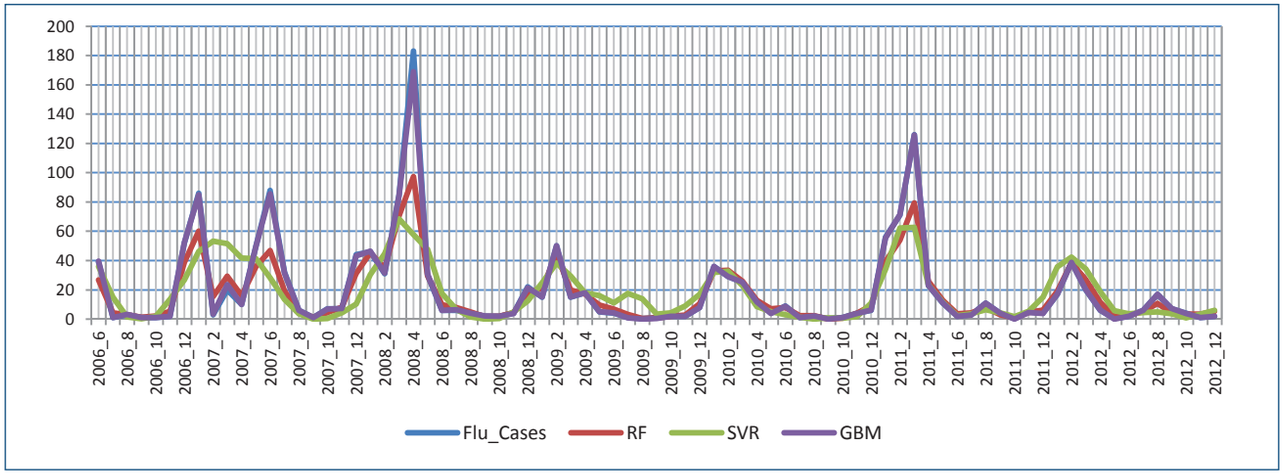
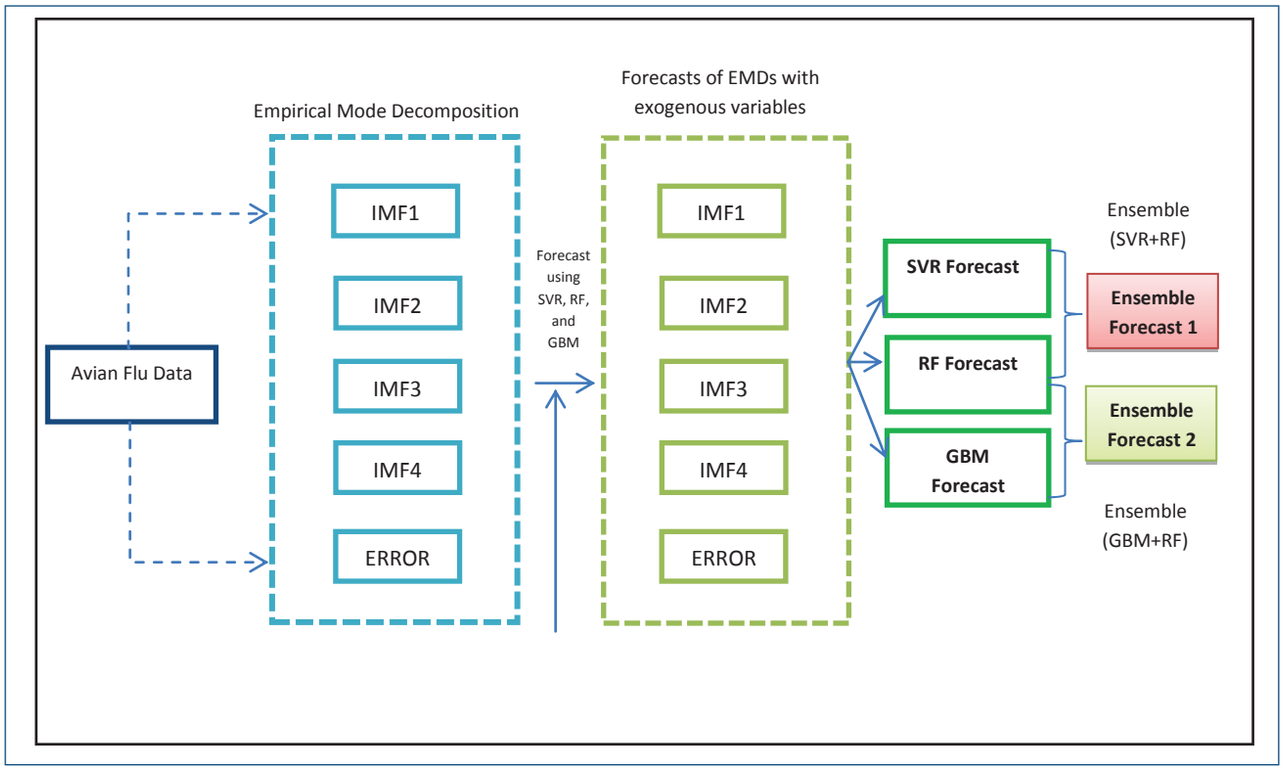


