SVM based LIFI-WIFI Vertical Handover

Project report submitted in partial fulfillment
of the requirements for the degree of

Bachelor of Technology
in
Electronics and Communication Engineering

by

Anant Sharma - 17uec019
Ayush Maheshwari - 17uec034
Kartik Singhal - 17uec062
Ritik Aggarwal - 17uec093

Under Guidance of
Dr. Akash Gupta

Department of Electronics and Communication Engineering
The LNM Institute of Information Technology, Jaipur

December 2020

The LNM Institute of Information Technology
Jaipur, India

CERTIFICATE

This is to certify that the project entitled “SVM based LIFI-WIFI Vertical Handover” , submitted by Anant Sharma (Roll no 17uec019), Ayush Maheshwari (Roll no 17uec034), Kartik Singhal (Roll no 17uec062) and Ritik Aggarwal(Roll no 17uec093) in partial fulfillment of the requirement of degree in Bachelor of Technology (B. Tech), is a bonafide record of work carried out by them at the Department of Electronics and Communication Engineering, The LNM Institute of Information Technology, Jaipur, (Rajasthan) India, during the academic session 2020-2021 under my supervision and guidance and the same has not been submitted elsewhere for award of any other degree. In my/our opinion, this thesis is of standard required for the award of the degree of Bachelor of Technology (B. Tech).

Date	Adviser: Dr. Akash Gupta

Dedicated to our Family, Friends and Mentor

Acknowledgments

We express our heartiest thanks and deep regards to our supervisor Dr. Akash Gupta for his continuous guidance, inspiration and constructive suggestions which were very crucial for this project. His encouraging remarks helped us in improving our skills and also overcome the obstacles at different stages of the project.

Abstract

The growth in data traffic in indoor wireless networks is explosive and with greater service demands comes greater data rate demands. To ensure high data rates lot of development has happened in LIFI technology which has the capability to increase the data rates to around 100 times than that of WIFI technology. But LIFI technology has it's own problems. So in this project the idea was to get rid of problems of LIFI technology while ensuring high data rates. The project is about how using LIFI-WIFI technology together with machine learning algorithms can lead us to great results.

Contents

Chapter *1*

Introduction

1.1 The Area of Work

In this project we identified how LIFI-WIFI technology can be really beneficial. Here we studied how LIFI-WIFI together can be used and how handover between the two can ensure continuous data transfer. We analyzed different machine learning algorithms and came up with SVM as the most accurate model. We also created a data-set that contained all the relevant variables needed to predict the target variable(the state of the network). We compared our technique with other techniques to measure the performance of our model. At last we analyzed the model more to ensure that our model is robust enough and can be deployed for practical data-sets.

1.2 Problem Addressed

A WIFI is known to provide a data rate of 32 mbps maximum, but LIFI can provide a data rate of 100 times more than that of WIFI. But LIFI technology is sensitive to blockage hence this project addresses the same problem and fixes it with various machine learning algorithms along with the use of a WIFI module with it.

1.3 Traditional WIFI System

Currently the indoor wireless networks are greatly dependent on WIFI technology. WIFI uses electromagnetic waves to modulate the message and transfer it with help of a device known as WIFI router. Due to large demand there is a lot of congestion on this spectrum and these frequencies are also known to harm human health. The data rate provided by WIFI is also low when compared to LIFI. All these factors motivates us to find an alternative system that can help reduce congestion on the radio spectrum and provide us high data rates.

1.4 LIFI only System

1.4.1 LIFI over WIFI

Use of LIFI in indoor wireless networks can provide great results. LIFI transmits data on high frequency light waves using light emitting diodes (LEDs), and uses photodiodes (PDs) to identify the received signals.LiFi has several appealing advantages, such as more data rates, safety for health, high security when compared with conventional WIFI technology and a wide free-licensed spectrum which overcomes the problem of radio spectrum congestion.

1.4.2 Problem with LIFI only system

Visible light is very easily affected by obstructions, which may result in unstable optical communication links. Therefore, the obstruction that LiFi needs to pass through is how to effectively deal with the blockage experienced by the channel.

1.5 Hetrogenous LIFI-WIFI system

To get high data rates and stablility in communication for mobile users, an indoor heterogeneous communication network having both LiFi and WiFi should be introduced, which can achieve better performance than either LiFi only or conventional WiFi system. The main idea behind this system is that we should be able to switch a user to WIFI whenever the user suffers a blockage from the LIFI module.

