Location Determination of a Mobile Device using IEEE 802.11 Access Point Signals

Siddhartha Saha Kamalika Chaudhuri
Dr Dheeraj Sanghi Dr Pravin Bhagwat

May 5, 2002

Abstract

One of the most popular wireless LAN systems in use these days is the IEEE 802.11. In this project, we examine whether it is possible to detect the location of a mobile user within a building using this technology. To do this, we exploit the fact that the signal strengths received by an user at different locations in a building is different. We build a database of signal strength information for various locations, and use this information to determine which location a given test data comes from.

Our experiments show that it is indeed possible to perform location detection using this approach. We have achieved a success rate of upto 90%, when the minimum distance between two points is 3.12 meters. This approach is also cost effective, since it does not need any extra hardware other than what is already present in a typical office building deploying a 802.11 wireless LAN.

1 Introduction

With the current increase in mobile computing devices and wireless LANS, position detection of a mobile user has become a pertinent issue. Several applications can be conceived of that can use this information. Museums can use such a technology to build location-aware hand-held devices. As an user walks into a room, holding such a device, information about the exhibits in that particular room flashes on the screen. Similarly, a system which can do location detection outdoors can be used for displaying context-sensitive information and advertisements to users in amusement parks, such as Disneyland. Indoor location aware systems may also be used to do guide users as well as robots in large and complex buildings, depending on the available granularity of the information.

One of the most popular wireless LAN systems which is currently in use in offices these days is the IEEE 802.11. A typical IEEE 802.11 Wireless LAN installation consists of two components:

1. A number of clients or mobile devices which communicate through radio

2. A set of Access Points which are fixed in location, and which act as bridges between the clients and the rest of the network, possibly through wire.

As a client moves from one place to another, it can associate or disassociate with Access Points within its range.

In our project, we attempt to do location detection in an indoor environment based on the IEEE 802.11 technology. A big advantage of choosing 802.11 is that we need no extra equipment; the system uses components which are expected to be a part of a normal office building with a Wireless LAN installed. This helps in making the system more cost effective than a system which uses extra sophisticated hardware devices.

Our report is organized as follows. The current section provides an introduction to the problem and discusses its applications. The next section discusses the background of the problem and some related work that has been previously done in this area. Section 3 describes our experimental methodology in detail, and Section 4 presents the results that we have obtained. Finally we present our conclusions in Section 5 and discuss the scope for future work in Section 6.

2 Background and Related Work

One of the most popular Wireless LAN systems in use today is the IEEE 802.11. First proposed in 1997, this standard is different from most other MAC layer standards in the sense that it uses more than one, very different physical layers. The original IEEE 802.11 was designed to work with three different physical layers. These are

1. Direct Sequence Spread Spectrum (DS-SS), operating in the 2.4GHz ISM (Industrial, Scientific and Medical) band
2. Frequency Hopping Spread Spectrum (FH-SS), operating in the 2.4 GHz band
3. Infra-Red, operating in a wavelength between 850 and 950 nm and requiring an un-obstructed line of sight

Out of these, the DSSS is the physical layer most frequently used, and it is the layer which we are using for the purpose of our experiments. Hence we will limit our discussions to this physical layer only. DSSS works by spreading a signal over a wide range of the 2.4 GHz frequency band. The available bandwidth is divided into 11 subchannels, each having a spread of 22 MHz. The data rate obtained depends on the type of coding used - it is 1 Mb/s with DBPSK modulation, 2 Mbps with DQPSK modulation and upto 11, 5.5 Mb/ps with CCK modulation. The maximum transmit power in this range can be 100 Mw. The available channels and their frequency ranges are shown in Figure(1)

The most common architecture of a 802.11 system consists of a set of clients which are mobile, and a number of fixed Access Points, possibly tuned to different channels. The whole setup, that is, the clients along with the Access Points,

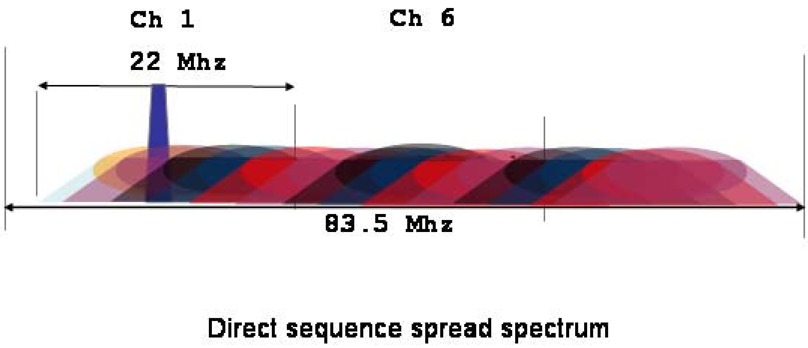

Figure 1: Channel frequency ranges in 802.11

is together called an Extended Service Set or ESS. The Access Points act as bridges between the mobile clients and the rest of the network. These are most commonly connected through Ethernet with the wired part of the network. A mobile client, to be able to send data, has to first associate and authenticate with an Access Point. All data sent and received by a client has to pass through its associated Access Point.

There are essentially two ways in which such a technology can be used to do location detection.

1. By measuring the time delay between transmitting a packet at the Access Point and receiving it at the client

2. By exploiting the fact that at different locations the signal strength received from the various Access Points is different

The first alternative would require highly synchronized clocks at the sender and receiver; hence it is impractical from an implementation point of view. The second approach, however, requires investigation. It is the second approach that we have adopted in our project. There are again two ways in which we could use the signal strength information. One way would be to use some theoretical signal propagation model and our information about the geometry of the building to fit some equations which gives the signal strength. Such an approach however is difficult to take; also it is rather unlikely to work in practice, except in some ideal or highly simplistic cases. The second way, which we actually adopt, is to build a database of signal strength data for a particular location and then use this training data to perform classification into various locations.