Fig. 9: Demonstration of Ensemble of various Machine Learning Models



Ensemble Approach

A state of art ensemble approach for harnessing the power of all the machine learning models was applied for coming up with the efficient solution. A flowchart explaining the ensemble strategy is shown in Fig. 9.

Ensemble was done in order to come up with a stable forecast for longer horizon. Each of these machine learning techniques had a better fit when compared to any of those classical benchmark techniques applied during the preliminary analysis. A lot of numerical iterative methods along with some intuitive combination of models were performed to come up with ensemble coefficients.

Table 1: Comparison of Mean Absolute Percentage Errors

Forecasting Methods	*MAPE(DEVJun'06 – Dec'12)*	*MAPE(VAL Jan'13 – Jul'14)*
Traditional Univariate Forecasting Methods		
Holt's Winters	152.0%	108.5%
ARIMA	90.7%	89.5%
AR-GARCH	102.3%	88.9%
Machine Learning Techniques		
Support Vector Regression	75.4%	69%
Random Forest	65%	60%
Gradient Boosting Machine	24.3%	49.2%

As shown in Fig. 9, Ensemble Forecast 1 is a combined result of first half of a year using SVR and second half of a year using RF model. Similarly, Ensemble Forecast 2 is a combined result of first half of a year using GBM and second half of a year using RF model.

Results

A summary shown in Table 1 compares results of individual machine learning models and shows improvement overtraditional univariate models. The model performance is compared in the hold out period (Jan'13 to Jul'14) basis the mean absolute percentage error. There was a significant improvement observed in the forecasts that were obtained using machine learning methods as compared to the traditional methods.

The results from the ensemble approach seemed to reduce on the error when compared to individual machine learning approaches and the traditional approaches. Ensemble 2, which constituted of intuitive ensemble of GBM and RF, considered the output of Gradient Boosting Machine for the first half of a year and output of Random Forest for the next half of the year to come up with the complete year forecast. This approach of ensemble significantly is reduced on the MAPE when compared to the best benchmark set by traditional models. For the hold out period Ensemble 2 model is reduced on the MAPE to 44.6% in comparison to best performing traditional model at 88.9% MAPE. Table 2 is a tabular view to the MAPE obtained in the development and validation periods for the two ensemble approaches used.

A visual display to show the model performance in the hold out period for different ensembled model is shown in Fig. 10. The green line which is the ensemble of GBM and RF is seen to trace the peaks to some extent and follow similar pattern as the actuals.

Mean absolute percentage error obtained from all the machine learning models along with ensemble models is further analysed and a comparison is done for

Table 2: Mean Absolute Percentage Errors for the Ensembled Models

Forecasting Methods	*MAPE (DEVJun'06 – Dec'12)*	*MAPE (VAL Jan'13 – Jul'14)*
Traditional Univariate Forecasting Methods		
Holt's Winters	152.0%	108.5%
ARIMA	90.7%	89.5%
AR-GARCH	102.3%	88.9%
Machine Learning Techniques		
Support Vector Regression	75.4%	69%
Random Forest	65%	60%
Gradient Boosting Machine	24.3%	49.2%
Ensemble of Machine Learning Methods		
Ensemble 1 (SVR+RF)	70.6%	67%
Ensemble 2 (RF+GBM)	28.7%	44.6%

