STSCI 5999
NASA Data Mining Project 2

Xiang Chen, Yicheng Wang, Jiahui Yuan, Shuting Zhao
NetIDs: xc335, yw488, jy723, sz435

Advisors: Prof. Stanislav Volgushev, Dr. Rodney Martin
Cornell University, NASA Ames Research Center

May 8, 2016

Abstract

This project is based on the scenario that heating system in NASA Sustainability Base has raised an issue of thermal discomfort. To address this problem, early warning algorithms are developed by applying several data partition methods to different models designed. Data mining is conducted in order to detect the future occurrence of low room temperatures. Prediction results provide comparable feedbacks to the other group analyzing the same dataset with ACCEPT, which is a toolbox developed by NASA for adverse event prediction. This report summarizes project findings, and discovers opportunities for future work.

1. Introduction

1.1 Background

During the heating season in NASA Sustainability Base, cold complaints have become frequent due to the fact that the building is running cold. Facility staff have been using special index to study low temperature causing as adverse events prediction in attempt to prevent thermal discomfort. Under the circumstances, novel machine learning and early warning algorithms are developed by applying several data partition methods to different models in order to improve the performance of adverse events prediction.

1.2 Introduction to ACCEPT

ACCEPT (Adverse Condition and Critical Event Prediction Toolbox), initially used by NASA for nuclear and space applications, provides "*an open-source tool which can be used specifically for the prediction or forecasting of adverse events in time series data*". It is "*a single, unifying framework in which to compare a variety of combinations of algorithmic approaches addressing this problem"* running in Matlab Programming Environment (Martin et al.). Since another group utilizes the toolbox for adverse events prediction with the same dataset, the goal of this project emphasizes on data analysis so that results generated from other statistical software are compared to outputs produced from ACCEPT and feedbacks are delivered to the other group.

1.3 Data Descriptions

92 days with each unit of 5 minutes of room temperatures are detected. Each day's dataset contains 287 or 288 rows and 11 columns. 24*60/5 = 288

The following table contains variable descriptions provided by client.

Table 1.1 Variable Descriptions

ID	Description
A	RF1 HWS VALVE 14
B	A1 DX CAP SIGNAL
C	RSB P1 START/STOP
D	CRCP VALVE S28A
E	GWRV LOOPOUT
F	M1 AVG FLOW
G	ZONE N121 N125 AVERAGE TEM
H	S1 DPT AVG C
I	HP3 HEAT STAGE TIMER
J	N1 COOLING OFF

2. Motivation

The goals of this project are:

1. To generate prediction results comparable to the output produced from ACCEPT.

2. To develop novel methodologies by which simple models can produce decent performance.

3. To provide comprehensive comparison among different methodologies for future detection system design.

3. Methodology

In this part, the work process and the methodologies in each step will be discussed. The process producing desired key performance metrics, FAR and MDR, consists of a series of comprehensive steps. In the first step, three **data partition** methods are developed which specified a certain proportion of the dataset as **Training** dataset, another part as **Testing** dataset. In the second step we fitted a **Model** to the data set, and applied the model to testing data set. In the third step, with a proposed **Threshold Selection Criteria**, the disagreement between predicted event and observed event will be measured as the two rates: False Alarm Rates and Missed Detection Rates.

3.1 Training and Testing Dataset

For training dataset, a model format is claimed in R, where the user inputs the choice of model and the related syntax claiming the selection of parameters. R package then returns a model within which a series of coefficients corresponding specific parameters are included.

Testing dataset contains the predictors and observed response. The model derived from training data is applied to the predictors and returns a predicted response vector, which is used to compare with the observed response.

3.2 Performance Metrics

After obtaining the predicted response vector, a program is implemented to traverse through the vector and make a thorough comparison with the observed predictor vector.

With a pre-determined threshold, events are divided into four scenarios:

1. If the observed response is smaller than 68.1 (which is the ground truth established by client), and the predicted response is smaller than the threshold, we will count this event as Event A.
2. If the observed response is smaller than 68.1,and the predicted response is larger than the threshold, we will count this event as Event B.
3. If the observed response is larger than 68.1, and the predicted response is smaller than the threshold, we will count this event as Event C.
4. If the observed response is smaller than 68.1, and the predicted response is larger than the threshold, we will count this event as Event D.

After comparing the whole vectors, the two rates can be calculated as:

FAR = (Number of Events B) / (Number of Events A + Number of Events B)

MDR = (Number of Events C) / (Number of Events C + Number of Events D)

In the context of the project, the two rates can also be interpreted as:

False Alarm Rate measures how often the system sends an alarm when there is actually no alarm.

Missed Detection Rate measures how often the system does not send an alarm when there should be an alarm.

There is a trade-off between false alarm rate and missed detection rate based on the fact that:

If the threshold is higher, there will be fewer predicted responses higher than the threshold, which results in fewer events B, and lower FAR. There will also be more predicted responses lower than the threshold, as well as more events C, resulting in higher MDR.

3.3 Threshold Selection Criteria

As implied above, changes in threshold will affect the two rates, so it is important to develop a reasonable threshold selection strategy. In this project, a first comparison will be conducted between the observed response and predicted response of the training dataset. Different thresholds are experimented on the training dataset and groups of rates corresponding to the thresholds will also be generated. In the end, one threshold will be chosen from the group of thresholds.

The simple philosophy is to choose the threshold that results in the “best” results, and apply the threshold to testing dataset. However, “best” should be further specified because in

most cases, there is no threshold that can produce the lowest FAR and MDR at the same time due to the trade-off discussed in Section 3.2.

In "Introduction to ROC Curve for Medical Researchers", (Kumar & Abhaya) introduces several criteria to select threshold:

1. points on curve closest to the (0, 1)
2. Youden index
3. Minimize cost criterion

After discussion with client and study of the data, indices that share similar philosophy to the discussion above are applied to threshold selection with some imposed specification (for instance, choose the one which returns the minimum addition of FAR and MDR where MDR is below 5%). Detailed introduction will be given in next section.

3.4 Data Partition

Data Partition is one of the most important concerns of this project. A regular practice is to randomly split the data into training and testing datasets. However, due to the quality of the data, not all models could be applied in this method. As a result, three data partition methods are developed and tested so as to meet the requirements of different models. They are: Sliding Window, Hemisphere and Triangle.

3.4.1 Sliding Window

Due to the fact that observations of response variable are not independent of each other over time, time series models can be applied. But in long term, the dataset shows great non-stationary nature, which leads to the usage of Sliding Window. Sliding Window uses only a small portion of the data instead of the entire dataset, or half of the dataset, to fit the model. To

predict the future temperature within a certain interval, only the most "relevant" observations are selected. For example, if the temperature within the interval between 12:00 A.M and 1:00 A.M on Tuesday is to be predicted, only the data on Monday will be used to calculate the coefficients of the model, and data before Monday will not be used. The design of this system will continuously adjust the model by replacing past data with the most current data, and then predict the future temperature within a pre-defined interval.

To produce the key performance matrix, Sliding Window will traverse across the data. Two interval sizes will be determined before running the program: the size of the training dataset and the size of testing dataset. Then the program will run more than 25,000 iterations (approximately the data size). In each iteration, it will start from where last iteration ends and move 1 unit forward, so the first observation of the former training dataset will be discarded and the next observation to the former training data will be added, which forms a new training dataset. And testing dataset is always the next N observations to the new training dataset, where N equals the pre-determined testing size.

The advantage of this method is simplicity while still keeping the accuracy. Fitting models over thousands of times seems to be computing-intensive, but for models such as auto regression, there is only one variable, which has much lower dimension than other applicable methods. If the methodology is implemented, it will be actually computing-saving than other methods that use all past data that keeps accumulating over time. Another advantage is that the models trained in this way are more sensitive in catching the undergoing trend by using the most current dataset. It is hard to fail if the change of temperature stays small overtime.

The disadvantage of this method is that the rates calculated this way do not contain the data from Day 1, which can only be used as training data rather than testing data, making it not

so comparable with rates resulting from other methods. Another disadvantage is the models will miss the detection of the long-term trend and seasonality.

3.4.2 Hemisphere

Hemisphere refers to the idea that splitting the dataset into two parts with the same size. One half of the data is used for training dataset and the other half is used for testing dataset. There are two ways to partition the data in Hemisphere: chronological and random splitting.

The chronological splitting partitions the data in a straightforward way: using the first half of the data as training dataset and the second half as testing dataset. However, in this project, the second half is chosen for training with the first half for testing. The reason why the partition is carried out this way is that there are more than 90% adverse events contained in the first half. Hence fitting the model to the first half may result in over-optimized result.

Another method partitioning the data into halves is random splitting, which shuffles the dataset with the constraint that keeping the number of adverse events in each dataset equal. In other words, both of the shuffled datasets include half of data which contains only adverse events and half of data which contains only non-adverse events. The advantage of random splitting is in accordance with how it is defined: the ratios between adverse and non-adverse events within testing and training datasets are relatively more “balanced” than chronological splitting. However, the cost is also obvious in that the training data may already contain the information of the testing data, undermining the effect of data partition.

3.4.3 Triangle

This method is mainly developed for producing comparable result with ACCEPT by applying similar data partition method adopted by ACCEPT. In this method, data is split into three sets: training, validation and testing. Training dataset contains only the data from days

where there is no adverse event, whereas testing dataset contains data of 2 days out of 12 days containing adverse events and validation dataset contains the rest 10 days. In this method, training dataset and validation dataset are combined together to train the model and only validation dataset is used to choose threshold, so that the method ensured enough dataset to train the model while the threshold is not chosen in a dataset with much more non-adverse events than adverse events.

The advantage of this method comes from the constraint imposed on testing dataset so that only days with adverse events are used for testing, preventing those data of the 80 days without adverse events from being involved in the testing process and producing over-optimized rates.

The disadvantage of this method also comes from the testing data. Since only two days are selected, the results produced in this way highly depend on the similarity between validation dataset and testing dataset. A light difference might lead to ridiculous results.

3.5 Model Introduction

In this project, four categories of models are introduced: linear models, classification models and non-parametric models.

In linear models, the response variable is still continuous and is manually converted to 0 and 1 later (by comparing to the threshold), whose advantage is in catching the trend of change of response variable.