Similar work has been done previously by Bahl and Padmanabhan [BP], who have also used IEEE 802.11 Access Point signals to detect the location of a user. In their paper they have presented two approaches to solve the problem. The first one is an empirical method, in which they have built a database of the

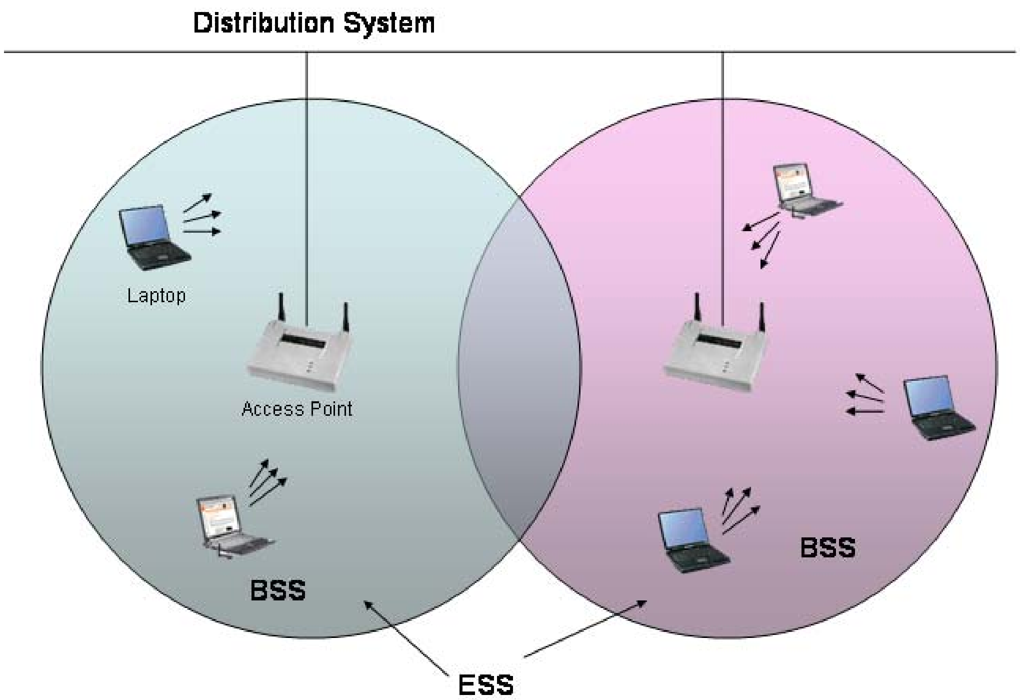

Figure 2: Architecture of a Typical 802.11 System

signal strength of three access points at various points inside a building, and used this data to determine the location of a mobile user. In the second approach, they tried to model the radio signal propagation inside the building by taking into consideration the walls and objects in the building. For the propagation modeling they have used a modeling scheme *Floor attenuation factor* along with some corrections using *Wall attenuation factor* [SR92]. But it was found that the results were better in the case of empirical model, where they used a nearest neighbor classifier. This can be primarily attributed to the complexity involved in accurately modeling radio propagation inside a building where there is a large number of movements of human beings and random disturbance.

Our approach is like their empirical approach. However, we do a more comprehensive study of the data and employ more sophisticated classifiers which take into account the distribution of the data.

3 Method

The solution method adopted by us is to build a database of signal strengths at several different locations in a building; we then use this data to classify a test sample into one of these locations.

This method, however has a number of challenges. The signal strength as received by a receiver at a spatial point is not constant, but varies with time. The variation in the signal strength is due to several miscellaneous factors - movement of air, change in temperature, movement of people, and other disturbances. The signal strength also varies with change in the orientation of the receiver's antenna with respect to the senders antenna. In addition, the radio used in the normal 802.11 LAN Cards found in the market are of poor quality and have some inaccuracies in their measurements of the signal strength. All these factors add up to a variation in the signal strengths at a place which is of the order of 5 to 7 dBms. A sample of signal strength information collected in the ground floor of the CC building where three Access Points have been installed is shown in Figure (4). Hence we can detect locations using this data, only if the variation due to other factors is much smaller than the variation in signal strengths due to change in location.

However if we look at the signal strength distribution of two locations which are some distance (20 m) apart - see Figure (3) - we can see that for these two locations, the signal strength distributions for all the channels are quite different and have little overlap. This suggests that the approach we are using may have some promise.

3.1 Experiment

To test the method we just outlined, we performed some experiments. We chose as our experimental test-bed, the Ground Floor of the Computer Center Building. Three Access Points were installed at three points in the CC - the CC Head Office, the Linux Labs and the Sun Labs. These Access Points were

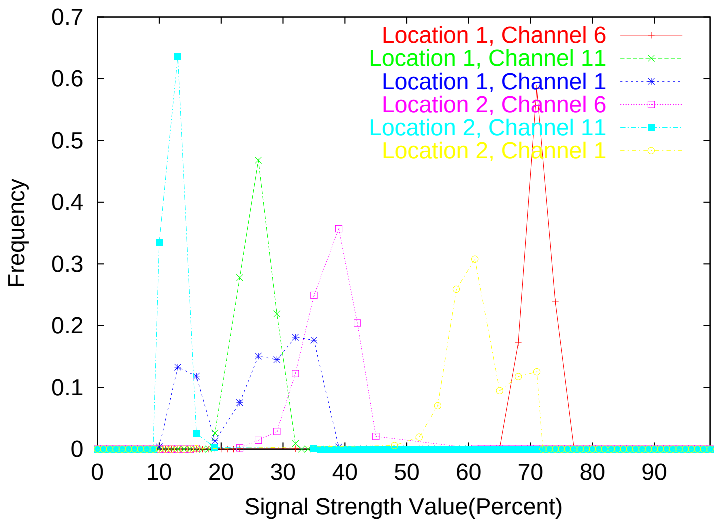

Figure 3: Frequency Distribution of Signal Strengths for two locations at a distance of 20 m

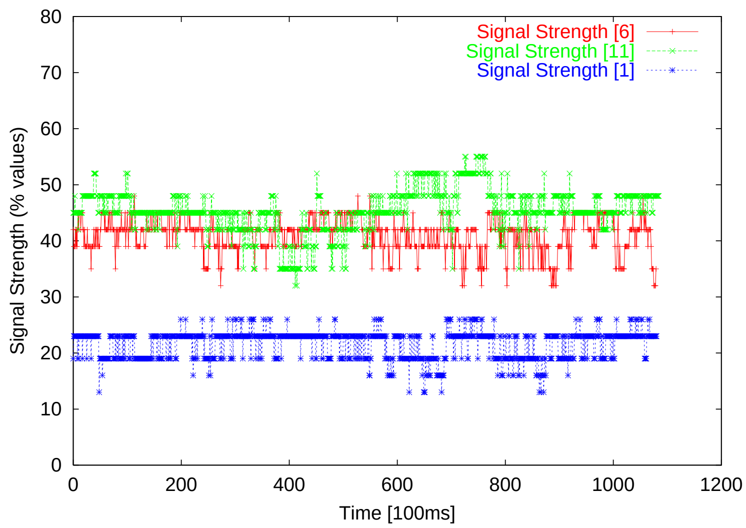

Figure 4: A sample of signal strength data recorded in the corridor in the CC Ground Floor

tuned to Channels 1, 6 and 11 respectively. The locations of these points has been shown in the Figure (5).