Fig. 10: Graph Showing the Actual vs Forecast for Ensemble 1 and Ensemble 2 Techniques

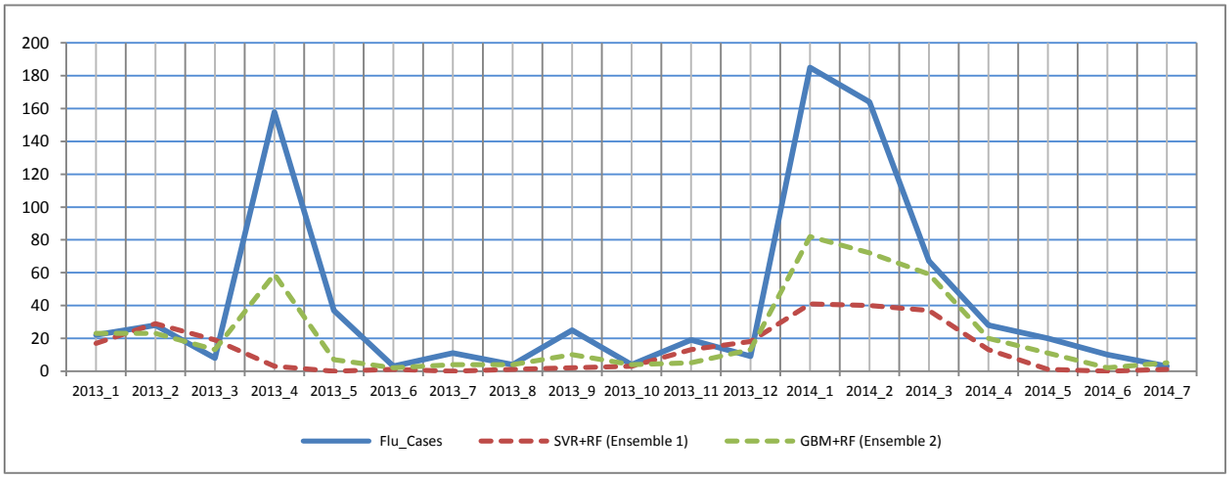
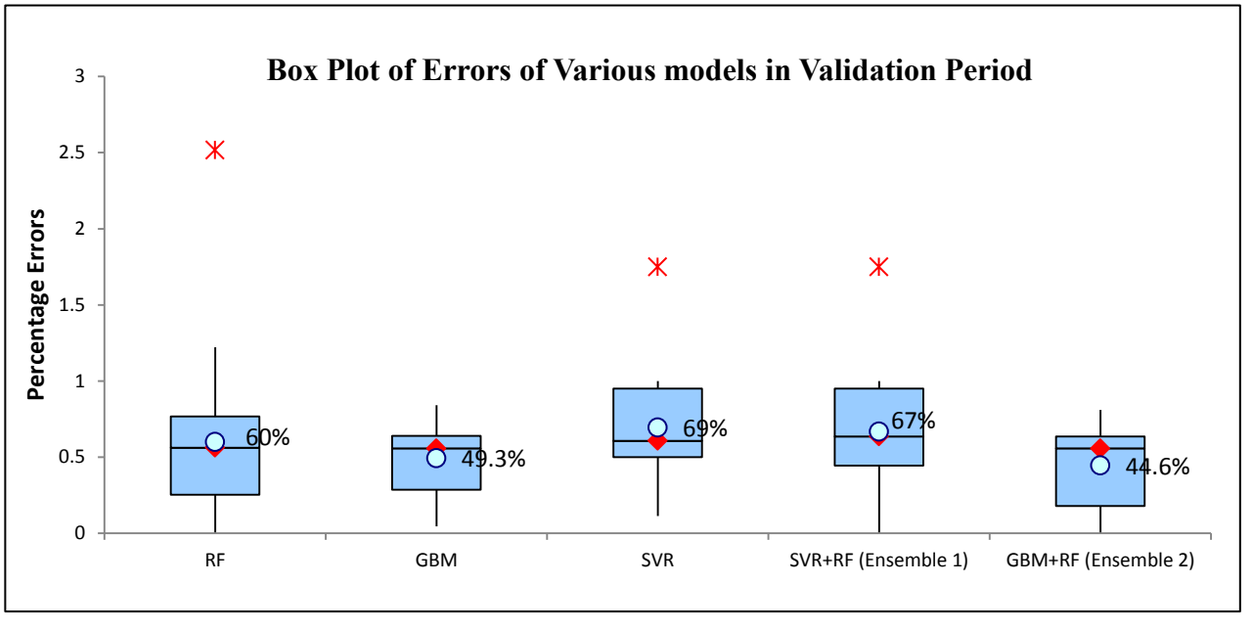


Fig. 11: Boxplot Analysis of Errors of Various Machine Learning Models and their Ensembles



the validation period (Jan'13 to Jul'14) using the boxplot technique.