In classification models, the response variable is first converted to binary variable and then the model is fit to the data, whose advantage is in completely separating those below the threshold from those above it.

In non-parametric models, the response variable can be either one, whose advantage is in making full use of the predictors which contain only several levels of value and are not helpful in the models above.

4. Models and Results

4.1 Sliding Window

4.1.1Time Series

The objective of this project is to improve the early warning algorithms to predict future adverse events. Time series analysis may help because it selects models to approximate up and down trends of a time series, and then forecasts future values based on the captured pattern of the observed data.

In time series analysis of this project, only response variable, room temperature, is involved and other predictors are left out. This means that appropriate time series models are applied to past y to predict future y.

One assumption of utilizing time series models is stationarity, which means that the time series' statistical properties such as mean, variance, autocorrelation, are constant over time. This assumption is tested by unit root test. Response variable of the whole dataset, which is over 25,000 observations, is not stationary. In addition, using past several months' temperatures to predict temperatures in a short time does not make much sense in reality. Therefore, time series models cannot be applied directly to the whole dataset. However, most randomly selected short period (for example, 24 hours) time series within the whole dataset are stationary. Hence sliding window is applied to obtain training and testing intervals because their sizes are pretty flexible under this data partition method.

There are several time series models, including AR, MA, ARCH. It shows that the observed temperature does not have a clear repeating pattern, so seasonality is not considered. In order to see whether time series models perform well, AR model (autoregressive), one of the simplest time series models is firstly tested, and it generates relevantly good results. Then more complicated models such as ARIMA (autoregressive integrated moving average) and GARCH (generalized autoregressive conditional heteroscedasticity) are experimented.

When the predicted temperature is below the manually selected threshold, an adverse event occurs and an alarm should be set. This threshold makes forecasted temperature a binary variable. When the predicted and true binary events are known, false alarm rate (FAR) and missed detection rate (MDR) are calculated based on the formula mentioned in 3.2. The lower these two rates are, the better one model is.

4.1.1.1 AR

Autoregressive models forecast the variable of interest using a linear combination of past values of that variable, which means that it is a regression of the variable against itself.

An autoregressive model of order p (AR(p) model) can be written as:

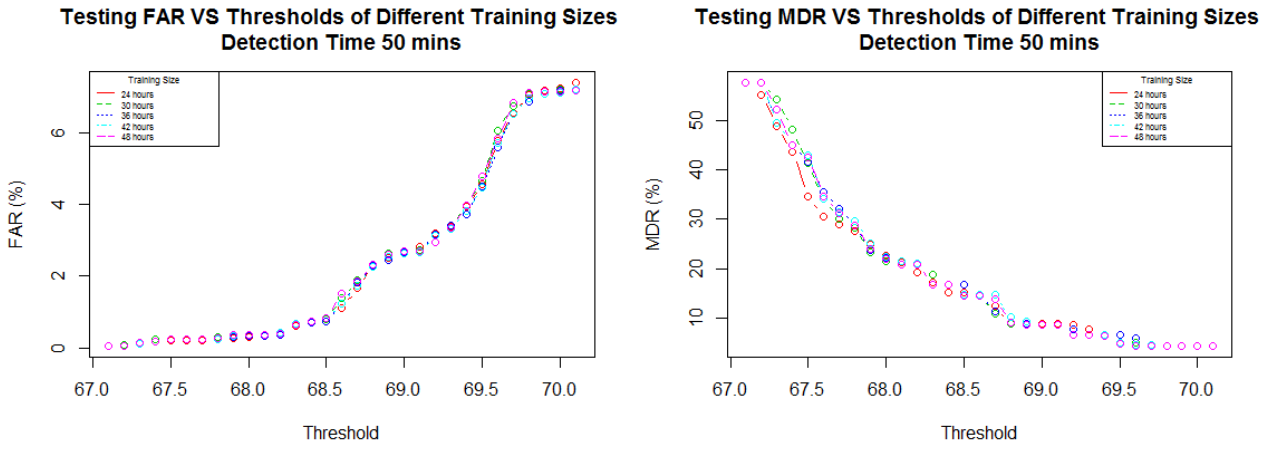
$$y_t = c + \emptyset_1 y_{t-1} + \emptyset_2 y_{t-2} + \cdots + \emptyset_p y_{t-p} + e_t$$

where c is a constant and e_t is white noise. $\emptyset_1$ is the coefficient of term y_{t-1}. y_{t-1} is the value of y when time is at t-1.

The sliding window moves one unit (5 minutes) forward each time. Since there are more than 25,000 observations, each model runs more than 20,000 times. 31 thresholds from 67.1 to 70.1 with increment of 0.1 are applied to generate different FAR and MDR on testing interval to reveal the relationship between threshold and two rates. Before determining order p, training and testing sizes need to be selected. Then R auto selects the AR order for each training dataset to

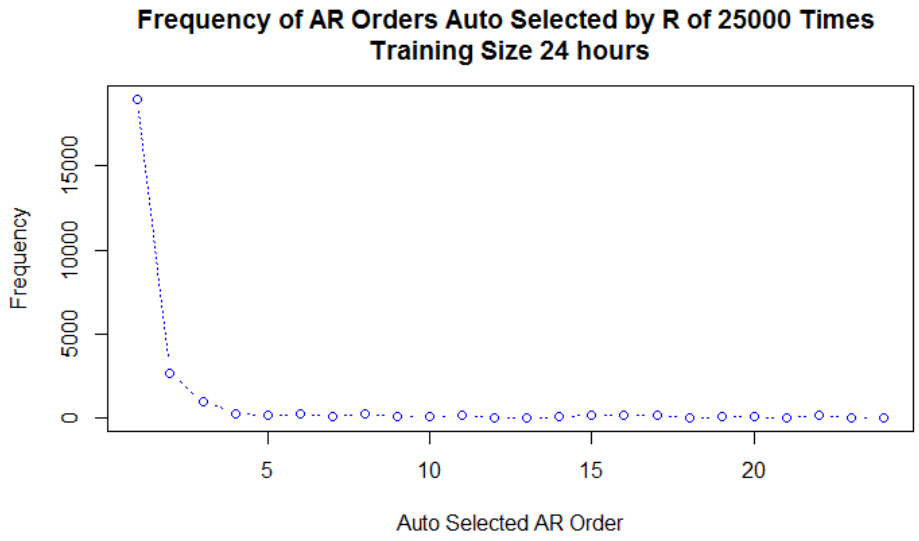
forest future temperature. Different training sizes are tested firstly: 3 hours, 6 hours, 12 hours, 24 hours, 30 hours, 36 hours, 42 hours, and 48 hours. When this size is below 1100 minutes, some training datasets cannot generate results because all the response variables within it are either adverse events or non-adverse events. Therefore, only results when training sizes are 24 hours, 30 hours, 36 hours, 42 hours, and 48 hours are compared. When testing size is larger than 50 minutes, the forecasted temperatures tend to converge, meaning that FAR and MDR do not follow a clear trend and they are results simply based on random guesses. So testing size (detection time) is fixed at 50 minutes (10 unit lags). FAR and MDR of 5 different training sizes at all the 31 thresholds are very similar (See Figure 4.1). Therefore, the smallest, 24-hour-long training interval size (288 observations) is chosen because the smaller the interval is, the more efficient the model can run.

Figure 4.1



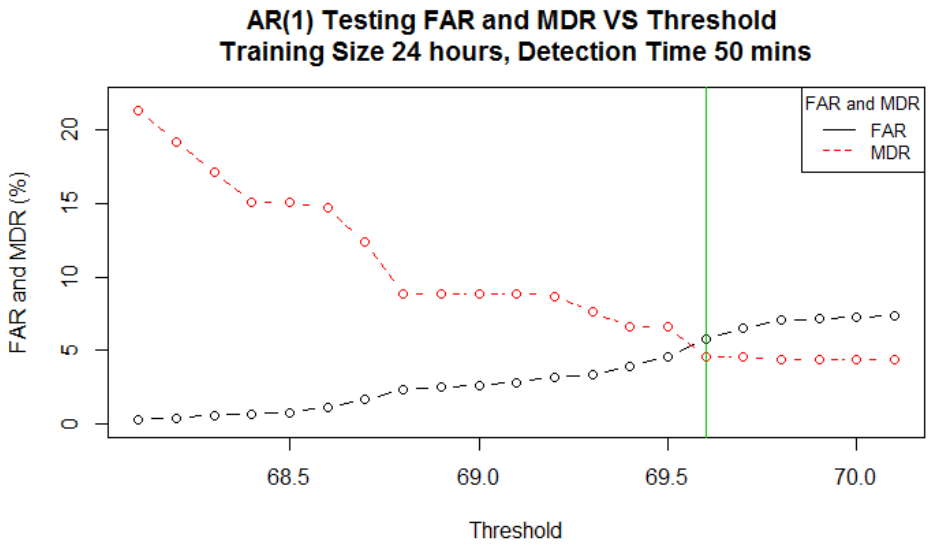
R auto selects order p in the previous steps, but a fixed p needs to be chosen for the convenience of future prediction. Most of the auto-selected orders are 1 (See Figure 4.2).

Figure 4.2



Thus, order 1 is selected and the results of AR(1) are generated (See Figure 4.3).