Our data collection system comprised of a laptop running Windows 2000. The Laptop was equipped with a Lucent Orinocco Wireless LAN Card. We found that unfortunately the routines that extract the RSSI (Receiver Signal Strength Information) from the card are proprietary information which is not divulged by Lucent. Similar is the case with the Cisco PCMCIA cards. Hence we were forced to resort to Windows utilities for signal strength measurement purposes. For the purpose of our data collection, we used an Airopeek Sniffer utility. The utility works by switching to a set of given channels at user defined frequencies, and capturing all signals that it gets on that particular channel. The signal strength information is recorded by the Sniffer as a percentage value; this value is obtained by the Sniffer by directly querying the card hardware. We found by experiment that 2 percent signal strength as shown by the Sniffer amounts to 1 dbm. Since our Access Points are set at channels 1, 6 and 11, we configured the Sniffer to cycle through the channels 1, 6 and 11 and log the data in these channels. The cycling frequency was set to 3 seconds, which means the Sniffer would switch channels once a second.

At each point, we collected the data for four orientations - North, South, East and West to study the effect of receiver orientation on the data. Data was collected for 5 minutes at a particular orientation per spatial point. We collected data for 19 different training points. We collected data for each point for different times of the day for two different days, at an interval of a few weeks. Data was also collected at neighborhood points of these sample points. After collecting the data, it was uploaded to a postgre server by HTTP upload. To automate the whole process, we wrote a few utilities on the client as well as on the server side. The client utility would run the Sniffer, collect the data for a predefined interval of time, then stop the Sniffer and upload it to the HTTP server. The server side scripts would receive the data via HTTP, and after parsing it, insert it into a database.

3.2 Observation

Analysis of the signal strength data reveals that the data is consistent for almost all spatial points - see Figure(6). Some points however, do show unusual variation.

We found that out of 19 points at different places in the CC, two points in the IBM Labs had a lot of variation in the signal strengths.

Surprisingly at a point just outside the IBM Labs on the corridor, the signal strength data was rather stable.

The signal strength data at the corridor point is shown in Figure(6), whereas a graph of the signal strength data at a point in the IBM Labs appears in Figure(7).

Observation of the signal strength information collected at different orientations shows that in most cases, the data remains the same on an average with

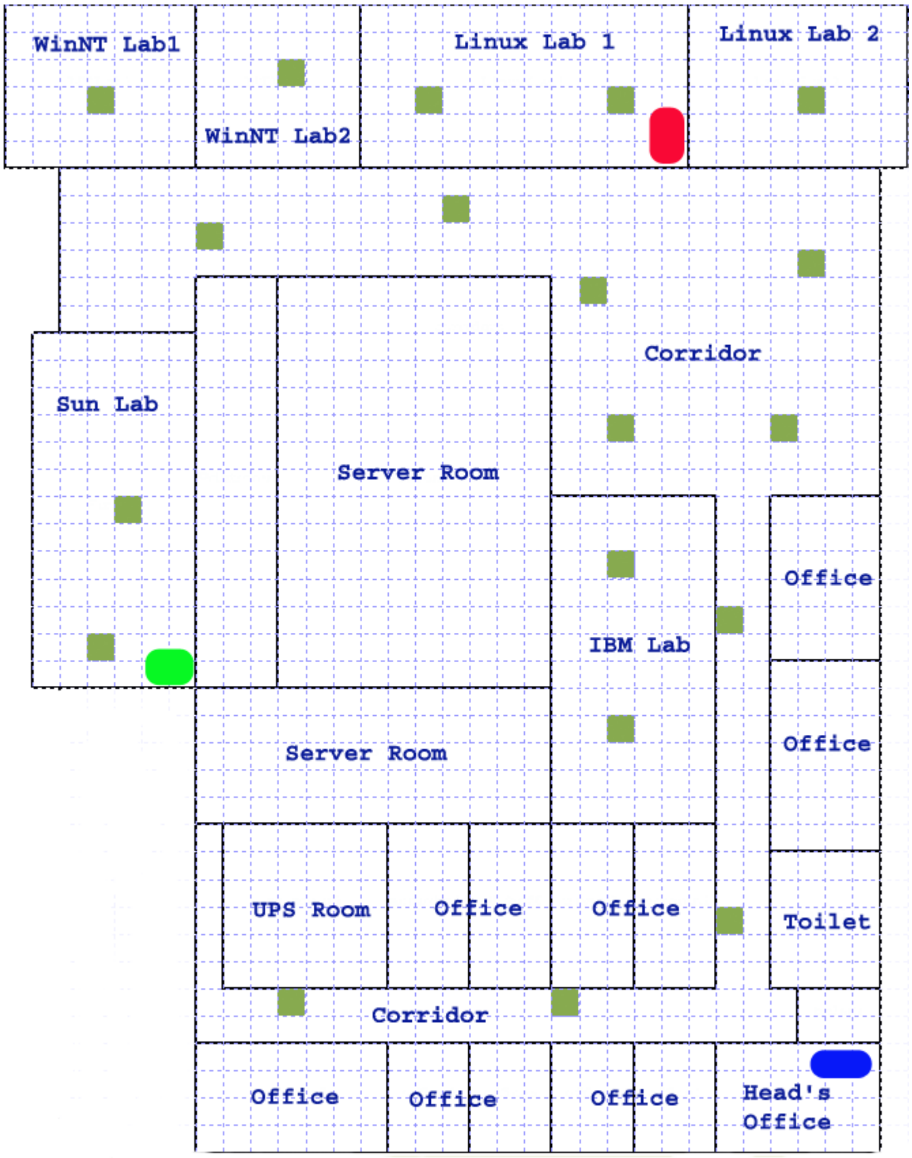

Figure 5: Map of CC showing the Access Points and Data Collection Points. The Ovals indicate the positions of the AP's and the colored squares are the points at which data is collected