Fig. 11. shows the distribution of errors for various machine learning models (RF, SVR and GBM) and their ensemble models (Ensemble 1 and Ensemble 2). It is seen that there are outliers in Random Forest, Support Vector Regression, and Ensemble 1 approaches due to which, these models do not yield a better MAPE in the validation period. In other words, these models fail to predict few data points closely. In Ensemble 2, it can be seen that the power of two models (GBM and RF) have reduced the MAPE to 44.6%, denoted by a small white dot. Also, there are no outliers for this ensemble demonstrating uniformity in error distribution and stability in the forecast.

Conclusion

Each method of forecasting has its own strength and weaknesses and hence an ensemble of these non-linear techniques tries to minimise on their shortcomings. In the present research work, ensemble of machine learning

techniques random forest and gradient boosting machine models provide an enhanced predictive ability over existing time series models (ARIMA) for the prediction of Avian Influenza outbreaks in Asian regions. This ensemble model takes advantage of each of its components random forest and gradient boosting, and recursive learning component, to generate good prediction efficiency. Also, the proposed approach is capable of handling time series prediction over a longer horizon. As next steps for its improvement additional factors could be incorporated into the predictive model. Also, more granular study for any specific country/location and incorporating the Geographic Information System to track the outbreak would improve the findings of research work further.

Acknowledgement

We would sincerely like to thank the leadership of Advanced Research and Analytics (ARA) for supporting our research work. We would also like to pay special gratitude tothe head of Instructional Design team (ARA) for helping us in figuring out relevant data for the analysis. Last but not the least, we also acknowledge the support and critical feedback of our lead time-series modeler.

References

Box, G., & Jenkins, G. (1970). *Time series analysis: Forecasting and control*. San Francisco: Holden-Day

Engle, R. F. (2001). GARCH 101: The use of ARCH/ GARCH models in applied econometrics. *Journal of Economic Perspectives,* 15(4), 157-168.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*,29, 1189-1232.

Husin N. A., Salim N., & Ahmad, A. R. (2008). Modeling of dengue outbreak in Malaysia: A comparison of Neural Network and Nonlinear Regression Model. *Proc. International Symposium in Information Technology*, 1-4.

Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shin, H. H., Zheng, Q., Yen, N. C., Tung, C. C., & Liu H. H. (1998).The empirical mode decomposition method and the Hilbert spectrum for non-stationary Time Series Analysis. *Proc Roy Soc London A,* 454, 903-995

Kane, M. J., Price, N., Scotch, M., & Rabinowitz, P. (2014).Comparison of ARIMA and Random Forest time series models for prediction of avian influenza H5N1 outbreaks. *Kane et al. BMC Bioinformatics,* 15, 276

Liu, Y. P., Wang, Y., & Wang, Z. (2013). RBF prediction model based on EMD for forecasting GPS precipitable water vapor and annual precipitation. *Advanced Materials Research*, 765, 2830-2834.

Labate, D., Foresta, F. L., Occhiuto, G., Morabito, F. C., Lay-Ekuakille, A., & Vergallo, P. (2013). Empirical mode decomposition vs. wavelet decomposition for the extraction of respiratory signal from single-channel ECG. *A comparison. Sensors Journal, IEEE,* 13(7), 2666-2674.