Figure 4.3



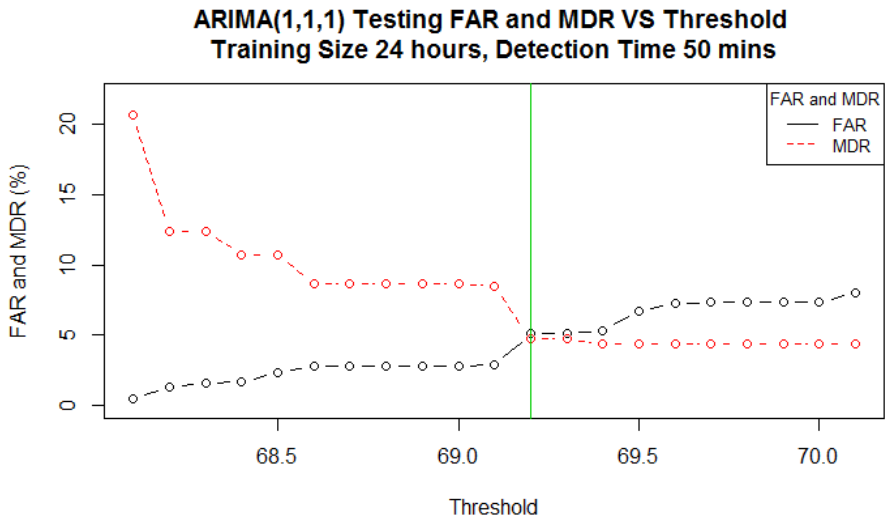
As detection time increases, both of the two rates increase because there is more uncertainty. When threshold gets larger, FAR gets larger but MDR gets smaller. Increasing threshold increases the probability to detect an adverse event, so FAR becomes higher, while MDR becomes lower. If threshold is set to extremely large, MDR will be close to zero, while

FAR will get relatively high. When threshold is larger than 69.8, two rates intend to change very slightly. It is suggested that the optimal threshold should be defined to keep the two rates the same because a tradeoff exists between them. For AR(1) model, 69.6 is the optimal threshold (green line on Figure 3 indicates this). AR with higher orders (2 and 3) are also tested in order to see whether they are better than AR(1), but their results are not far from those of AR(1). Therefore, AR(1) is the preferred autoregressive model among all the AR models.

4.1.1.2 ARIMA

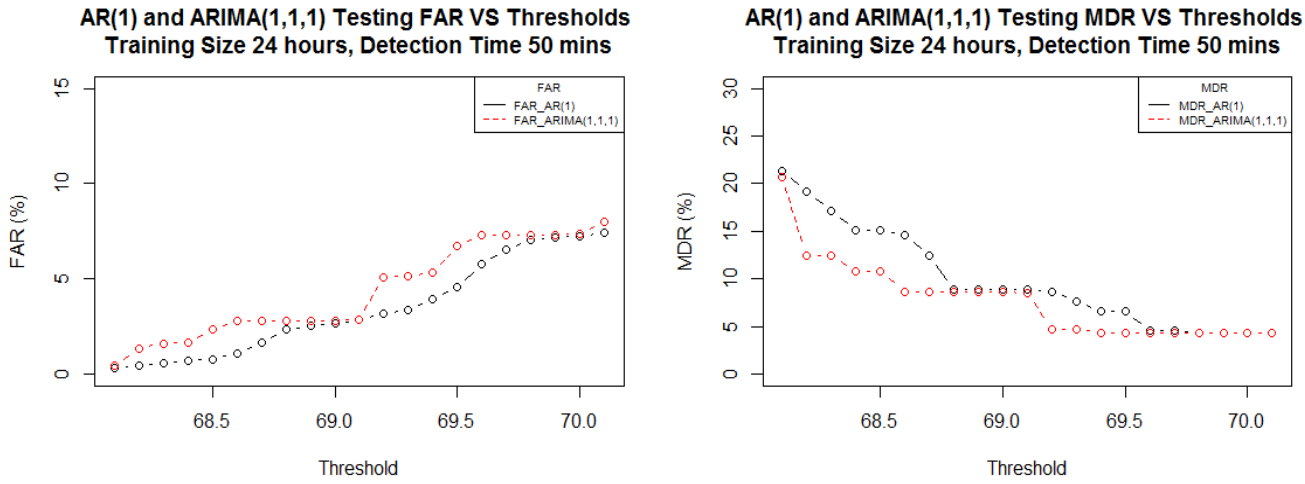
Autoregressive integrated moving average model is a generalization of ARMA (autoregressive and moving average) model. It is denoted as ARIMA(p, d, q), where p is the order of AR, d is the degree of differencing, and q is the order of MA. The same methodology is applied to select training and testing sizes, model orders, and thresholds for ARIMA model as that of AR. Training and testing sizes keep the same, while model orders are p = 1, d = 1, and q = 1 and optimal threshold is 69.2. FAR and MDR follow the same trends as those of AR as threshold increases. The detailed results are shown in Figure 4.4.

Figure 4.4



Fixing the detection time to be 50 minutes (or other time below 50 minutes), the performances of AR(1) and ARIMA(1,1,1) are similar (See Figure 4.5). The parsimonious model, AR(1) is therefore preferred.

Figure 4.5



4.1.1.3 GARCH

Generalized autoregressive conditional heteroscedasticity is applied when the error terms are thought to have a variance. Orders auto-selected by R generate results that are much worse than those of AR and ARIMA. Therefore, this model is not considered for further exploration.

Overall, AR(1) is selected among all the time series model experimented. Its results will be compared with other models later.

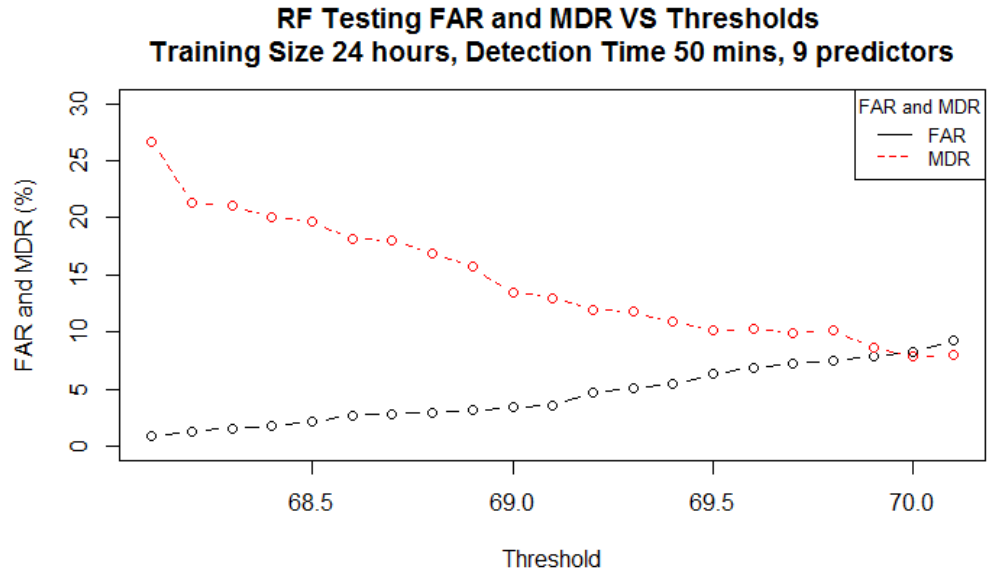
4.1.2 Random Forest

For random forest (RF), predictors instead of response variable are applied to predict future temperatures. RF randomly selects a pre-set fixed number of predictors for forecasting, and it is unknown which predictors it has chosen. RF is considered as a “black box”

and it is difficult to clarify a model for future prediction. Although not being the candidate for adverse event prediction, it is worth trying as it shows how good the FAR and MDR can be when using X to predict Y. Here current X predicts current Y (no lags for predictors) because just a general idea on the RF's performance is needed.

Different numbers of predictors from 6 to 9 are tested. In order to make the results comparable to those of time series, training size is set to 24 hours and detection time is set to 50 minutes. Thresholds range from 68.1 to 70.1. The detailed results are in Figure 4.6, which indicates that when threshold increases, FAR increases but MDR decreases.

Figure 4.6



Fixing detection time to be 50 minutes at optimal threshold of AR(1) 69.6, number of predictors does not affect much on the results of random forest (See Figure 4.7).

Figure 4.7

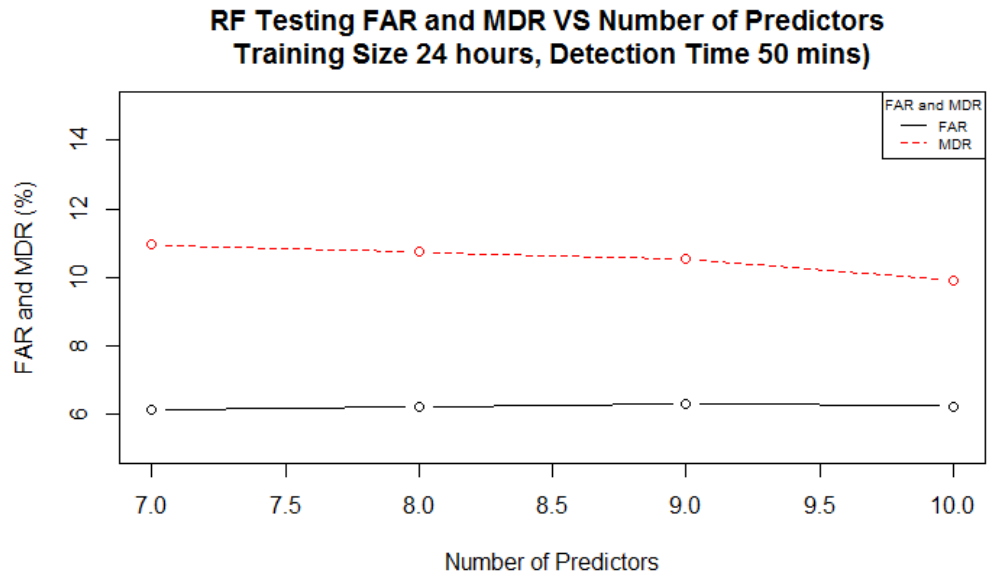
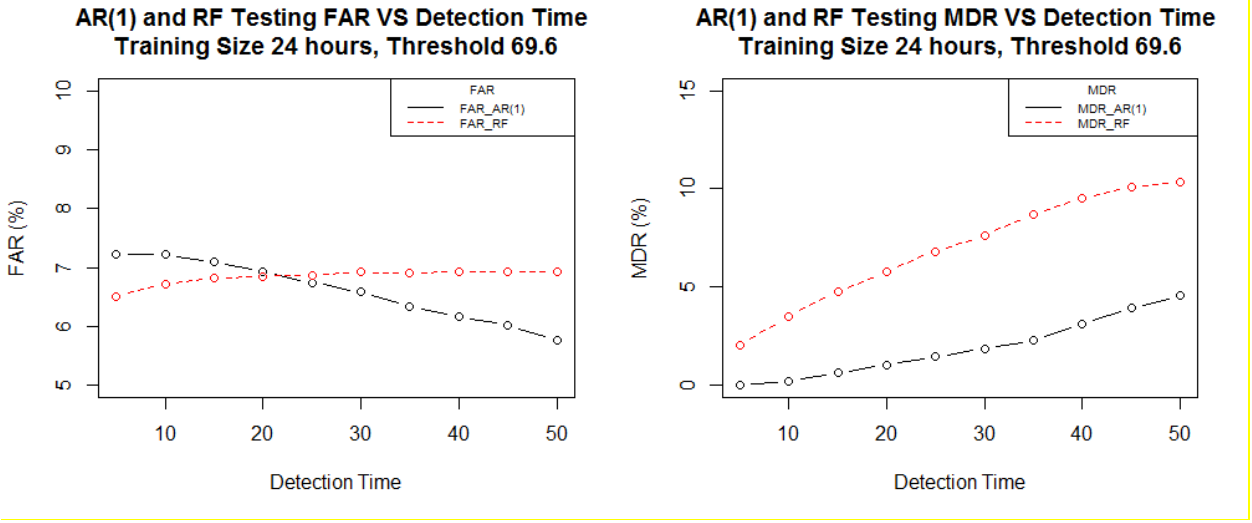


Figure 4.8 compares the performances of AR(1) and random forest. FAR of these two models are close to each other, but MDR of AR(1) is much better than that of random forest. Past temperatures are probably more powerful to predict future adverse events than X predictors.