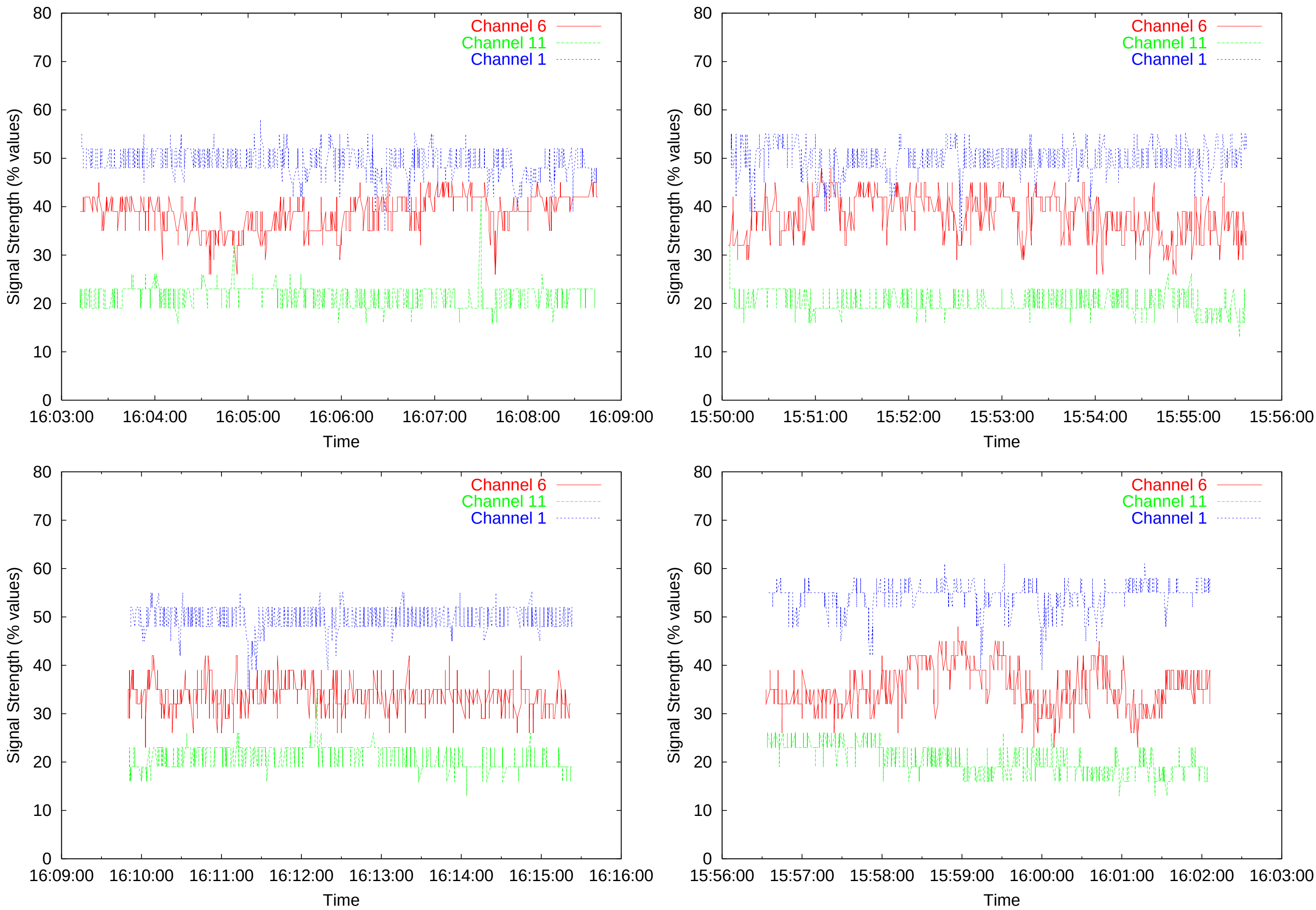

Figure 6: Consistent data - Signal Strength data in the corridor outside the IBM Labs. Graphs in order from the upper-left - North, South, West and East. X axis - time, Y axis - signal strength value in percent as reported by Airopeek

