**Optimized Protocol for IOT's Data Transmission Using Machine**

**Learning**

### By

**M.Awais Ali**

01-133162-084

**Aizaz**

01-133162-007

## Supervised by:

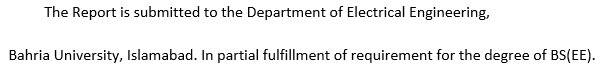
Ammara Nasim

**CO-SUPERVISOR:**

Dr. Saleem Aslam

****

{Session 2016-20}



****

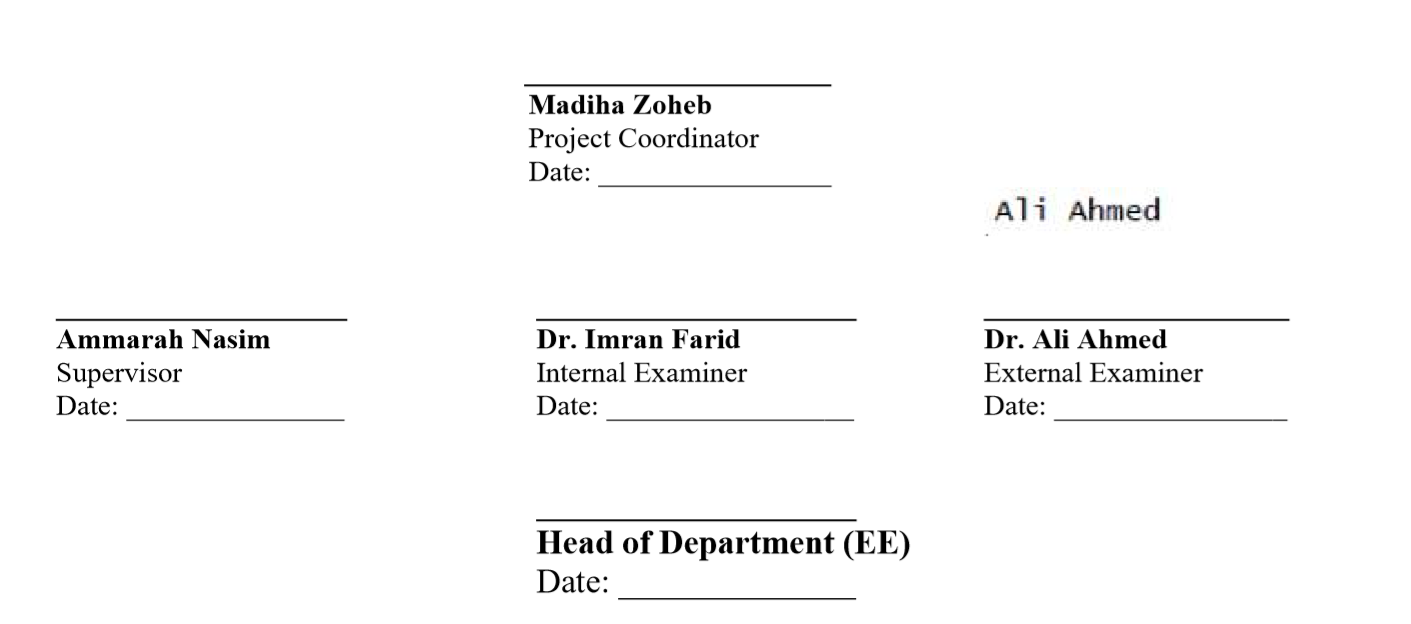
**Bahria University Islamabad**

**Department Electrical Engineering**

**Dated: 15-07-20**

**CERTIFICATE**

**We accept the work contained in the report titled (Optimized Protocol for**

**IOT's Data Transmission Using Machine Learning) as a confirmation to the required standard for the partial fulfillment the degree of BEE.**

For my parents,

who supported a little warrior through thick and thin. I owe you everything.

For my teachers,

who define everything, without them I am stuck between nothing and something.

**Acknowledgements**

Everyone will tell you it is not easy to complete a final year project, but most don’t mention how lonely it is. I could not have done it without these people. Dr. Taimoor Hassan, if I have a chance to choose a mentor, surely it will be you. Dr. Saleem Aslam, who guided and inspired me throughout this project. Dr. Adil Ali Raja, for advice and counseling. Murtaza Shah, for motivating me whenever I was about to crash. Madam Ammara Nasim, for helping me during difficult time.