Figure 4.8

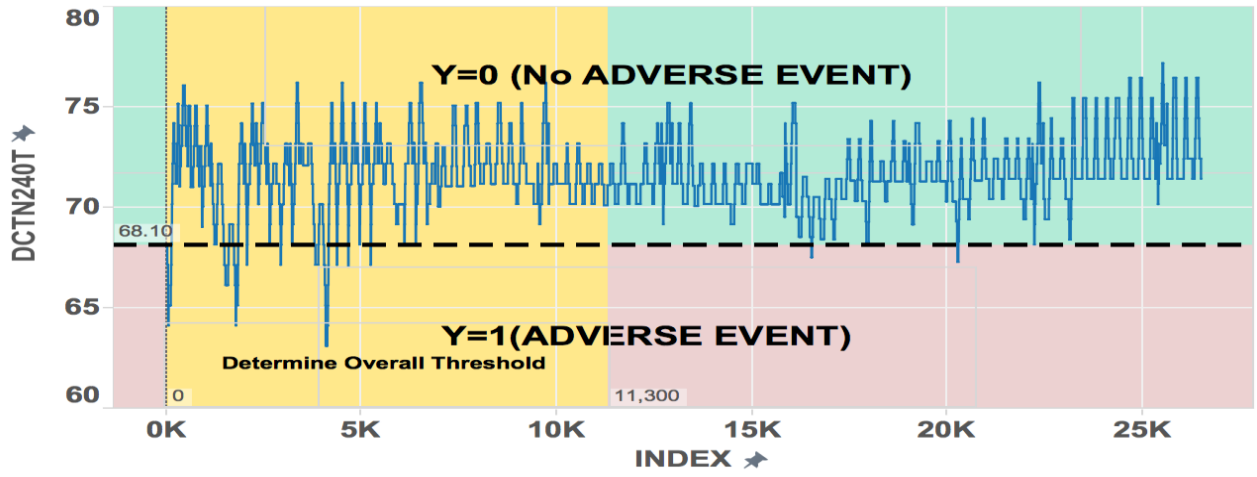


4.1.3 Logistic Regression

There are generally two ways to predict adverse events. The most common way is to firstly fit a model such as time series model mentioned before and then categorize the predicted value, in our case the predicted temperature above and below manually selected threshold, and compare it with the actual classified response variable. The other way is to make the response variable firstly be classified into categories, then fit a model, and finally compare results versus the actual classified responses. Logistic regression, one of the most common ways to deal with dataset with categorical dependent variable, is a typical example and it is discussed as follows.

To begin with, the continuous response variable is firstly converted into binary variable with '1' for adverse events ($y < 68.1$) and '0' for non-adverse events ($y >= 68.1$). The following graph shows that the longest interval between two adverse events is around 11000 observations. Since logistic regression cannot be fitted when there is no adverse events presented in the dataset, the minimum window size is therefore set to 11300 observations in this case.

Figure 4.9



The graph also illustrates the classification method used to prepare the data and window size (Yellow Field). After the window size and overall methodology are determined, variable selection is performed. To simplify the model and avoid overfitting, only variables that are

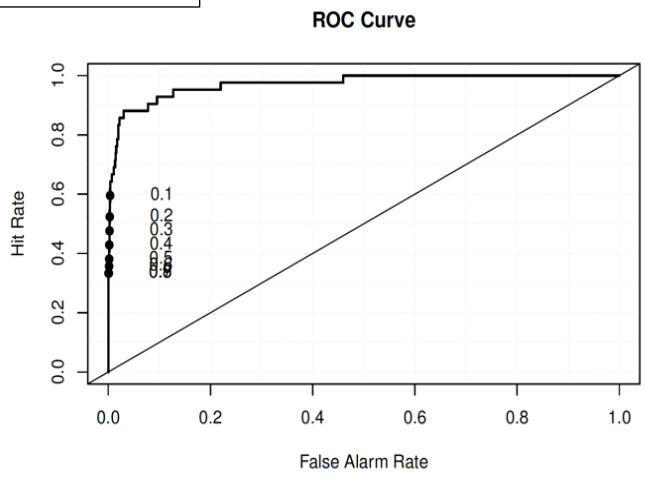
significant (p value < 0.05) in the fitted model of the first window are kept. The final logistic model is

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 D_{t-10} + \beta_2 G_{t-10} + \beta_3 H_{t-10} + \beta_4 y_{t-10} + \beta_5 I y_{t-10}$$

The window moves one unit (5 minutes) forward each time to run the model. Overall, the loop runs approximately 15000 times. Detection time is set to 50 minutes (10 unit lag). Manually selected threshold is one that makes FAR and MDR almost the same.

Upon completion, the ROC curve is generated by comparing the actual and predicted binary values on different thresholds. Figure 4.10 shows the ROC curve for different thresholds at 50 mins' detection time. This curve shows that FAR and MDR based on previously selected threshold are 2.79% and 14.29%, which are still high. One of the possible reasons is that the true optimal threshold might fall out of the ideal threshold selection region. Also, it is possible that logistic regression is not a favorable model in general for sliding window data partition method because such methodology is generally used to catch the trend of the continuous data. If the response variable is converted into binary, the information as well as the trend of the dataset are lost.

Figure 4.10



Considering the huge amount of time involved, only the results for detection time 5 minutes and 50 minutes are generated. For detection time 5 minutes, both FAR and MDR are close to 0%, yet since the detection time is too short, the result is not helpful here.

4.2 Hemisphere

4.2.1 Random Sampling

4.2.1.1 Logistic Regression

Before applying Logistic Regression to the Hemisphere – Random Sampling Splitting Method introduced in the previous chapter, the response variable, room temperature, is first converted into binary while all the predictors were "lagged", which forms a new dataset with Iy_t (indicator of response variable at time t) and predictors $X_{t-\tau}$ (predictors at τ units ago). Since the other ten variables show low correlation with the response variable and do not provide too much help in prediction of response, past data of response variable $Y_{t-\tau}$ is also added to the predictors to improve the performance of the model. After the preparation is finished, Hemisphere – Random Sampling partition method is performed to produce training and testing dataset. And the logistic regression is applied on training dataset.

Variable selection is then conducted to apply the most significant predictors to the model and avoid overfitting. However, after numerous experiments, different variables are selected depending on the value of τ, which means there are no constant best variables. In addition, changing the number of variables does not have significant influence on FAR and MDR. Therefore, most of predictors are not removed. Also, though suggested to remove in other models, some variables are proved to have a good performance in logistic model for random sampling. Overall, the logistic model used in Random Sampling splitting method is

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 A_{t-\tau} + \beta_2 B_{t-\tau} + \beta_3 D_{t-\tau} + \beta_4 E_{t-\tau} + \beta_5 F_{t-\tau} + \beta_6 G_{t-\tau} + \beta_7 H_{t-\tau} + \beta_8 J_{t-\tau} + \beta_9 y_{t-\tau} + \beta_{10} Iy_{t-\tau}$$

When the coefficients for the logistic model are determined after fitting on training dataset, the fitted response is compared against the observed response in the training dataset to

determine the threshold. According to Section 3.3, there are three popular indices. In the context of this project, lower MDR is given more weight than FAR, while the three indices introduced above are valued equally, which is not an applicable feature in the project. As a result, a new index can be proposed: choosing the threshold returning the minimum MDR in training dataset, which can also lead to a low MDR in testing dataset if the nature of training dataset and testing dataset are similar, which can be assumed in Hemisphere – Random Sampling splitting method. An R function which is designed to efficiently determine the optimal threshold based on our 'optimal' criteria is then produced:

autoselectlog(Y, $\hat{Y}$, MAX, MIN, $Increment$, Z)

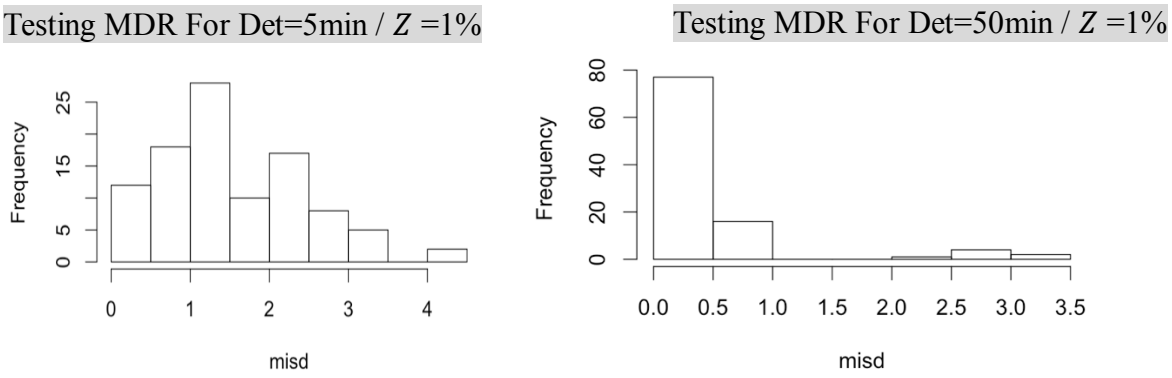
(For Binary Response Only)

Table 4.1

Arguments	Explanation
Y	Actual Value Y
$\hat{Y}$	Fitted Value Y
MAX	Upper bond for Threshold (UB)
MIN	Lower bond for Threshold (LB)
$Increment$	Threshold Increment Each Time from LB to UB
Z	The MAX allowed misdetection rate for training dataset

However, even though restricting MDR for the training dataset can effectively control MDR for the testing dataset, the power of such mechanism decreases as the detection time decreases. Here are the testing dataset MDR for detection time set from 5 minutes ahead and 50 minutes ahead when setting Z =1% and simulating the process for100 times in random sampling.