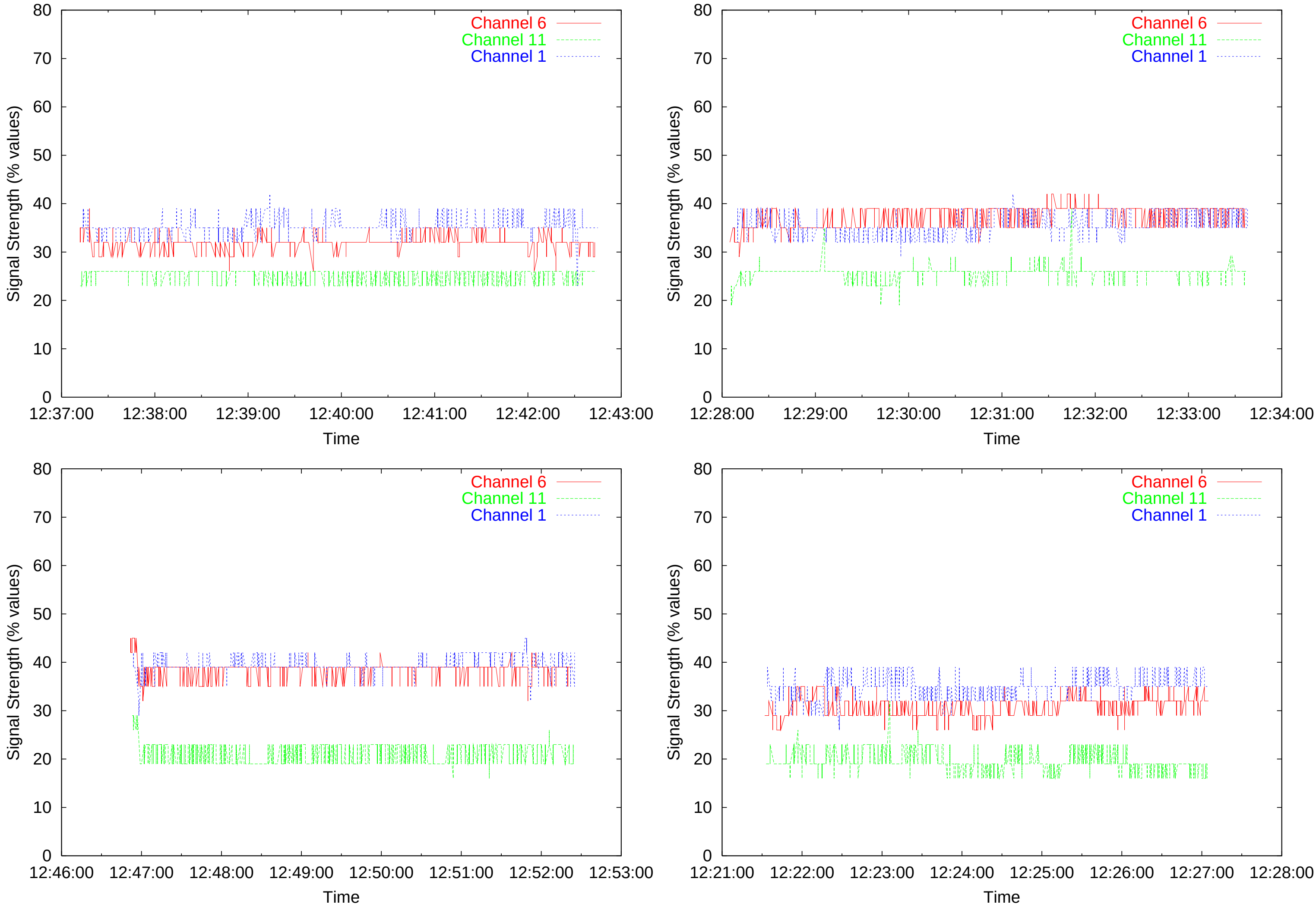

Figure 7: Inconsistent data - Signal Strength data in the IBM Labs. Graphs in order from the upper-left - North, South, West and East. X axis - time, y axis - signal strength value in percent as reported by Airopeek

changing orientation. However, some variation has been observed in few other cases. Analysis of those particular cases show that this occurs when the body of the user obstructs a particular direction from which the signal is coming. It was also observed that the data collected while facing the East is very similar to data collected while facing the West, and similarly for North and South. This happens even if the data obtained while facing North is somewhat different from the data obtained while facing East. However, the overall variance in the signal strength information is quite less across orientations (about 4 dbm) than across different locations. Hence for the purpose of our classification, we do not take into account the effect of orientation.

3.3 Preprocessing

After the data is obtained, it is preprocessed to remove the CRC error packets. Preprocessing is also done to remove the outliers from the training data. A data sample is said to be an outlier, if it lies two standard deviations beyond the mean of a given channel at a given location. It was found that very few (about 0.1%) of the total number of sample points are outliers.

3.4 Classification

Our main problem is basically a classification problem. The training set is a sample of the signal strength information at various points in the CC. The test set is a sample of signal strength information at some unknown location within the CC, possibly at a different time of the day or at a different day of the week. Our problem will be to classify the test sample into one of the predefined classes, based on the information obtained from the training data.

We have employed three methods for the purpose of classification. The first of these is a nearest neighbor classifier; the second is a back-propagation neural network. The third classifier is a modification of the nearest neighbor classifier, based on the distribution values. In this section, we first present a brief overview of these classifiers. In the next section, we describe the results that were obtained by using these classifiers.

3.4.1 Nearest Neighbor Classifier

The first classifier that we used is a nearest neighbor classifier. Consider a sample of the signal strength information obtained from the system for a given time interval. The sample will contain signal strength information corresponding to the three channels or three access points. The 3-tuple (c_1, c_2, c_3), denoting the average of the first, second and third channels, respectively, can be thought of as point in 3-D space.

In this classifier, using the training data, we first calculate the mean of each channel $M_j = (c_{j1}, c_{j2}, c_{j3})$ values for each location. This constitutes the location profile of a given location. Now given a test window of samples, we calculate the mean of the three channels $M' = (c'_1, c'_2, c'_3)$ corresponding to the window. Then our classifier would determine that the test data belongs to location j if

$$dist(M_j, M') <= dist(M_k, M'), \forall k \neq j \tag{1}$$

In other words, our classifier chooses that class whose center is nearest in feature space to the mean of the current data, where nearest is defined in terms of some distance measure. The distance measure we use is the Euclidean distance.

3.4.2 Neural Network

Our second classifier is a Back-propagation neural network. A neural network is some sort of a "black box" information processing unit, which is widely used for the purpose of classification of the data. But before we go into the details of this classifier, let us first explain in brief the basics of a neural network.

A neural network consists of a number of neurons. A neuron is an information-processing unit, which has the following components:

1. A set of input links, each of which is associated with a synaptic weight w_i
2. An adder for summing the input signals, weighted by the respective synaptic weights
3. An activation function, usually nonlinear, for limiting the amplitude of the output of the neuron

Thus a set of input signals coming in through the input links of a neuron are weighted by its synaptic weights, added up and finally passed through the activation function to get the output. Figure(8) shows a model of a neuron.

A neural network is nothing but a collection of neurons arranged in a particular manner to form a network, where the output of some neurons feed into the input of some others and so on. A Back-propagation neural network is basically a multilayer feed forward neural network. Such a network typically consists of several layers - a set of source nodes comprising the input layer, one or more hidden layers and an output layer of neurons. The term "feed forward" indicates that input connections go only in one direction, that is, from input to hidden, hidden to output etc, and not the other way round.

Training of such a network is done by adjusting the synaptic weights such that given a particular input, the network provides a particular output. For a Back-propagation Network, the training is done by using an iterative algorithm called the error back-propagation algorithm. For details of this algorithm, see [Hay99]

The Back-propagation neural network we used had 3 inputs, one for the mean of the signal strengths of each channel, in the given time frame. It had a hidden layer with 20 nodes and an output layer consisting of 19 nodes, one for each location in our training set. The output of the network is thus a 19-dimensional vector, which gives us the probability of the input data belonging to each class.