1.5.1 System Model

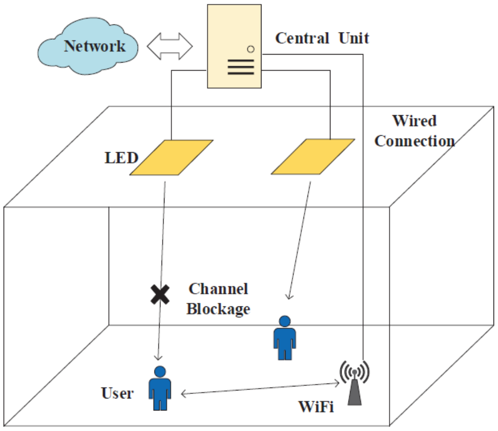


Figure 1.1 System model prototype

As shown in Figure 1.1,we have considered an indoor hybrid LiFi and WiFi network. A WiFi access point is placed in the corner of the room to cover it entirely, and the WiFi rate remains constant. A few LEDs are placed on the ceiling as our LIFI access points.Only either of the LiFi access point or WiFi access point can be accessed by the user at a given instant of time. When a user is on a LIFI channel, it will connect to the nearest LiFi access point or if simply put together to the nearest LED. We are assuming that there is a central unit present in the room that is responsible for deciding the network access should be from LIFI or WIFI for each user.

1.5.2 Working

There can be two ways in which handover can occur between LIFI and WIFI modules.

1.5.2.1 Immediate-vertical handover

In this type of handover, the handover between LiFi and WiFi is immediately performed when a blockage is experienced by the LIFI channel and the user will be switched to its initial channel i.e.to a LIFI channel back if the blockage disappears. Immediate-vertical handover ensurea the continuity of data transfer, but the frequent handovers between both the networks present could significantly decrease the effective time of data transmission,which leads to low equivalent data rates.

1.5.2.2 SVM based vertical handover

In this type of handover, we introduce a support-vector-machine(SVM) based network access type decision scheme, which can effectively reduce the number of frequent handovers and hence increasing the effective data rate. This model has more practicality, where the central unit as shown in figure 1.1 should be able to predict the network access type on the basis of the information of various channel blockage parameters provided to it. In this system we design a SVM model based on the already known channel blockage parameters and their network access types and hence this model can be deployed into the central unit to do it's work. The main requirement of this system is that the model which is being created should cover a wide range of possibilities and should not over-fit else it will provide vague results for the user. The results obtained after simulation show that this scheme has the ability to achieve higher equivalent data rate for users if compared to other types.

Chapter *2*

Litrature Survey

1. **Kaixuan Jiyz, Tianqi [1] :** This research paper was based on the necessary parameters that come into play while using LIFI and WIFI together and their right values so to as to create a dataset that can work be used to create an SVM model.

2. **Daniela Alexandra Embus [2] :** This research is based on supervised learning on a larger scale. The types of handovers, how can they be used are listed in this paper and how a user with his phone if moves in a room with multiple access points can be connected to the best access point using handover and machine learning technique at different point of time.

3. **DeepMIMO [3] :** This research paper is based on millimeter wave based base stations that are assumed to be aligned in a street and a lot of users are assumed at different points of street and the user is able to connect to the best station with least blockage at any point in the street.

Chapter *3*

Proposed Work

We propose the following work in our project:

1. First we created a dataset based on different blockage parameters and their network access type to train our SVM model.

2. The second stage of our project was to build a model on the constructed dataset with best hyper-parameters that can provide the best accuracy.

3. Lastly we ensured by analyzing the model more that this model can be deployed to practical values and does not overfit on the training dataset.

4. This model can be used to determine the network access type on providing it with the blockage parameters of the current scenario.

All the code was done in R language in this project and the following libraries were used:

- Caret : For cross validation and model building

- ggplot : For exploratory analysis of the dataset

- e1071 : For tuning the hyper-parameters of the SVM model

- Randomforest : For tuning the hyper-parameters of random forest model

3.1 Training dataset construction

For the training and testing of a SVM model, large number of samples are needed.Further, these samples are separated into training and testing samples for generating the SVM model and for verifying the effectiveness of the trained model. There are three parameters that are identified for determining the network access type. We created one lac samples to make the model more generalised.