Figure 4.11



Consequently, the definition of "best" threshold varies as the detection time changes. Thresholds resulting in lower MDR on training dataset are preferred if the temperature to be predicted is more close to current time, and vice versa. This principle is constantly applied when producing the final outputs for all models in Hemisphere Random Sampling.

4.2.1.2 Linear Regression

To evaluate the performance of Logistic Regression in Hemisphere – Random Sampling Method, multiple linear regression, as the most common regression method, is also performed and the results are compared in the later section.

Similar to the Logistic Model in the previous part, all X predictors are "lagged" and thus become $X_{t-\tau}$. Similarly, in order to produce a robust model, $y_{t-\tau}$, $y_{t-\tau-1}$…..$y_{t-\tau-10}$ is added as the predictors to predict y_t. As the dataset is ready for next step, Hemisphere – Random Sampling partition method is again performed to produce training and testing dataset. And the Linear regression is applied to training dataset for variable selection.

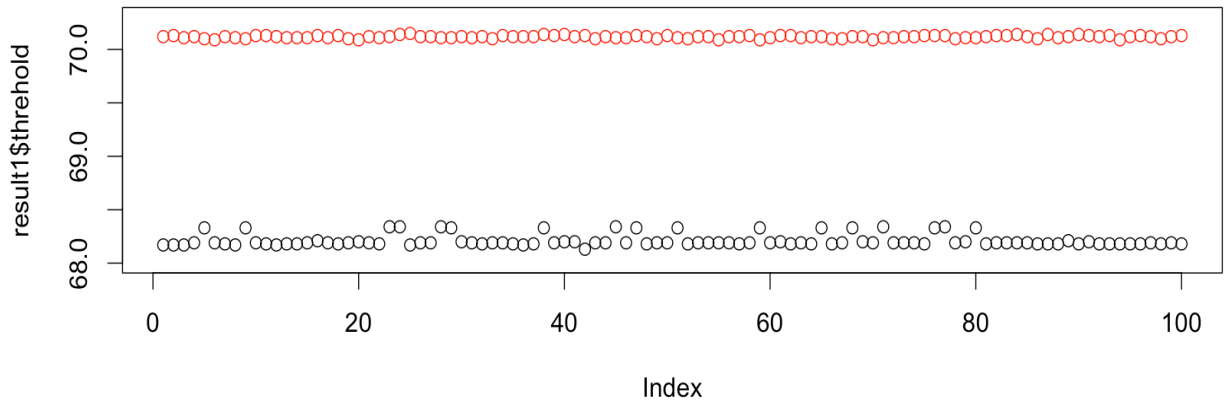
Adding more lagged terms of y such as $y_{t-\tau-1}$…..$y_{t-\tau-10}$ to our predictors is expected to produce better results in the model. However, it turns out that none of them are significant except $y_{t-\tau}$. This is because our response variable as well as other predictors remain constant from time

t-τ-1 to t-τ-10. Thus, keeping only $y_{t-\tau}$ and dropping all other past responses could help to prevent overfitting of the training dataset. In addition, since the other predictors are less significant in our model, the interaction terms as well as transformations are not further considered. To demonstrate, our final linear regression model used in Random Sampling splitting method is

$$\hat{Y}_t = \beta_0 + \beta_1 B_{t-\tau} + \beta_2 D_{t-\tau} + \beta_3 F_{t-\tau} + \beta_4 G_{t-\tau} + \beta_5 H_{t-\tau} + \beta_6 y_{t-\tau}$$

For the optimal Threshold Selection on linear regression, the R function *'autoselectlog'* mentioned in previous section for binary response is modified into '*autoselectcon'*, which can be used for numeric response variable. The nature of the function remains the same, while the running time is much longer as the optimal threshold appears to be significantly higher when detection time becomes longer. Therefore, picking the right lower bond of threshold for different detection time is able to optimize the program running time. The below graph shows the optimal threshold generated for detection time 50 (Red Dots) and detection time 5 (Black Dots) with 100 simulations in random sampling ($Increment$=0.01). Meanwhile, the minimum MDR and detection time is still positively related in linear regression for producing optimal threshold. However, such relation is mitigated compare to that of logistic regression in random sampling.

Figure 4.12

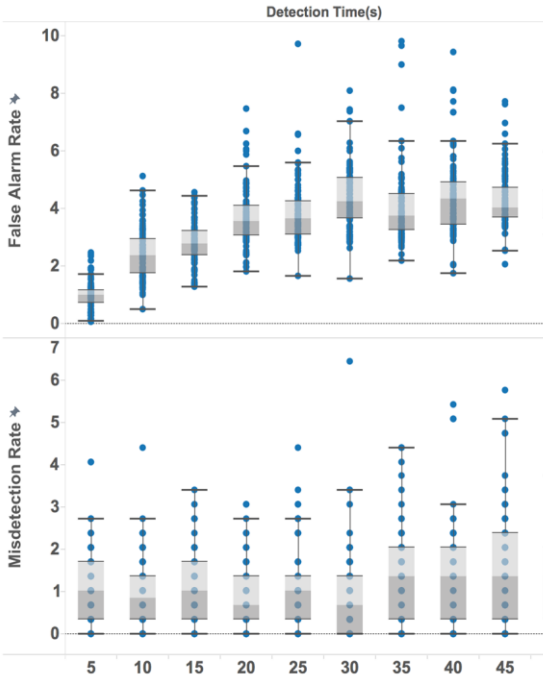


4.2.1.3 Results Comparison

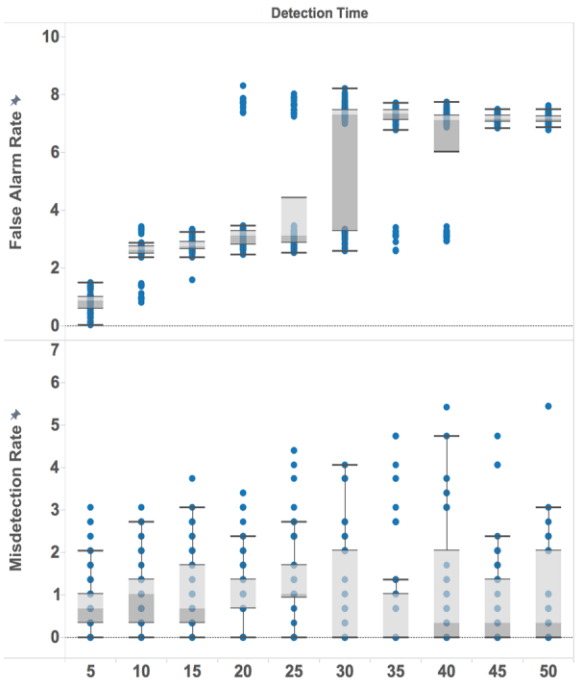
To compare the performance between logistic and linear regression in Hemisphere – Random Sampling partition method, two box plots of our results are shown below when simulating the process for 100 times for each detection time unit. The x-axis is the detection time while the y-axis shows the corresponding FAR and MDR. As the detection time becomes longer, FAR are increasing while MDR are stable for both models. This shows the effectiveness of our mechanism in controlling MDR on testing dataset by restricting MDR on training dataset.

Figure 4.13

Boxplot for Logistic Regression Results



Boxplot for Linear Regression Results



Overall, Logistic Regression performs better than Linear Regression not only because the former model produces better FAR and MDR for each detection time, but also because of the stability of the Logistic Regression performance that can be observed from the boxplot.

The below graph is another look of the performance comparison between two models in Hemisphere – Random Sampling partition method, with the detection time fixed to 50 minutes. The logistic regression performs better than the linear regression, and there is a clear cut-off line for the FAR between both models. The mean simulation results of both models are also presented in the below table.

Figure 4.14

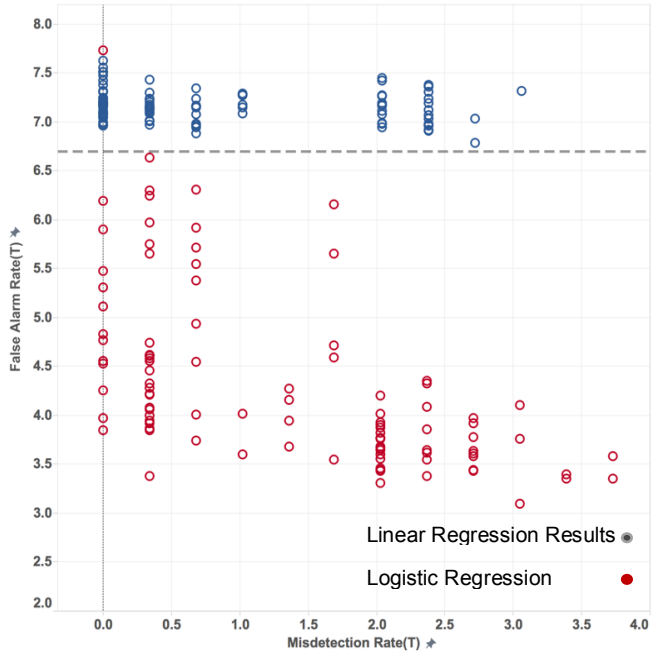


Table 4.2

Mean Simulation Results			
Model	DET	MDR	FAR
Logistic	50 mins	1.36%	4.31%
Linear	50 mins	0.93%	7.16%