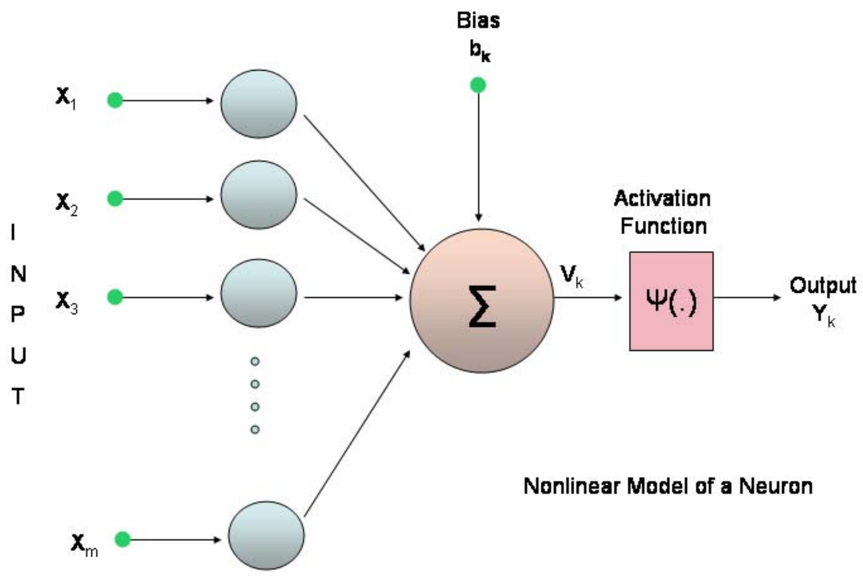

Figure 8: A Neuron

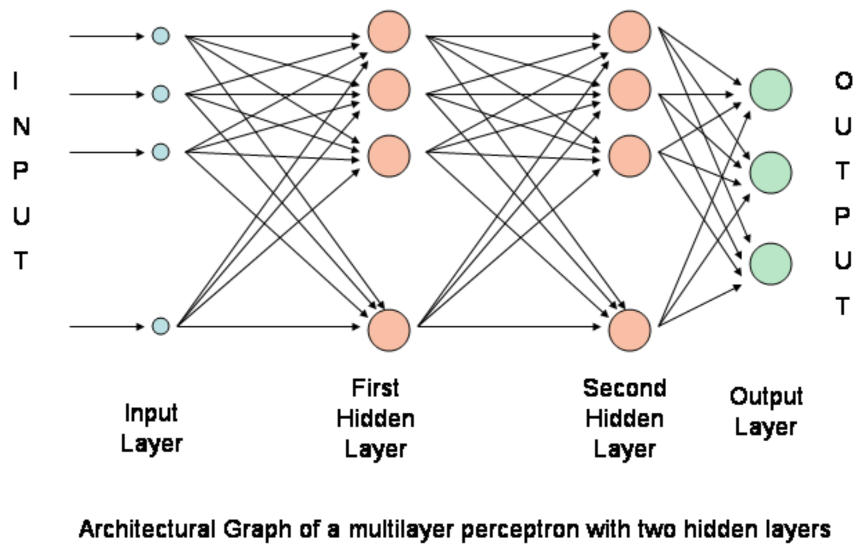

Figure 9: Architecture of a Typical Multilayer Feed Forward Neural Network

3.4.3 Modification of Nearest Neighbor Classifier

In addition to the two above classifiers, we have used a modification of the nearest neighbor classifier to incorporate unable to classify values and the effect of distributions. In this case, the location profile for a given location consists of the values $M_j = (c_{j1}, c_{j2}, c_{j3}), S_j = (s_{j1}, s_{j2}, s_{j3})$. M corresponds to the mean of the data samples for the first, second and third channels respectively, whereas S corresponds to the standard deviation values for these three channels. Given a test window of samples, in this case also, we calculate the mean of the three channels $M' = (c'_1, c'_2, c'_3)$ corresponding to the window. Then our classifier would determine that the test data belongs to location j if M' lies within the rectangular parallelepiped with M_j as center and $4s_{j1}$, $4s_{j2}$ and $4s_{j3}$ as the length of the sides, respectively. More mathematically, the condition reduces to

$$c_{j1} - 2s_{j1} <= c'_1 <= c_{j1} + 2s_{j1} \tag{2}$$

$$c_{j2} - 2s_{j2} <= c'_2 <= c_{j2} + 2s_{j2} \tag{3}$$

$$c_{j3} - 2s_{j3} <= c'_3 <= c_{j3} + 2s_{j3} \tag{4}$$

If a sample point M' lies within no such region, we say that we are unable to classify it. If a sample point lies within more than one such region, we break the tie by choosing the region whose center is nearest to M'.

4 Results

4.1 Simulation Experiments

To assess the performance of the classifiers, we performed a set of simulation experiments. Each simulation consists of 2000 trials. In each trial, we choose a random test location; for that location, we pick up observations from a random offset in the test data file for k consecutive seconds. These observations are now fed into the respective classifier to find if the location is classified correctly. The value of k is varied from 3 to 180, in units of 3. k is always a multiple of 3, as the Sniffer cycles through the 3 channels 1,6 and 11 once in 3 seconds.

4.2 Error Measures

To measure the severity of the failures, we also introduce a quantity called *Error Distance*. This is defined as the spatial distance between the original point to which the data belongs, and the point which is reported by the classifier. The distance here means the Euclidean distance in 2D. We calculate two measures of error distance, each depicting a particular trend in the error of a classifier. The first variant, call it *Average Error Distance* is defined as the sum of all the error distances averaged over all the runs of the simulation. The other variant, called *Residual Error* averages the total error distance over the total number of misclassifications.

More mathematically, let N_T be the number of trials in a simulation, and N_E be the number of trials for which the classifier yields a wrong value of location. Then the *Average Error Distance* is defined as

$$A_e = \frac{\Sigma_{i=1}^{N_T} dist(l_i, y_i)}{N_T} \tag{5}$$

and the *Residual Error Distance* can be defined as

$$R_e = \frac{\Sigma_{i=1}^{N_T} dist(l_i, y_i)}{N_E} \tag{6}$$

Here l_i is the original location of the test data used for the i^{th} trial and y_i is the value of the location obtained from the classifier for the i^{th} trial.$dist(x_1, x_2)$ denotes the spatial distance between the 2D points x_1 and x_2.

In case the classifier admits of "unable to classify" values, these definitions are slightly different. We sum the error distances over those trials only for which the result is either a correct or an incorrect classification, that is the output is not a "unable to classify". We also compute the *Average Error Distance* by dividing the total error distance by $N_T - N_C$, where N_C is the number of trials for which "unable to classify" output has been obtained.

4.3 Simulation Results

Our simulations show that an accuracy of about 90% was obtained, when the minimum distance between the points is 3.12 meters. Figure (10) shows a plot of the accuracy of detection versus the time frame for which data is collected for the three different classifiers. Surprisingly, all three classifiers gave similar accuracies. This suggests that the inaccuracies are not due to an improper classifier, but because the distribution of the data is such that there is some overlap between the signal strength distributions for some locations. Another point to be noted in the results is that the accuracy does not change significantly with the increase in sample size.

Of the three classifiers which we have chosen, the Neural Network and the Modified Nearest Neighbor Classifier admits of "unable to classify" values as well. We found that out of all trials which did not classify correctly, about 25% of trials gave a verdict of "unable to classify" for the neural network. For the modified nearest neighbor classifier, this figure was about 50%.