3.1.1 Blockage occurrence rate

As the name suggests it tells us about the number of times the communication between source and destination suffered blockage It ideally is the times of channel blockages happened during a fixed period T. The blockage occurrence rate ideally is an integer between 0 and 10 if we take the fixed time period to be a minute.

3.1.2 Blockage occupation rate

It is the ratio between the period of time when a user experiences channel blockage and the fixed time period stated above. In other words it tells us about the time for which communication was blocked in fixed time T. Blockage occupation rate is a real number between 0 and 1 as it is the percentage of time, blockage is experienced in time T.

3.1.3 LIFI transmission rate

This variable is nothing but the transmission rate provided by LIFI. We considered our WIFI module to have a fixed transmission rate of 32 mbps, so we know that ideally LIFI should provide us a transmission rate of 3200 mbps. To generalise our model we considered LIFI transmission rate to lie between 2900 to 3200 mbps.

3.1.4 Determining network access type

We consider three different network access types:

1. In the ”WiFi only” network access type, the user is connected to WiFi for all the time and this is denoted by k=1.

2. In the ”LiFi only” network access type, the user is connected to LiFi for all the time without any handover and this is denoted by k=2.

3. If a blockage occurs and user is switched from LiFi to WiFi and after the disappearance of blockage it is switched back to its initial state, i.e. to a LiFi channel, this type of network access is defined as ”LiFi/WiFi” and is given by k=3.

Our main aim is to provide the user with that type of network access that has maximum data rate. The equivalent data rate at any network access type can be given as: (efficiency of LIFI)*(transmission rate of LIFI)+ (efficiency of WIFI)*(transmission rate of WIFI). So whichever state k=1,2 or 3 provides maximum equivalent data rate should be the network access type that user should be provided with. Efficiency of LIFI and Efficiency of WIFI for different network access type can be given by figure 3.1 and figure 3.2 respectively where miu is the occupation rate vector and lambda is the occurrence rate vector and TH is the ratio between the time interval T and a single vertical handover overhead .

$$\tau_{\kappa,n}^{\text{LiFi}} = \begin{cases} 0, & \kappa = 1, \\ 1 - \eta_n, & \kappa = 2, \\ \max\left\{1 - \eta_n - \lambda_n T_H, 0\right\}, & \kappa = 3. \end{cases}$$

Figure 3.1 LIFI efficiency

$$\tau_{\kappa,n}^{\text{WiFi}} = \begin{cases} 1, & \kappa = 1, \\ 0, & \kappa = 2, \\ \max\left\{\eta_n - \lambda_n T_H, 0\right\}, & \kappa = 3. \end{cases}$$

Figure 3.2 WIFI efficiency

3.2 Model creation

We split the whole dataset into two parts in the ratio of 7:3 - ”train” and ”test” dataset. This was done to create our model on train dataset and then validating the model’s accuracy on the test dataset. We tested different machine learning algorithms such as random forest, gradient boosting etc. But SVM gave best accuracy of 99.7 percent on the test dataset.

3.2.1 SVM (State vector machine)

To understand why SVM gave good results in this case we need to understand what SVM does.

In the SVM algorithm, every point is plotted as a n-dimensional space point with the value of a particular coordinate equals to the value of each feature. Then, a classification is done by locating the hyper-plane that separates the two or more classes in a well manner. In figure 3.3, a linear hyper-plane is easily identifiable between the two classses and hence the SVM classifier can easily fit a linear hyper-plane between them to differentiate both classes but in figure 3.4 we can see the two classes cannot be separated using a linear hyper-plane but it can be seen a circular hyper-plane can be used to separate the two classes, for this SVM algorithm has a kernel trick which can fit a non-linear hyper-plane in these type of scenarios.