Wu, M. C., & Hu, C. K. (2006). *Empirical mode decomposition and synchrogram approach to cardiorespiratory synchronization*. Phys. Rev. E **73,** 051917

R Core Team: R: (2014). *A language and environment for statistical computing.*Vienna: R Foundation for Statistical Computing. Retrieved from http://www.Rproject.org/ISBN 3-900051-07-0

Scholkopf, B. (1997). *Support Vector Learning*. R. Oldenbourg Verlag, Munich

The United Nations Food and Agricultural Organization: EMPRES-iGlobal Animal Disease Information System. Retrieved from http://empres-i.fao.org/eipws3g/

the National Centers for Environmental Prediction: Global Weather Data for SWAT. Retrieved from http://globalweather.tamu.edu/

Vapnik, V. (1995). *The nature of statistical learning theory*. Springer-Verlag, New York

Appendix A

Code for forecasting monthly Avian Influenzausing Traditional techniques like Holt's Winter, ARIMA, AR-GARCH

```
#########Tile: Flu Forecasting using Traditional techniques like Holt's Winter, ARIMA, AR-GARCH

# Removes the junk variables from the memory

rm(list=ls())

#########loading required packages for Analysis

library(TSA);library(tseries);library(forecast);library(fGarch);library(PerformanceAnalytics);library(rmgarch)

#########Setting working directory

setwd("/Users/desktop/DATASET/Research")

#########Reading Dataset

flu<- read.csv("flutrain.csv")

flu<- flu[,4]

hist(flu)

hist(exp(flu))

length(flu)

# Converting Data to time series object

t_data<- ts(flu$Flu_Cases[1:79], frequency=12, start=c(2006,6))

#Holt Winter

holt<- HoltWinters(t_data)

holt_forecast<- forecast.HoltWinters(holt, h = 19)

holt_forecast

plot(holt_forecast)

holt<- as.numeric(holt_forecast$mean)

write.csv(round(holt), file="Holt.csv")

#########creating reg variable to have intercept

t <- seq(1,79,1)

nt<- seq(80,98,1)

str(flu)

#transforming time series.

flue<- ts(sqrt(flu),frequency=12)

class(flu)

# creating Arima using Auto.Arima

flu.auto<- auto.arima(flu)

adf.test(flu)

adf.test(flu,k=12)

adf.test(diff(flu,lag=12))

adf.test(diff(diff(flu,lag=12)))

acf(flu,lag.max=200)

pacf(flu,lag.max=200)

acf(diff(diff(flu,lag=12)))

eacf(diff(diff(flu,lag=12)))

#craetingArima model manually

flu.arima<- arima(flu,order=c(0,0,1),

seasonal=list(order=c(0,1,1),period=12),xreg=t)

fluarpreA<- predict(flu.auto,n.ahead=19)

fluarpre<- predict(flu.arima,n.ahead=19,newxreg=nt)

#writing csv file

write.csv(fluarpre$pred,file="ffores.csv")

write.csv(fluarpreA$pred,file="fforea.csv")

plot(residuals(flu.arima),type="p")

res<- residuals(flu.arima)

resa<- residuals(flu.auto)

Box.test(resa)

resasq<- resa*resa

Box.test(res)

Box.test(resasq)
```

```
resq<- res*res

Box.test(resq)

plot(resq,type="l")

acf(resq)

pacf(resq)

eacf(resq)

eacf(resasq)

#Aarch Modeling for Residual

#spec <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = #c(1,1)), mean.model = list(armaOrder = c(0, 0), include.mean = FALSE), # distribution.model = "norm")

eacf(resasq)

#creating Garch model from residual from autoarima

resag<- garchFit(~garch(1,0),data=resa,cond.dist="norm",include.mean=FALSE)

resagp<- predict(resag, n.ahead=19)

eacf(resq)

#creating Garch model for residual of Manual Arima Model

resgp<-garchFit(~garch(2,0),data=res,cond.dist="norm",include.mean=FALSE)

resqg<- predict(resgp, n.ahead=19)

write.csv(resqg$standardDeviation,file="sd1.csv")

write.csv(resagp$standardDeviation, file="sd2.csv")
```