The Average and Residual Error Distances for the three classifiers are plotted in Figures (11) and (12). These also give similar values for all three classifiers, and do not vary significantly with sample size. It is seen that the values of Residual Error Distance for the Nearest Neighbor Classifier is higher than those for the other two classifiers, and it also increases with increasing sample size. This can be attributed to the fact that this classifier has no "unable to classify" values; hence if the test data at a location is very much different from the training data, it gets classified to some arbitrary location which may be far away, thus contributing a large value to the total error distance. The other two

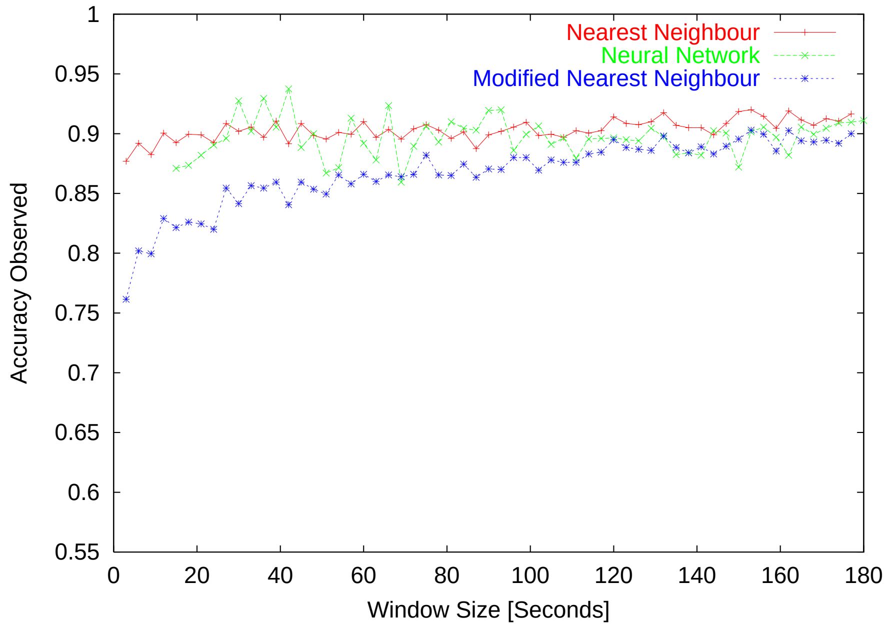

Figure 10: Variation of Accuracy of detection with Sample Window Size, for each Classifier.

classifiers would however declare such a trial as "unable to classify", and would not take it into account while calculating the error distance measures.

Figure (13) shows the Average Error Distance when the test data is collected from points which are distinct from, but are near to the training points. As one can see, the Average Error distance in this case lies mostly between 2 and 3. This indicates that it is possible to perform location detection in a building by collecting data only from a set of sample training points, and then using this data to perform classification.

5 Conclusion

From our results, it can be concluded that position can be determined in an indoor environment using IEEE 802.11 Access Point signal strengths. The accuracy of location determination depends on how close the two points to be distinguished are. Because a major component of the signal received in an indoor environment is due to reflection, and not due to direct line of sight, the results are also expected to depend on the geometry of the place. A building with a large number of small rooms is expected to give better results than a building with large hallways. In our case, when the minimum distance between two training points was 3.12 meters , the accuracy obtained was upto 90% in the CC Ground Floor. This was in a rather hostile environment, as the CC

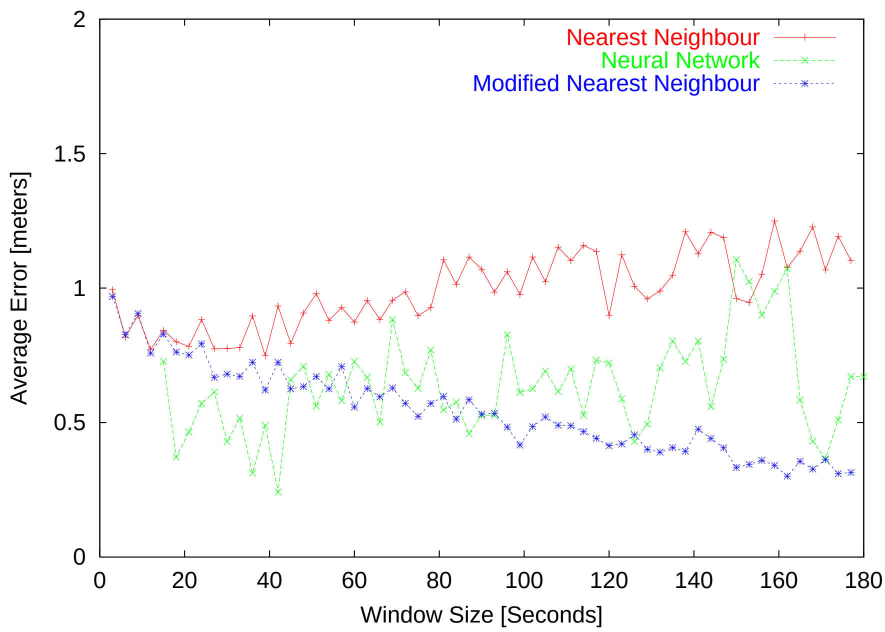

Figure 11: Variation of Average Error Distance with Sample Window Size for each Classifier

Ground Floor has only a small number of large rooms. In this case, the signal strength distributions obtained at close-by locations have high overlaps - see Figure(14), which shows the frequency distribution of signal strength values at two locations at a distance of 3.12 meters. We can expect that better accuracy will be obtained in an office building with a large number of small rooms.

Our results also show that the percentage accuracy of classification does not vary significantly with the type of classifier used. This suggests that the errors in classification are not due to the use of an improper classifier, but due to the distribution of the data itself. The signal strength distribution is such that there is some amount of overlap between the distributions of signal strength values at two spatial locations - which is giving rise to the inaccuracy of classification.

A third conclusion that can be obtained from our experiments is that the accuracy does not change significantly with the increase or decrease in the time frame for which the test data is taken. This suggests that we need to take the sample of signal strengths at a given location for only a small amount of time to achieve reasonable accuracy. This makes our method of position detection quite practical. This also means that this method can be applied to indoor location tracking - that is, determining the position of a moving user in a building, provided the user is not moving too fast.