4.2.2 Chorological Splitting

4.2.2.1 Logistic Regression

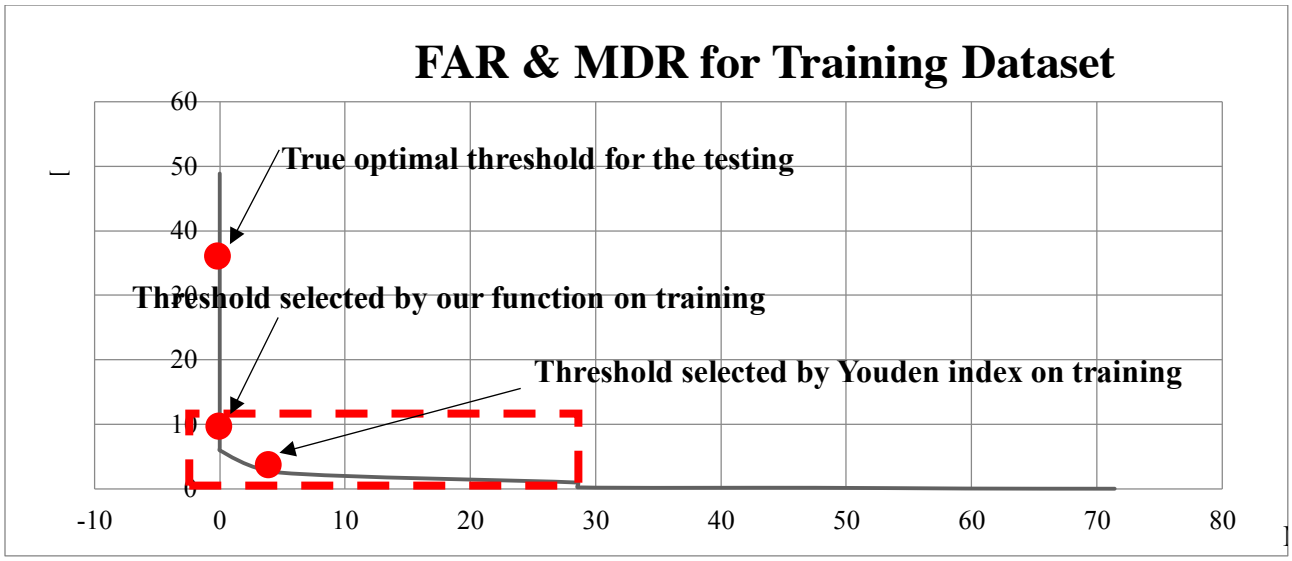
The process for data preparation in logistic regression Chorological Splitting is the same as that in Random sampling, yet the variable selection process is very different. Considering the data property, which the training dataset only consists 7% of the total adverse event observation.

Due to the small amount of adverse events, variables are selected more strictly than random splitting to avoid overfitting. Therefore, even though the stepwise selection for both direction on training dataset suggests using $D_{t-\tau}, F_{t-\tau}, G_{t-\tau}, H_{t-\tau}, y_{t-\tau}$, a few variables are eliminated in order to produce better FAR and MDR on training dataset. Overall, the logistic model used in Chorological splitting method is

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 G_{t-\tau} + \beta_2 H_{t-\tau} + \beta_3 y_{t-\tau}$$

The optimal Threshold selection in Chorological Splitting is much more difficult than that in Random Sampling. This is because the number adverse events and distribution of variables are different in the two datasets. We can see from the graph that the true optimal threshold for the testing dataset falls out of the optimal threshold selection region of the training dataset. Thus, even though the MDR is restricted as low as possible (Z =0.01%) in the training dataset, the MDR for the testing dataset is still relatively high.

Figure 4.14



4.2.2.2 Linear Regression

After initial data is prepared and partitioned into training and testing dataset. The variable selection process is again performed based on the principle of minimizing FAR and MDR in training dataset while keeping the model as simple as possible to prevent overfitting. The Linear Regression model used in Chorological splitting method is

$$\hat{Y}_t = \beta_0 + \beta_1 D_{t-\tau} + \beta_2 F_{t-\tau} + \beta_3 H_{t-\tau} + \beta_4 y_{t-\tau}$$

The problem for optimal threshold selection in linear regression chorological splitting model is even worse than that of logistic regression. Although the Z value is kept to (0.01%), the MDR on testing dataset is still very high and not satisfying. Fortunately, FAR for the testing dataset is greater than 2% throughout detection time from 5 minutes to 50 minutes, which means that the two rates are still in the ideal threshold selection region for testing dataset.

4.2.2.3 Artificial Neural Network (ANN)

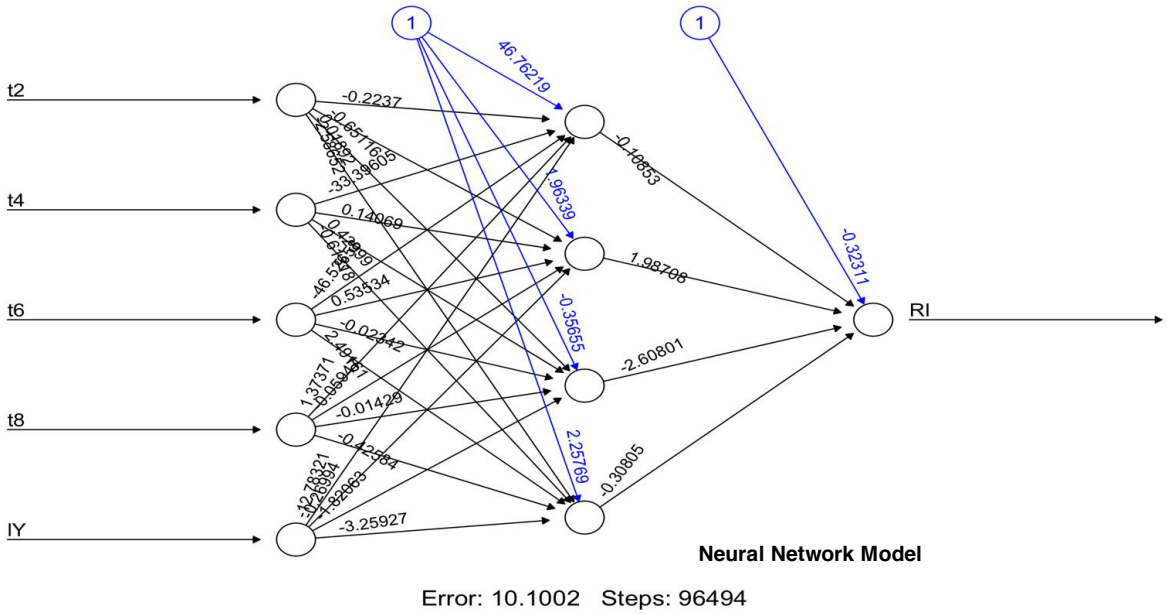
Since the testing MDR for both logistic and linear model in Chorological Splitting is not satisfying, a more advanced and complicated model Neural Network (ANN) is tested in order to produce better outcome. Neural network is always an appealing model in machine learning algorithm since the model function is inspired by the structure of the brain, which is apart from the traditional methods. According to Professor Russell form Hartford University, “In an analogy to the brain, an entity made up of interconnected neurons, neural networks are made up of interconnected processing elements called units, which respond in parallel to a set of input signals given to each. The unit is the equivalent of its brain counterpart, the neuron.” Also, since one of the most important features of ANN is its extraordinary capacity in adapting to the new environments. Thus, ANN could be a critical part in improving the early warning algorithm in this project.

The data preparation process for ANN is the same in chorological linear regression case. *glm()* function is used for initial variable selection in order to avoid overfitting as well as reducing computation time for model fitting. Consequently, only the variables that are significant ($B_{t-10}, D_{t-10}, F_{t-10}, H_{t-10}, y_{t-10}$) in *glm()* is used to fit in the ANN. After all predictors and response variables are prepared, data normalization based on min-max method (scales the data into the interval [0,1]) is performed to enhance the possibility of convergence in the ANN algorithm. Data demoralization is performed later before choosing threshold in training dataset and generating FAR and MDR in the testing dataset.

According to Dr. Nislsen in *"Neural Networks and Deep Learning"*, "neural networks with a single hidden layer can be used to approximate any continuous function to any desired precision". In addition, based on the nature of the algorithm, more hidden layers produce more complicated models which might over fit the training dataset. (Such assumption is proved in our experiment as well) Based on these evidence, one hidden layer is chosen.

Even though dataset is normalized before fitting in ANN, the training process is still very difficult as algorithm can hardly converge using the default Neural Network R package settings. Also, because of this reason, only the case when detection time is 50 mins is trained and tested in the ANN model. Additionally, it is worth to mention that 'Logistic' is the default activation function here. Fortunately, with 4 nodes in a single hidden layer, the model is able to converge and the below graph displays the neural network model with detection time 50 mins.

Figure 4.15



To simply explain the model, the black lines are the connections between each layer and the corresponding numeric values are the weights for each connection. The blue lines are called the 'bias term' which could be interpreted as the intercept of a linear model.