-M. Awais Ali

I’m extremely grateful to my parents, their unrelenting support made me capable of taking on this project and I would like to express my deepest appreciation to my partner without whom this project would not have been possible. I would like to extend my sincere thanks to our Supervisor Madam Ammara Nasim, her experience, suggestions and invaluable insight provided guidance and support we needed to complete this work.

-Aizaz Jabbar

**Abstract**

Internet of Things (IoT) based sensor networks have gained popularity in the recent years and they become vital for supporting high data rate real time applications. By the end of 2020 there will be 31 billion IoT devices around the world. To achieve efficient data transmission each IoT node has to gain an understanding of recent time and spectral feature of channel in order to increase the throughput. The literature has proposed different methods such as channel allocation and channel quality measurement protocol for multiple channel sensor networks. As far as we know there are few protocols that can adapt and learn with respect to the changing channel attributes in IoT network to maximizing data transmission and channel throughput. We propose an automated self-learning and modifiable protocol that can automatically transmit multiple user data efficiently through recognizing channel frequency and time features. The proposed protocol is novel in a way that it can develop understanding and grasp itself to changing network dynamics such as network density and amount of data to be transmitted. This is achieved by constantly extracting well defined features from a network. Best channel selection according to time and spectral characteristics is done by utilizing these features. Each node is stocked with non-linear support vector machine with Gaussian radial basis kernel function classification model to make the decision for using Time based partitioning or Frequency based partitioning. This protocol shows promising results in increasing network density. It uses the bandwidth efficiently and shows better data transmission.