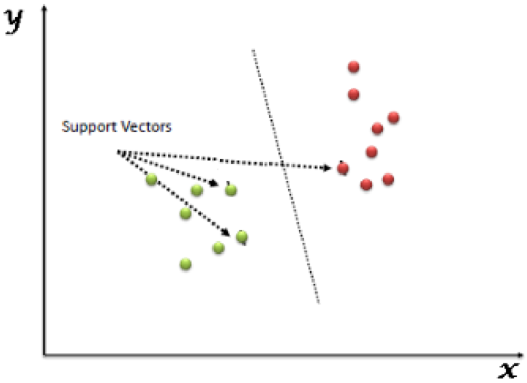


Figure 3.3 Linearly Separable

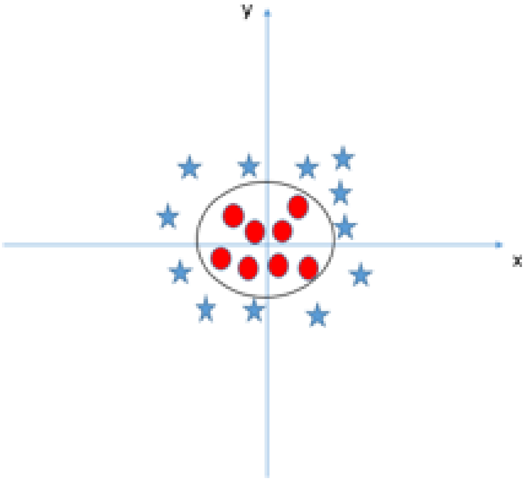


Figure 3.4 Linearly not separable

3.2.2 Why SVM ?

As we can see from section 3.2.1 that SVM works best for identifying those type of classes which when plotted on a n-dimensional plane can be separated linearly or non-linearly.
So in our case we know that there are two blockage parameters and we know that for high values of occurrence and occupation rate we should keep our user on WIFI and for low values we should keep our user on LIFI and somewhere in between the state with handover will occur. So this scenario proves that a hyper-plane can easily differentiate the three classes because when the three classes map to specific values of parameters hence SVM best works here.

3.2.3 Hyper-parameters selection

Svm has different hyper-parameters that need to be tuned to get best results.

3.2.3.1 Kernel

This hyper-parameter can have values such as ”linear”, ”polynomial” etc, in general it specifies the type of hyper-plane we want to fit on our dataset (linear or non-linear).

3.2.3.2 Degree

This hyper-parameter is specified only when the kernel type is ”polynomial” or ”rbf”. We specify the degree of the polynomial that is used to separate the classes here.

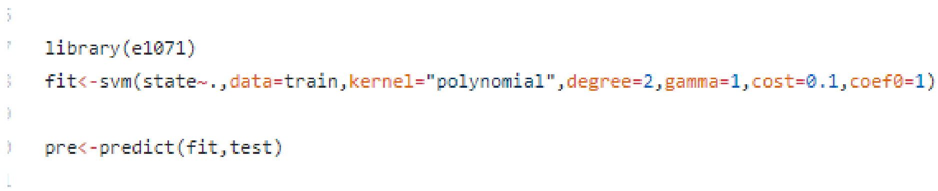
3.2.3.3 Gamma

A SVM can cause problem of over-fitting sometimes which is discussed in later part of this report. This parameter directly is responsible for the extent of over-fitting, higher the value of gamma, higher is the over-fitting.

3.2.3.4 Cost

This parameter defines the extent of margin in a model. Margin is the shortest distance between the data points and the hyper-plane. This is considered more important when we are fitting a linear hyper-plane but not in the case of non-linear hyper-plane.

So to get the best hyper-parameters we used grid search to identify the best possible set of values to get the best possible accuracy. In grid search our machine runs each and every combination of hyper-parameters and checks accuracy for each combination and provides us with the best set of values. So the combination with best set of accuracy is given as:
kernel=”polynomial”
degree=2
gamma=1
cost=0.1
Finally figure 3.5 shows how SVM model with these hyper-parameter can be created.



```
library(e1071)
fit<-svm(state~.,data=train,kernel="polynomial",degree=2,gamma=1,cost=0.1,coef0=1)

pre<-predict(fit,test)
```

Figure 3.5 Code to build SVM model on train dataset

3.2.4 How will this work ?

Any model is not of any use until we know how can this work in practical cases. As shown in section 1.5.1 the system model shows how a central unit will decide which network access type should be provided to the user. Now if the central unit has this model which has classified different network access types on the basis of one lac different data points, we just need to provide the central unit with three values - blockage occurrence rate, blockage occupation rate and the LIFI transmission rate and hence it will give out the network access type accordingly.