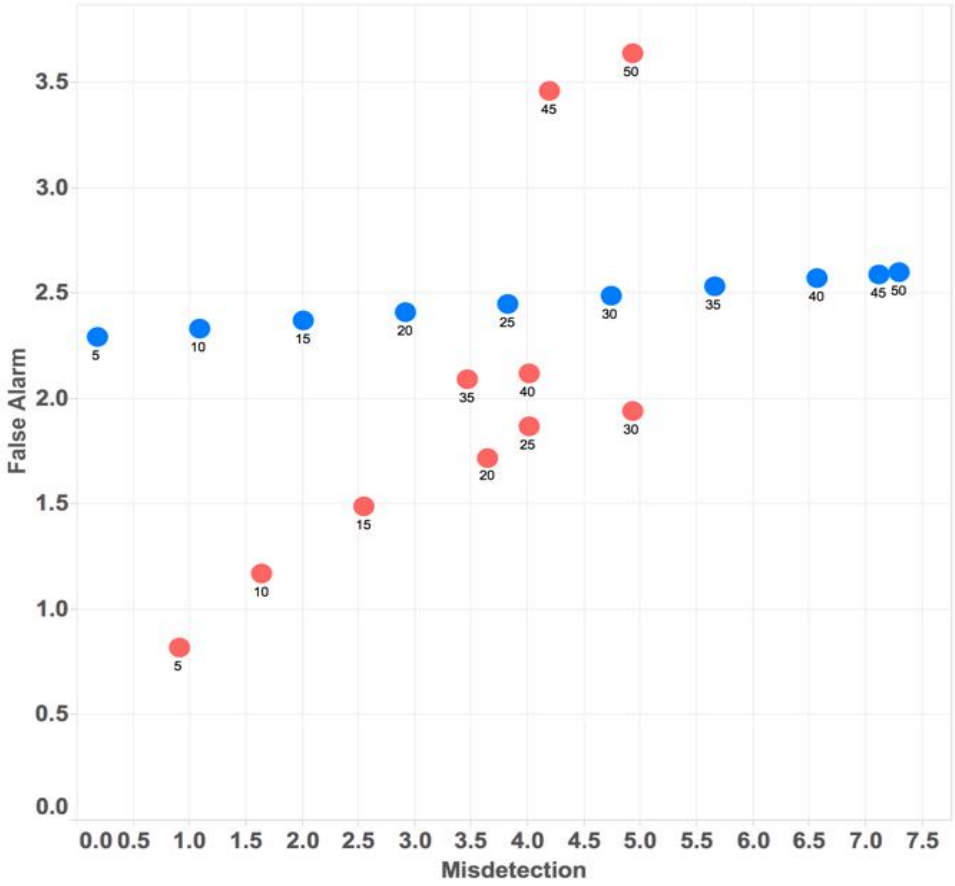
The results of optimal threshold selection in ANN model for chorological splitting is much better than that of Logistic Regression and Linear Regression. The Z value, the max allowed misdetection rate for training dataset, is still kept to 0.01%. The FAR and MDR for the testing dataset is presented in the results section.

4.2.3 Results Comparison

The graph below shows the model performance between logistic and linear regression of Chorological Splitting for detection time from 5 minutes to 50 minutes. As mentioned in previous section, even though MDR is restricted to 0.01% (Z) on training dataset, MDR in the testing dataset could not be easily reduced. However, from the logistic regression results, a pattern can be told that MDR of testing dataset is restricted when it reaches 5%. This might be

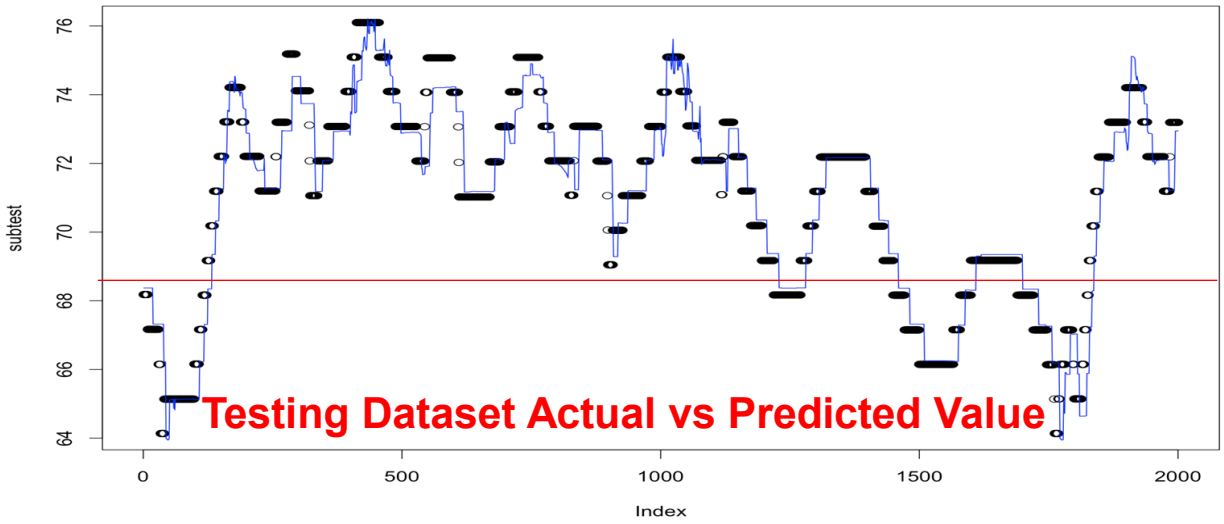
the effect between MDR and detection time previously discussed in the chapter. (See 4.2.1.1 Optimal Threshold Selection for Logistic Regression in Hemisphere – Random Sampling Method)

Figure 4.16



The below graph shows the response variable and predicted results in testing dataset for ANN using Chorological Splitting Method. The model catches the overall trend even in some extreme cases, which suggests that ANN could be a good model in our early warning algorithm.

Figure 4.17



To conclude, the table on the right shows the results of FAR, MDR for Logistic, Linear and ANN on testing dataset. Although FAR in ANN is higher than that of the other two, MDR of ANN is significantly lower. Another important finding worth repeated mention in chorological splitting method is that 7% of adverse events observations in training dataset is used to predict 93% of the adverse events in testing dataset. Therefore, the performance of ANN is considered satisfying. In addition, by manually adjusting algorithm parameters, there might be some potential to improve the ANN model.

4.3 Triangle

The third partition method is called triangle. The main purpose of this method is to generate results comparable to the output produced from ACCEPT. Hence the room temperatures at three detection times are tested, which are 80, 100, and 110 minutes. The following models for prediction are experimented based on model representativeness: Logistic regression is a typical model for binary variables; linear regression is a typical model for continuous variables; random forest is a typical model for non-parametric variables.

4.3.1 Logistic Regression

It is observed that among 92 days of data, there are 12 days with adverse events (low room temperatures below the threshold of 68.1 degrees Fahrenheit) and 80 days without any adverse events. In order to keep consistent with the setting from ACCEPT, the 12 days of temperature data are selected from the 80 days for training, which are day 2, day 10, day 11, day 12, day 13, day 17, day 18, day 20, day 25, day 27, day 28, and day 29. 2 days from 12 days with adverse events are chosen for testing data in order to produce prediction rates, which contributes to 66 permutations in total. The other 10 days out of 12 are combined with the 12 other days' training data so that the behavior of both adverse and non-adverse events can be better studied. Therefore, a total of 22 days are chosen to build training model and used to determine threshold. As introduced in 3.2.1.1, the same method to take lag is applied to logistic regression.

Variable selection starts from applying all predictors, and then certain variables are removed from model at 95% significance level. It is also experimented to add polynomials of each variable to predictors; however, the summary output shows that there is not much improvement in the model. The prediction results imply that model with all variables performs best in producing both MDR and FAR.

In this part, two threshold selection criteria are imposed:

1. Among the observations with MDR below 6%, return the one with the minimum sum of MDR and FAR (due to the fact that after MDR becomes lower than 6%, FAR will rise much more quickly).
2. Choose the threshold value with the lowest absolute value of difference between MDR and FAR (suggested by client).

4.3.2 Linear Regression

The same 12 days without adverse events are combined with 10 days selected from 12 days with adverse events for training. 2 days are randomly picked from 12 days with adverse events for testing. Since there are 66 possibilities to choose 2 from 12 days, each model is tested 66 times.

Similarly, variable selection based on significance and polynomial combination are also experimented in linear regression model. It turns out that applying all variables to prediction model returns the best prediction results in linear regression model.

Threshold selection for linear regression model follows the same rule introduced in 4.3.1.

4.3.3 Random Forest

As introduced in 4.3.1 and 4.3.2, the same datasets are used for training and testing in random forest model.

For random forest model, instead of which variables selected specifically, the experiment objective is how many variables selected. By adjusting number of parameters from 10 to 1, different sets of MDR and FAR are returned. After experiments and observations, when parameter number is 9, the prediction rates are lowest. Hence 9 variables are chosen for random forest prediction model.

Optimal threshold selection for random forest model is the same as 4.3.1 and 4.3.2.

4.3.4 Results Comparison

Figure 4.18 Detection time = 80 minutes

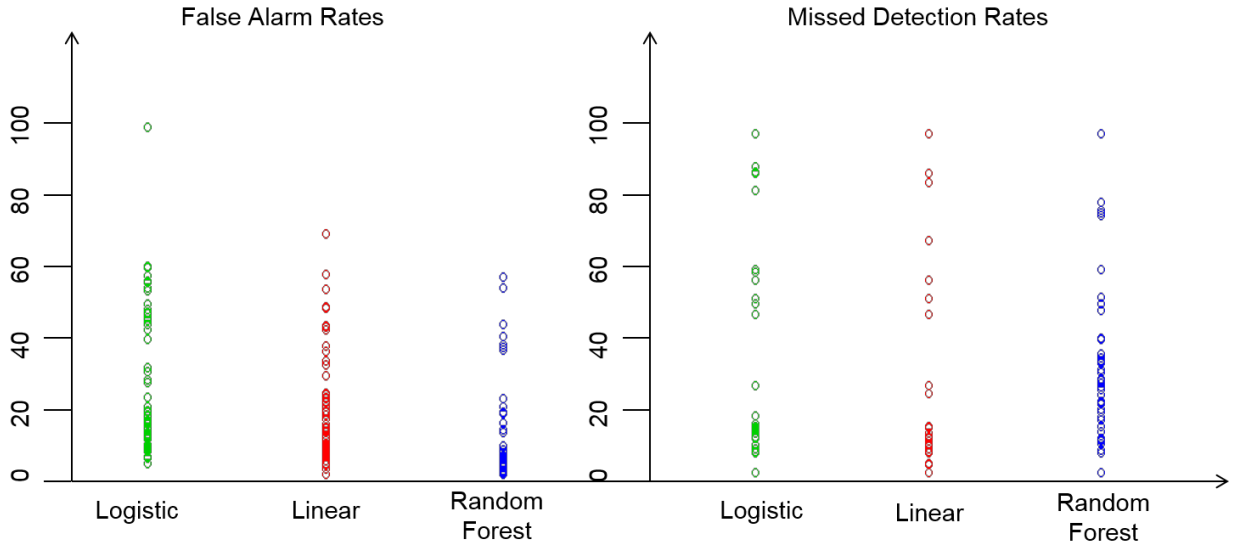


Table 4.3 FAR at detection time = 80 min (selected)