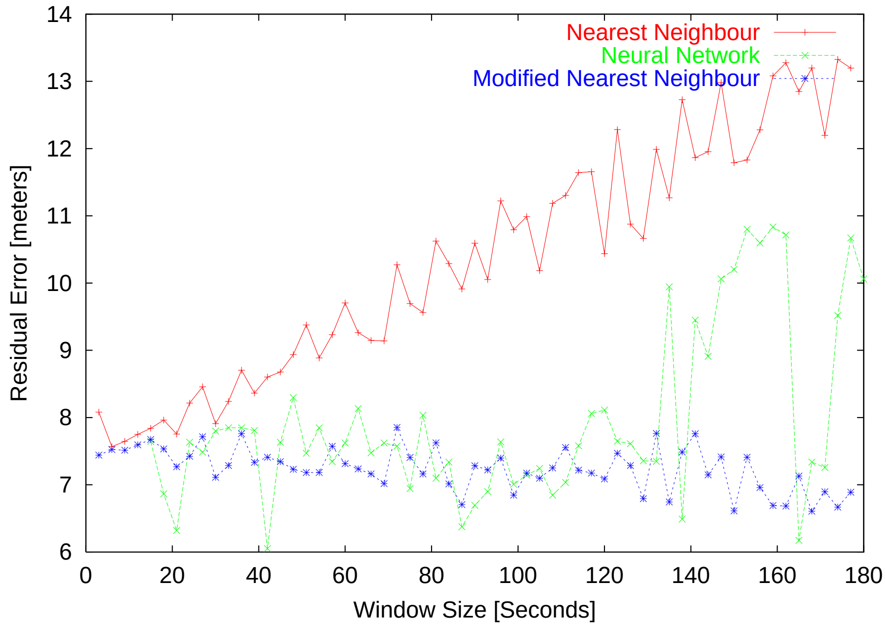

Figure 12: Variation of Residual Error Distance with Sample Window Size for each Classifier

6 Future Work

There is further scope for work in this area. Our work takes a set of discrete points and tries to determine if the current location of the user is one of these points. The next step to this classification would be to perform interpolation. That is, given a test sample data, the system would try to pinpoint its exact location by performing interpolation between the spatial points in the training set.

Another idea that can be further pursued is classification of the data not in the time domain, but in some other feature space. For example, one can try to classify the data in the frequency domain by using the Fourier transform. Such a transformation may be able to yield better results than simple classification of the data in the time domain.

7 Acknowledgements

We are grateful to Dr Dheeraj Sanghi for his constant encouragement and guidance throughout the project. We would like to thank Dr Pravin Bhagwat, for his invaluable suggestions and constant guidance, without which this work would not have been possible. Our thanks to Dr Rajat Moona and Dr Deepak Gupta, members of our BTP committee for their constructive suggestions and ideas.

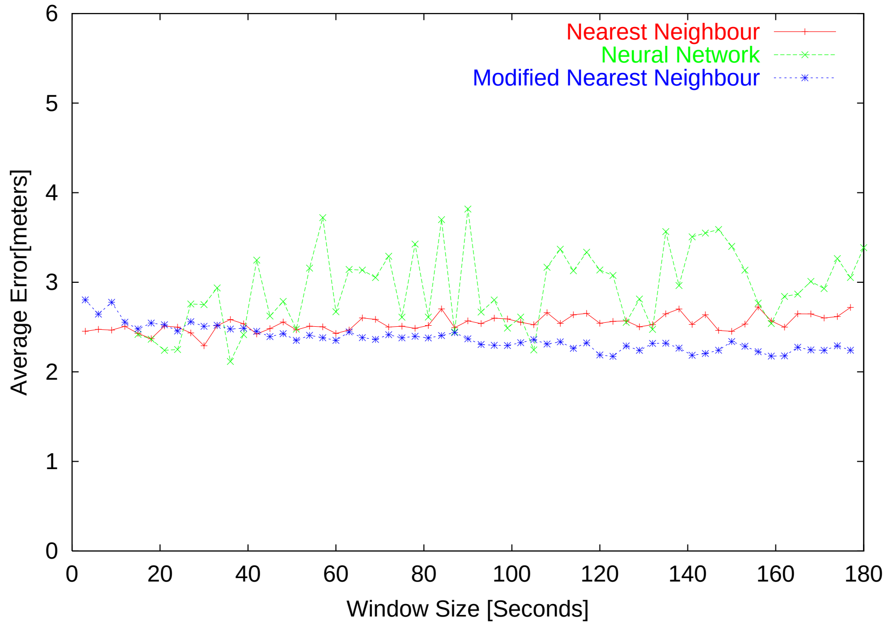

Figure 13: Variation of Average Error Distance with Sample Window Size for each Classifier for data collected near (but not exactly at) the training points

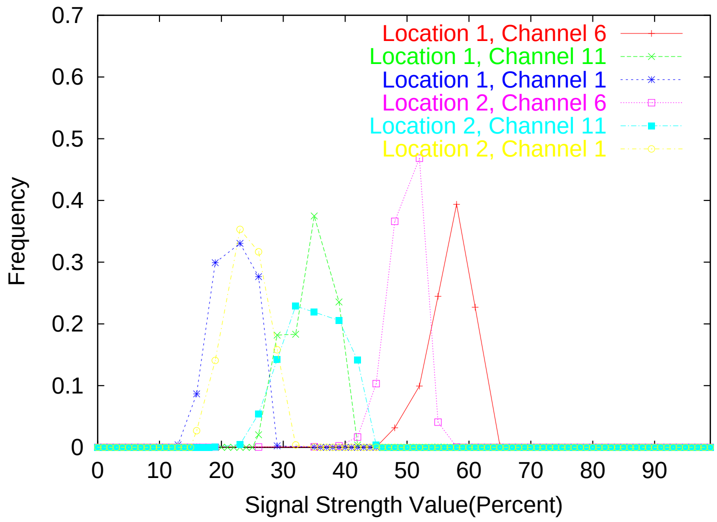

Figure 14: Frequency Distribution of signal strength values at two locations at a distance of 3.12 meters

Finally, we would also like to thank our colleagues Utkarsh and Ambuj for their help in developing our data collection utility.

References

[BP] Paramvir Bahl and Venkata N Padmanabhan. Radar: An inbuilding rf based user location and tracking system. *Microsoft Research Technical Report*.

[Hay99] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Pearson Education Inc, 1999.

[SR92] S. Y. Seidel and T. S. Rapport. 914 mhz path loss prediction model for indoor wireless communications in multifloored buildings. *IEEE Transactions on Antennas and Propagation*, 1992.