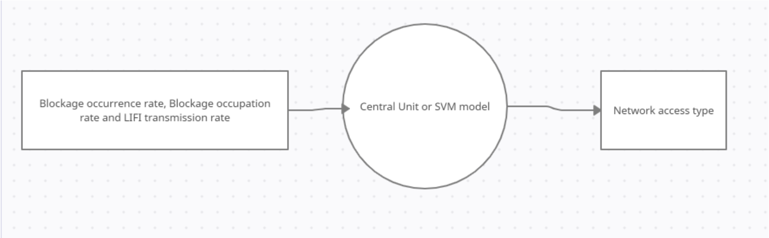

Blockage occurrence rate, Blockage occupation rate and LIFI transmission rate
Central Unit or SVM model
Network access type

Figure 3.6 Block diagram of model's practical use

3.3 Model analysis

This stage of the project includes to check whether our model will be able to provide same kind of accuracy as it is able to provide on the dataset created by us. For this we should ensure that our model does not overfit.

3.3.1 What is overfitting ?

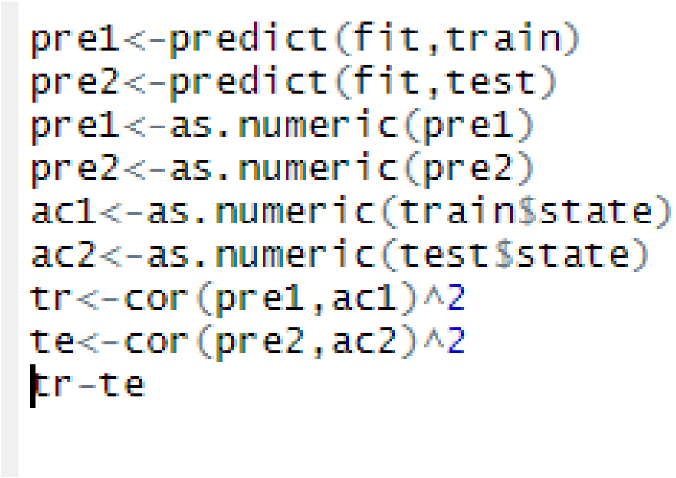
When the training data is modelled very good by a model, Overfitting occurs. It happens when a model is unable to give same level of performance on new data to the extent of its learning of the details of the training data.
So we can detect if our model overfits or not in two ways:

3.3.1.1 Using k-fold cross validation

K-fold cross-validation is a machine learning method which is used to evaluate the skill of a machine learning model on unseen data. This is nothing but a re-sampling technique, for example 10-fold cross validation implies we divide our dataset into 10 groups and train our model on 9 of the groups and test it on the 10th one, and further we keep each group as a test dataset one at a time while using others as a train dataset. Using this we see that if our model is giving different accuracies for different groups, if yes then our model is overfitted and will give bad results when a new data is served to it.

3.3.1.2 Using R-squared value

In our project we used this technique to identify if our model is overfitting or not. In machine learning R-squared value of any dataset tells us about the performance of that model on that particular dataset. So ideally the difference between the R-squared value of the train dataset and test dataset should be close to 0 if our model is not overfitting. This difference being close to 0 ensures that our model is performing equally well on train and test dataset.
Mathematically R-squared value of any data can be calculated as correlation between actual value and predicted value squared. The code to take out this correlation is given in figure 3.7.



```
pre1<-predict(fit,train)
pre2<-predict(fit,test)
pre1<-as.numeric(pre1)
pre2<-as.numeric(pre2)
ac1<-as.numeric(train$state)
ac2<-as.numeric(test$state)
tr<-cor(pre1,ac1)^2
te<-cor(pre2,ac2)^2
tr-te
```

Figure 3.7 Code to determine overfitting

In our case this value came out to be 0.003377 which is close to zero. Hence we can confirm that this model can give good result if any new data is thrown to it and it is not overfitted.

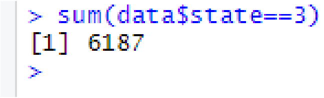
Chapter *4*

Simulation and Results

We used two machine learning algorithms on same train dataset and used same test dataset to check the accuracy so that results come out to be fair.