**Table of Contents**

Certificate i

Dedication ii

Acknowledgements iii

Abstract iv

Table of Contents v

List of Figures vi

List of Tables vii

1. Introduction 1

1.1 Background 2

2. Literature Review 7

1. Requirement Specification 12

3.1 TDM Block 13

3.2 FDM Block 14

3.3 Communication Block 15

3.4 Machine Learning Block 16

3.5 Data Set 17

1. System Design 19
2. System Implementation 24

5.1 System Model 25

5.2 Tools and Technology 27

5.3 Processing Logic 28

1. Evaluation 32

6.1 Results and Evaluation 33

1. Conclusion 39

Reference 42

Appendices 46

**List of Figures**

Figure 1.1 Internet of Things 3

Figure 1.2 Proposed System Model 4

Figure 1.3 Machine Learning Understanding 6

Figure 3.1 Time Slots 14

Figure 3.2 Spectral Partitioning 15

Figure 3.3 Support Vector Machines 17

Figure 4.1 Proposed Multi-resolution channels 20

Figure 4.2 Block Diagram 20

Figure 4.3 TDM Block Diagram 21

Figure 4.4 FDM Block Diagram 21

Figure 4.5 Communication Block Diagram 22

Figure 4.6 Machine Learning Block Diagram 22

Figure 4.7 Proposed Classification System 23

Figure 5.1 Proposed Multi-resolution channels 25

Figure 5.2 TDM Logic Block 29

Figure 5.3 FDM Logic Block 30

Figure 5.4 Cross Validation 31

Figure 5.5 Machine Learning Logic Block 31

Figure 6.1 Classification Accuracy 33

Figure 6.2 Signals from Sender and Receiver 34

Figure 6.3 Multiplexed Signal 34

Figure 6.4 Signal with AWGN 35

Figure 6.5 Signal from Nodes 35

Figure 6.6 TDM Signal 36

Figure 6.7 Signal at Receiver end 36

Figure 6.8 Real Time Feature Extraction 37

Figure 6.9 Real Time Prediction Logic 37

Figure 6.10 Real Time Decision Logic 37

Figure 6.11 Dataset Plot 38

**List of Tables**

Table 3.1 Data Set 18

Table 3.2 Proposed System Performance 37

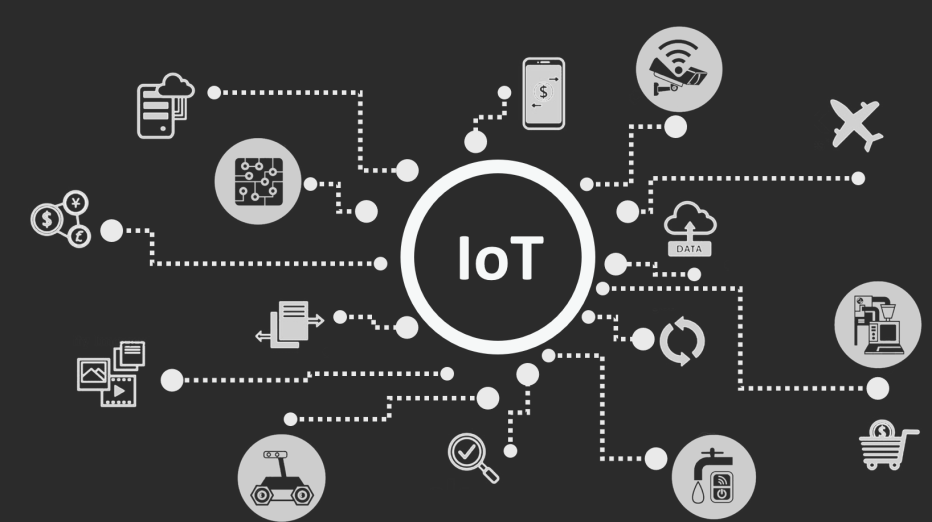
**Chapter # 1**

**Introduction**

* 1. **Background**

Implementation of different applications related to surveillance and health are dependent on’ Internet of Things’ (IoT) based wireless sensor networks [2]. Internet has accelerated the pace of advancements. IoT refers to devices which can communicate with each other with or without human intervention. Figure 1 shows the huge connectivity of different devices with each other with or without human interference. A network having some devices, objects and sensors, having controllers and working by following a protocol is called an IoT network. By 2020 there will be 31 billion active IoT devices around the globe [22]. A research by Gartner predicted that 70 percent of IoT devices will be using cellular network by 2022. A single IoT network consists of two to three sensors at least, but there could be more than hundred devices in a single IoT network. As the sensors inside an IoT network are usually small and they have finite amount of energy, they also consist of a very small processing unit and memory. For effective data transmission inside an IoT build wireless sensor network, one of the vital tasks is to find out the passage connecting source and destination [3]. There is a link between energy consumption and data transmission.

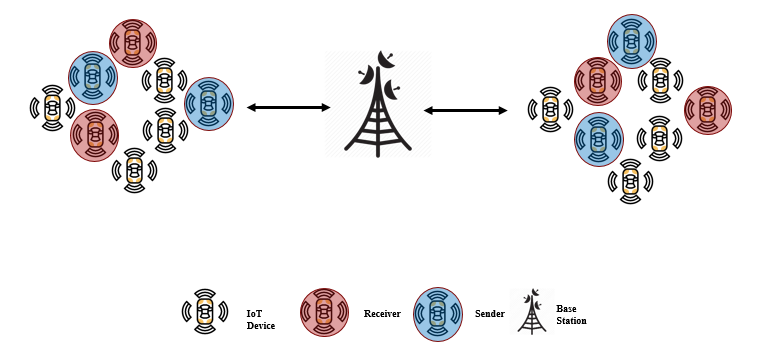
Due to this huge rise in number of IoT devices future networks will be dense. Density of a network leads toward poor transmission of data if proper measures are not taken. Researchers have already put efforts on energy utilization in sensor node to enhance the data transmission [4]. Different methods have been introduced to enhance the data transmission rate inside a wireless sensor network such as multichannel protocol for effective



**Figure 1.1: Internet of Things (IoT’s)**

communication and fixed slots-based hybrid partitioning channel protocol which have some fixed time and frequency-based slots for effective data transmission. Figure 2 shows an IoT based sensor network. Two nodes inside an IoT network are connected to one another through an active track which is the finest accessible channel selected by the deploying advanced protocol.

Once different network metrics and network topological information are passed to the core system which will process these metrics and assign each node a channel according to its need. This technique is different from conventional wireless sensor network data transmission. There is a mechanism for transmission of large amount of data in multichannel IoT network.



**Figure 1.2: Proposed System Model**

If the frequency assigned at the transmitting end and receiving end do not match, then it may cause power dissipation as there is now need for the switching of channels. Further we know that the switching causes delays which may result into loss of data, affecting the performance of the entire system.