Linear Regression	Logistic Regression	Random Forest
57.58	60	17.88
13.26	16.71	3.17
10.7	15.58	0
7.69	22.38	2.1
31.49	38.95	43.37
12.54	17.31	4.78
10.49	14.45	2.1
9.18	11.06	5.41
6.06	13.75	4.43
20.57	11.72	2.15

Table 4.4 MDR at detection time = 80 min (selected)

Linear Regression	Logistic Regression	Random Forest
0	0	21.13
0	0	22.96
6.19	0	20.35
2.63	0	39.47
0	0	9.94

0	0	35.1
7.02	0	32.46
10.17	10.17	50
7.02	0	39.47
0	12.8	48

Figure 4.19 Detection time = 100 minutes

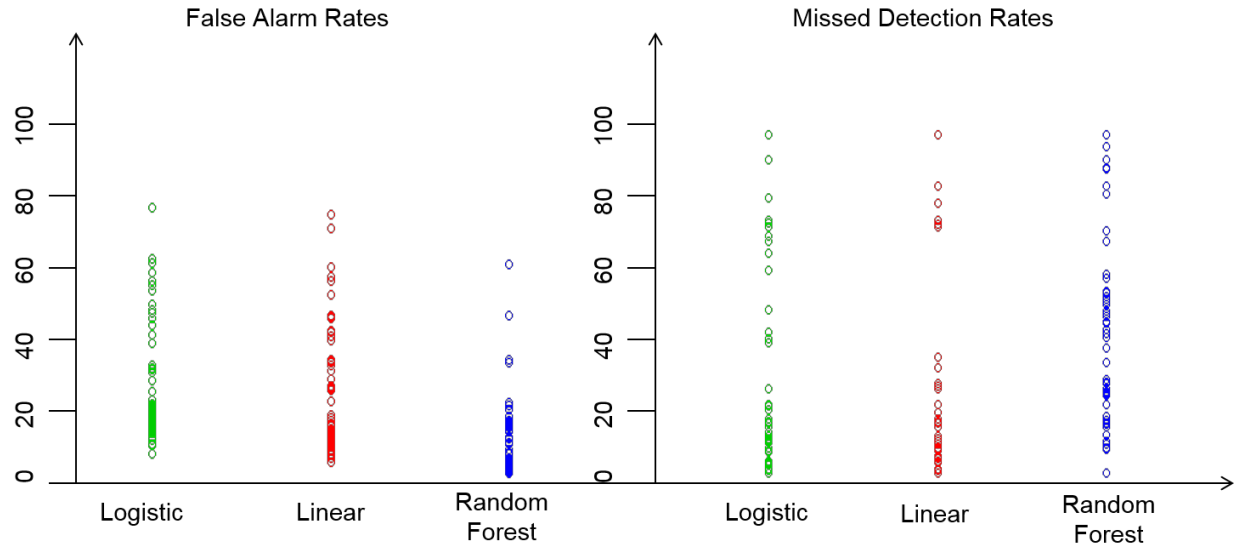


Table 4.5 FAR at detection time = 100 min (selected)

Linear Regression	Logistic Regression	Random Forest
58.02	60.49	20.06
15.07	18.55	5.51
12.74	16.27	0.94
13.24	23.17	2.36
31.46	39.04	44.38
15.62	19.22	3
12.29	15.37	1.18
7.88	11.69	4.3
6.62	15.13	3.07
11.41	13.11	0.97

Table 4.6 MDR at detection time = 100 min (selected)

Linear Regression	Logistic Regression	Random Forest

0	0	23.7
0	0	16.84
0	0	27.03
3.57	0	50.89
0	0	14.53
0	0	40.1
0.89	0.89	53.57
10.34	10.34	47.41
7.14	2.68	46.43
15.45	15.45	57.72

Figure 4.20 Detection time = 110 minutes

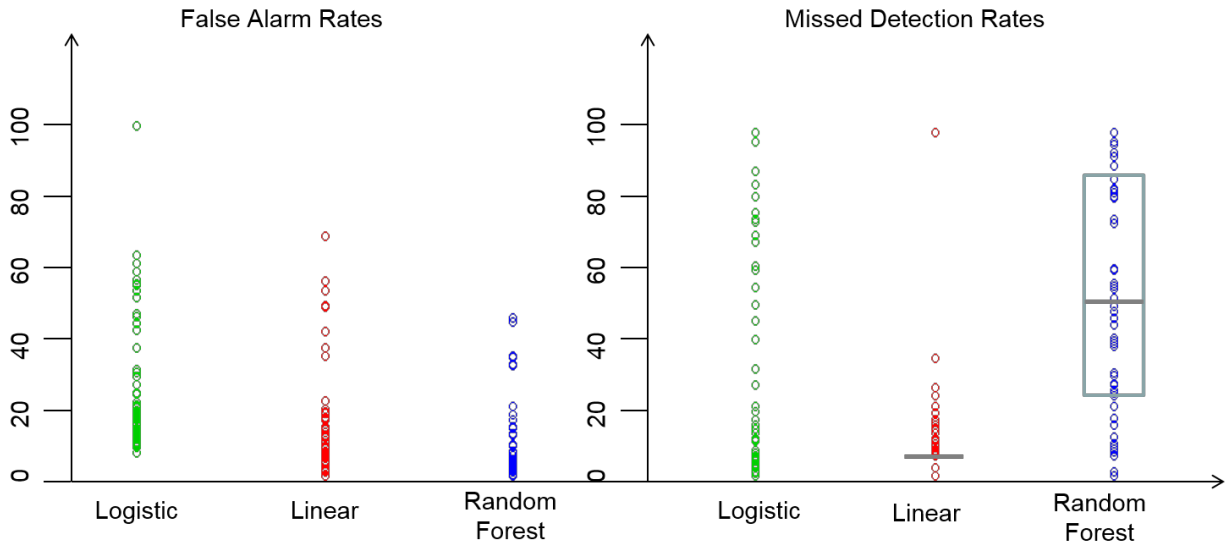


Table 4.7 FAR at detection time = 110 min (selected)

Linear Regression	Logistic Regression	Random Forest
55.9	60.87	19.88
12.75	19.71	3.48
8.53	17.06	1.42
3.56	23.75	2.61
16.95	63.28	45.2
13.21	20.42	1.8
8.55	16.39	1.66
7.19	12.23	9.11
3.56	16.15	3.33
3.66	23.66	2.2

Table 4.8 MDR at detection time = 110 min (selected)

Linear Regression	Logistic Regression	Random Forest
0	0	24.88
0	0	49.46
6.42	0.92	38.53
7.27	0	54.55
0	0	1.13
0	0	43.94
7.27	2.73	83.64
10.53	10.53	51.75
7.27	4.55	54.55
15.7	0	60.33

As shown in figures and tables above, linear regression model produces the best prediction results for MDR, while random forest model produces the best results for FAR because there is a trade-off between the two rates. Generally speaking, the linear regression model performs the best when producing both MDR and FAR. Additionally, there is an overall trend that both prediction rates increase as the detection time goes up. This is due to the loss of predicting power with farther prediction unit. An interesting result is also observed: After removing the days 16, 19, 23, 26, 61 and 73, both MDR and FAR tend to be much lower, since the data for these 6 days is identified as problematic. The finding turns out to agree with the results generated from ACCEPT. Furthermore, compared to the output from ACCEPT, triangle method does not produce so competitive prediction results as ACCEPT does.

5. Conclusion and Recommendation

To summarize the work conducted in this project, three data partition methods are developed and tested. Four categories of models are applied in accordance with the quality of

data and feature of models. Different parameters in addition to various selections of variables are so also experimented within each model. Series of models in a simple format with only a few variables and short computing time have returned promising results in the first two partition methods: sliding window and hemisphere, while they fail to produce similar outcomes in the third method due to the limit of testing dataset. Compared with more advanced machine learning tools such as Supporting Vector Machine, simpler model costs less to train and modify as the dataset keeps changing while still holding decent agility to detect upcoming adverse events, which is worth future exploration and improvement.

For future work, the following attempts are suggested:

1. Conduct more thorough studies of data as more are collected. As there are only a few observations in current dataset, there might be potential facts regarding the data to be uncovered.
2. Experiment more models and data partition methods. Due to the limit of time, only a few models are tested. Though current data partition methods show plausible quality, a more comprehensive process or combination might still improve the results.
3. Compare more threshold selection strategies. There are still many published selection methods not applied and tested in this project. Also, a more dynamic selection is worth testing. For example, for the training dataset, it does not have to return a threshold with fixed numeric value. Instead, it could return the difference between that value and the mean of response variable and apply the difference to the testing dataset and find a more "customized" threshold.

References

Martin, Rodney, Santanu Das, Vijay Janakiraman, Stefan Hosein " ACCEPT: Introduction of theAdverse Condition and Critical Event Prediction Toolbox " *NASA Technical Report*, , Nov. 2016.

Basak, Aniruddha, Ole Mengshoel, Stefan Hosein, and Rodney Martin. "Scalable Causal Learning for Predicting Adverse Events in Smart Buildings." (2016). *Sustainability Base*. Web. 1 Mar. 2016. <https://c3.nasa.gov/dashlink/resources/954/>.

Basak, Aniruddha, Ole Mengshoel, Stefan Hosein, Rodney Martin, Jayasudha Jayakumaran, Mario Gurrola Morga, and Ishwari Aghav. "Identifying Contributing Factors of Occupant Thermal Discomfort in a Smart Building." (2016). *Sustainability Base*. Web. 1 Mar. 2016. <https://c3.nasa.gov/dashlink/resources/955/>.

Kumar, Rajeev, and Abhaya Indrayan. "Receiver Operating Characteristic (ROC) Curve for Medical Researchers." *Indian Pediatr Indian Pediatrics* 48.4 (2011): 277-87. Web. 1 Mar. 2016.

Russell, Ingrid. "Definition of a Neural Network." *Definition of a Neural Network*. 1996. Web. 1 Mar. 2016.

Nielsen, A. Michael. "Neural Networks and Deep Learning", Determination Press, 2015