4.1 Exploratory Analysis

After the dataset creation we identified how building a model according to this dataset will ensure less handovers than in the immediate handover case. As shown in figure 4.1 the number of states with handovers are 6187 which is approximately 6 percent of all the values. Hence only 6 percent of the scenarios will suffer handovers. Graphically the distribution of three different states is displayed in figure 4.2.



```
> sum(data$state==3)
[1] 6187
>
```

Figure 4.1 Number of states with handovers

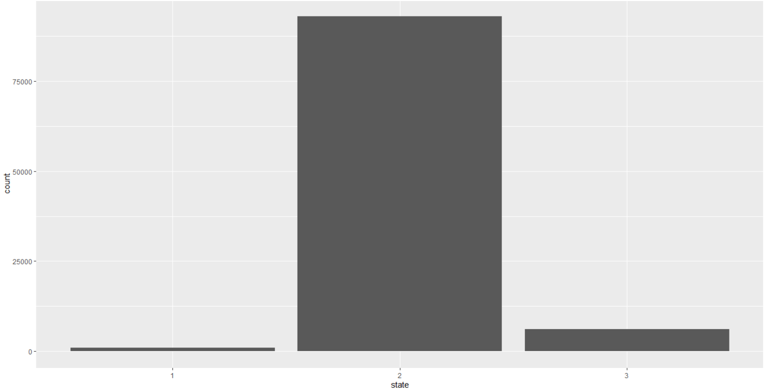
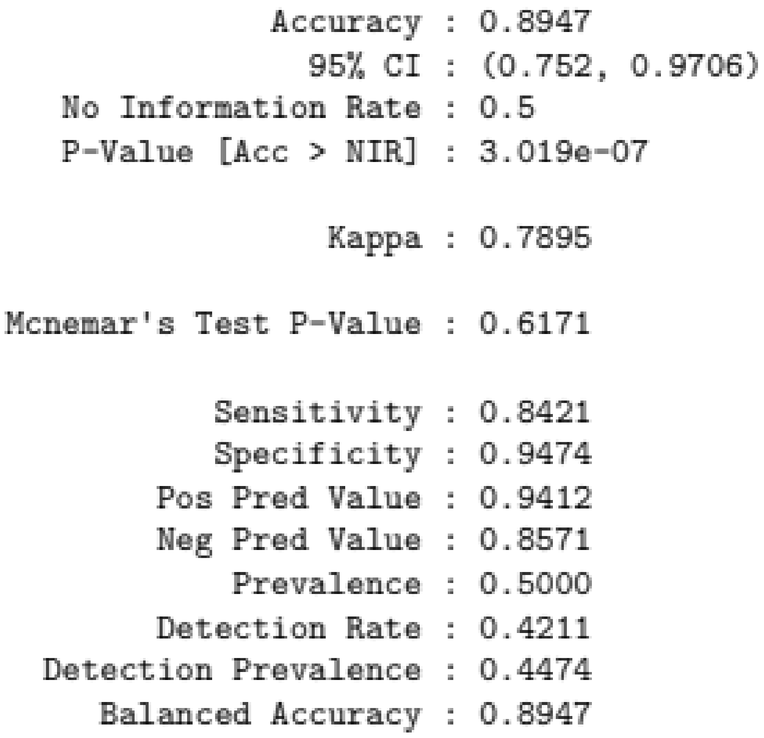


Figure 4.2 Distribtion of all three states

4.2 Using random forest

Using random forest algorithm on our test dataset we were able to get an accuracy of 89 percent approximately as shown in the confusion matrix in figure 4.1.