Appendix B

Code for forecasting monthly Avian Influenzausing Machine Learnings methods (Random Forest, Gradient Boosting Machine and Support Vector Regression)

```
rm(list=ls())

root<- "C:\\Users\\vshar50\\Documents\\Research_n_Development\\Research Papers\\DATA"

setwd(root)

library(data.table)

data<- as.data.frame(fread("Weather_Data.csv"))

t_data<- ts(sqrt(data$Flu_Cases)[1:79], frequency =12, start= c(2006,6))

library(EMD)

### Extracting the first IMF by sifting process

par(mfrow=c(2,3))

tryimf<- extractimf(t_data, check=TRUE, boundary="wave")

### Empirical Mode Decomposition

par(mfrow=c(4,3))

try<- emd(t_data, boundary="wave", plot.imf=TRUE)

# Collecting IMF

# These are the series to be forecasted

imf1= try[[1]][1:98]

imf2= try[[1]][99:196]

imf3= try[[1]][197:294]

imf4= try[[1]][295:392]

error= try[[2]]

training<- cbind(data[1:79,5:ncol(data)], error[1:79])

names(training)<- c(names(data)[5:ncol(data)], "error")

testing<- cbind(data[80:98,5:ncol(data)], error[80:98])

names(testing)<- c(names(data)[5:ncol(data)], "error")

library(caret)

# setup learning method

require(randomForest)

library(doParallel)

# try the random forest fit

# using parallel computation if available

set.seed(9)
```

```
rfGrid = expand.grid(mtry = c(3,5,7,9,11,15,20))
cluster<- makeCluster(detectCores())
registerDoParallel(cluster)
# applies for each classification or regression fit
fitControl<- trainControl(
method = "repeatedcv",
number = 5,
repeats = 5,
classProbs = FALSE,
verboseIter = TRUE,
preProcOptions=list(thresh=0.95,na.remove=TRUE,verbose=TRUE),
seeds = NA,
allowParallel = TRUE
)
paste(names(training),collapse="+")
##############################################
###
# Forecasting IMFs 1st , 2nd, 3rd, 4th and error using Random Forest
cluster<- makeCluster(detectCores())
registerDoParallel(cluster)
#Random Forest Code
fit.raf<- train(error~<List of Variables>,
data=training,
method="rf",
preProcess=c("center","scale"),
tunelength=10,
tuneGrid = rfGrid,
trControl=fitControl,
ntree = 1000,
importance=TRUE,
metric="RMSE")
stopCluster(cluster)
predicted.raf<- predict(fit.raf,newdata=testing)
fitted.raf<- predict(fit.raf,newdata=training)
##############################################
##############################
#Support Vector Machine
library(e1071)
cluster<- makeCluster(detectCores())
registerDoParallel(cluster)
svm<- train(error~<List of Variables>,
data=training,
method = "svmPoly",
trControl = fitControl,
preProc = c("center", "scale"),
tuneLength = 10,
metric = "RMSE")
stopCluster(cluster)
predicted.svm<- predict(svm,newdata=testing)
fitted.svm<- predict(svm,newdata=training)
##############################################
##############################
#Gradient Boosting Machine
set.seed(9999)
cluster<- makeCluster(detectCores())
registerDoParallel(cluster)
gbmFit<- train(error~<List of Variables>,
data = training,
method = "gbm",
```

```
trControl = fitControl,
verbose = FALSE,
## Only a single model can be passed to the
## function when no resampling is used:
tuneGrid = data.frame(interaction.depth = 5,
n.trees = 500,
shrinkage = .1),
metric = "RMSE")
stopCluster(cluster)
predicted.gbm<- predict(gbmFit,newdata=testing)
fitted.gbm<- predict(gbmFit,newdata=training)
#Predicted Residue
par(mfrow=c(1,1))
pres<- error[80:98]- (predicted.raf)
plot(predicted.svm, type="l")
plot(error[80:98], type="l")
plot(error, type="l")
#Predicted and Fitted
raf<- c(fitted.raf,predicted.raf)
svm<- c(fitted.svm,predicted.svm)
gbm<- c(fitted.gbm,predicted.gbm)
final<- cbind(data$Key,error, raf,svm,gbm)
write.csv(final,"LogError.csv", row.names=FALSE)
#Storing the Predicted IMF's
imf1_pred<- predicted.raf
imf2_pred<- predicted.svm
imf3_pred<- predicted.svm
imf4_pred<- predicted.gbm
error_pred<- predicted.raf
final_series=imf4_pred+imf3_pred+imf2_pred+imf1_
pred+error_pred
final_series<- final_series^2
final_series1=ifelse(final_series<0,0,round(final_series))
flu=imf1+imf2+imf3+imf4+error
flu<- data$Flu_Cases[80:98]
abs_error= abs(final_series1-flu)
MAPE= mean(abs_error/flu*100)
```