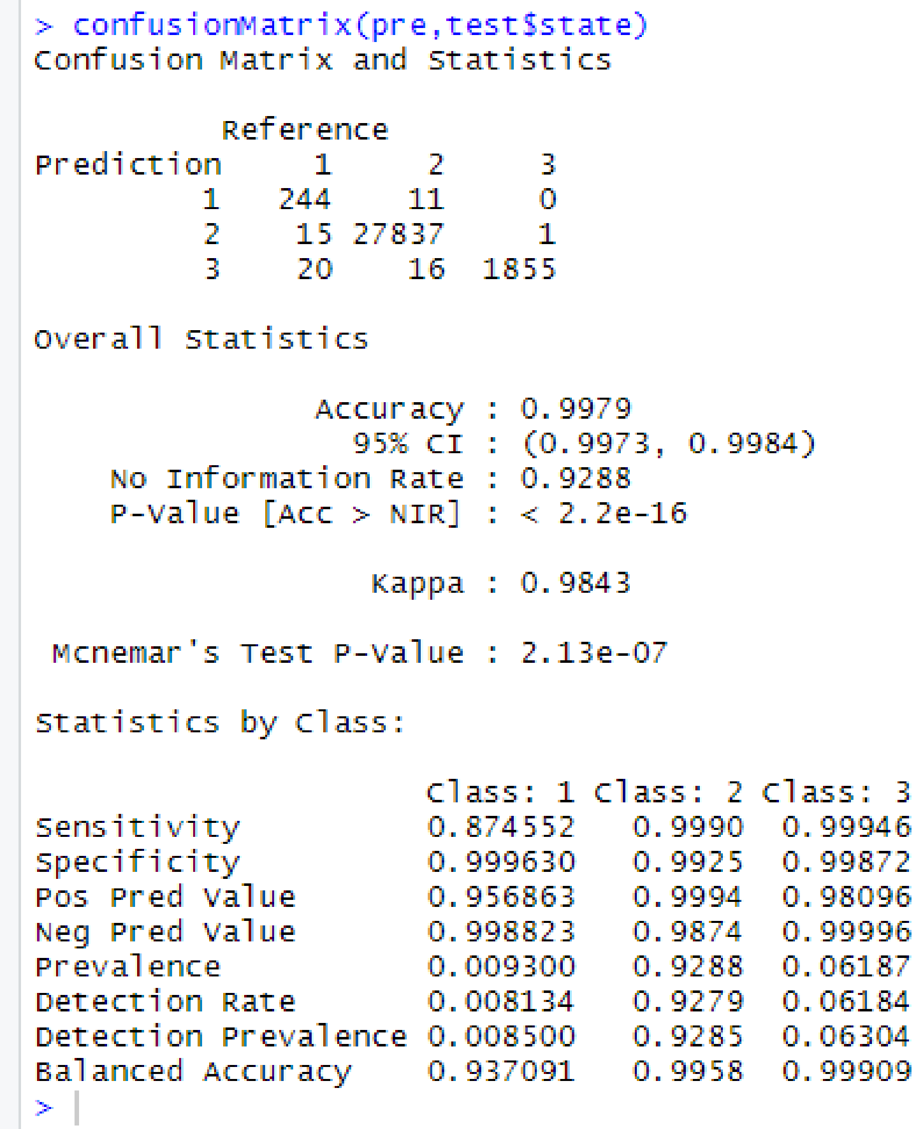


```
               Accuracy : 0.8947
                 95% CI : (0.752, 0.9706)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : 3.019e-07

                  Kappa : 0.7895

 Mcnemar's Test P-Value : 0.6171

            Sensitivity : 0.8421
            Specificity : 0.9474
         Pos Pred Value : 0.9412
         Neg Pred Value : 0.8571
             Prevalence : 0.5000
         Detection Rate : 0.4211
   Detection Prevalence : 0.4474
      Balanced Accuracy : 0.8947
```

Figure 4.3 Confusion Matrix that shows the accuracy of random forest

4.3 Using SVM (State Vector Machine)

Using SVM on our test dataset we were able to get an accuracy of 99 percent as shown in figure 4.2 which is significantly greater than the random forest algorithm due to reasons stated in chapter 3.

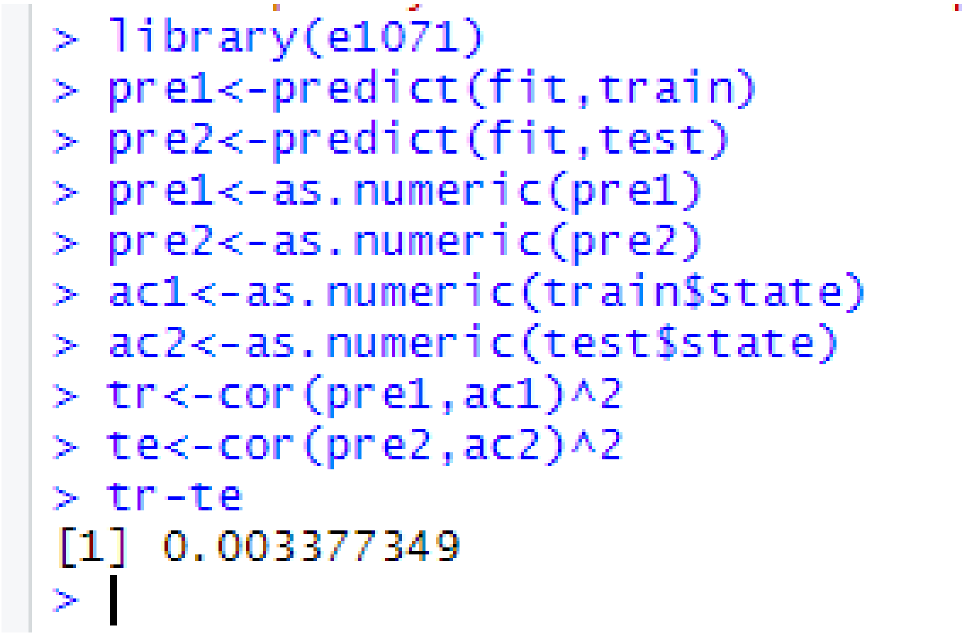


```
> confusionMatrix(pre,test$state)
Confusion Matrix and Statistics

          Reference
Prediction     1     2     3
         1   244    11     0
         2    15 27837     1
         3    20    16  1855

Overall Statistics

               Accuracy : 0.9979
                 95% CI : (0.9973, 0.9984)
    No Information Rate : 0.9288
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9843

 Mcnemar's Test P-Value : 2.13e-07

Statistics by Class:

                     Class: 1 Class: 2 Class: 3
Sensitivity          0.874552   0.9990  0.99946
Specificity          0.999630   0.9925  0.99872
Pos Pred Value       0.956863   0.9994  0.98096
Neg Pred Value       0.998823   0.9874  0.99996
Prevalence           0.009300   0.9288  0.06187
Detection Rate       0.008134   0.9279  0.06184
Detection Prevalence 0.008500   0.9285  0.06304
Balanced Accuracy    0.937091   0.9958  0.99909
> 
```

Figure 4.4 Confusion Matrix that shows the accuracy of SVM

4.4 Checking overfitting condition

As discussed in chapter 3 the difference in R-squared values of train and test dataset should be close to 0 if our model is not overfitting, in our case it came out to be 0.00337 as shown in figure 4.3 which is very close to 0 hence our model is robust enough to be deployed for any new values.



```
> library(e1071)
> pre1<-predict(fit,train)
> pre2<-predict(fit,test)
> pre1<-as.numeric(pre1)
> pre2<-as.numeric(pre2)
> ac1<-as.numeric(train$state)
> ac2<-as.numeric(test$state)
> tr<-cor(pre1,ac1)^2
> te<-cor(pre2,ac2)^2
> tr-te
[1] 0.003377349
> 
```

Figure 4.5 Results to ensure no overfitting exist

Chapter *5*

Conclusions and Future Work

We can ensure a data rate that is 100 times more than the traditional system of WIFI using this LIFI-WIFI SVM based handover system. We created one lac values to train our dataset so that we can cover all the scenarios that can occur and our model can predict it effectively. We tried ensemble technique such as random forest to test the accuracy but the most accurate model came out to be SVM due to reasons already discussed above. Then after model creation we were able to prove that this SVM model can predict accurate values on outside data as well.

5.1 Scope of further work

This model is limited to handover between LIFI-WIFI in a particular room or a floor, but in our view this model can be implemented on a larger scale as well.
Such as handover between 4G-5G cells. That would require a lot more number of parameters and a lot more number of data points to create this kind of model, but with sufficient resources this can be done efficiently.
Our model has learnt from one lac data points and it predicts values on the basis of that particular dataset only. A model can be created which is self learning, which can be able to add new data points into it's scope for more accurate results.

Bibliography

[1] Kaixuan Jiyz, Tianqi Maoy, Jiaxuan Cheny, Yuhan Dongz and Zhaocheng Wang. SVM-based Network Access Type Decision in Hybrid LiFi and WiFi Networks

[2] Ahmed Alkhateeb DeepMIMO: A Generic Deep Learning Dataset for Millimeter Wave and Massive MIMO Applications

[3] Daniela Alexandra Embus, Andres Julián Castillo, Fulvio Yesid Vivas, Oscar Mauricio Caicedo, and Armando Ordóñez NetSel-RF: A Model for Network Selection Based on Multi-Criteria and Supervised Learning