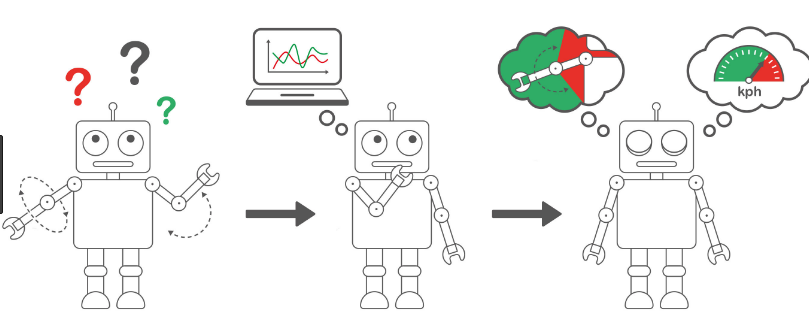
To overcome such scenarios packet-based channel assignment is discarded and stream-based channel assignment protocols are introduced to overcome the performance loss [5].

Many researchers have proposed utilization of fixed time and frequency-based slots for better transmission which are also known as hybrid channel partitioning protocol. However, these protocols are not automated or conscious in the course of learning the dynamic network metric with respect to time. Most of the work has been done on developing a better channel quality index metric measurement protocol for optimum channel selection. We propose an automated protocol that has the ability to transfer multiple user data effectively on the basis of time and frequency functions of the channel. Proposed protocol is distinctive as it is able to learn and improvise itself to growing network compactness.

The proposed protocol employs Gaussian radial basis function as its kernel function for support vector machine. In a multiuser environment for efficient data transmission purposed protocol assign each node a best channel selected upon the recent features of channel frequency and time. The purposed protocol outperformed other solution in means of channel fairness, mean channel blocking probability, and attained classification accuracy of 95%.

Set of features are obtained from an IoT network and then these features are aligned in such a way that they can be treated as machine learning algorithm input. There are several machine learning algorithms, but we deployed support vector machine which is also known as large margin classifier. It is a relatively fast algorithm and better for making optimal decision between high dimension data. Machine learning is method of training a machine on a data so that it can make rational decision.

Support Vector machine is unique because of its kernel function. As SVM with no kernel is also known as linear SVM. Kernels are also known as similarities functions. Our deployed SVM gives binary output in the form of 0 and 1 where 1 represent time-based partitioning and 0 represents spectral based partitioning.



**Figure 1.3: Machine Learning Understanding**

Once the decision is given by trained machine learning model then optimal channel is selected for data transmission.

**Chapter # 2**

**Literature Review**

In the previous chapter we discuss about the background of optimized protocol for IoT data transmission using machine learning. In this chapter we discuss the literature review of optimized protocol for IoT data transmission using machine learning. We compare our work with the previous work related to our project. We also discuss technical differences between our proposed protocol and previously proposed protocol.

Each IoT sensor node has a finite amount of energy. It consists of a small processing unit and memory. Here, the utilization of energy is crucial. Many researchers have proposed different methods for improving the energy utilization in a wireless sensor node for improving the data transmission rate. In order to maintain a high data transmission, it is necessary to improve the energy utilization of a sensor node [6]. Our proposed protocol focuses on the effective bandwidth utilization. Assigning each node, a transmission slot according to its need.

In wireless sensor network, there is a single channel for communication between sensors, but different methods have been proposed which suggest the use of multichannel communication rather than a single channel communication. Apart from this, fixed time based, and frequency-based slots have been introduced. These are known as hybrid channel protocol for better transmission rate. Our proposed protocol does not utilize fixed time based and frequency-based slots rather the numbers of slots are decided at the real time so that there should not be any wastage of bandwidth.

In conventional wireless sensor network each node has its own channel but this method becomes useless when number of nodes increases. As the number of nodes increase there will be increase in the number of channels. Bandwidth is not unlimited. We have a fixed amount of bandwidth. Sometimes the data transmitted by a node is far less than the capacity of the channel which ultimately leads us to the wastage of resources.

To overcome the wastage of bandwidth problem multichannel based IoT network needs some method through which they can select a channel which is suitable for transmitting large amounts of data. Sometimes the data transmitted by different nodes have a priority like some nodes need to send their data prior to other nodes.

There is a study that proposes to use different protocol to calculate channel quality and stability in a territory where we have multiple users [7]. But our proposed protocol is not about measuring the channel quality it is more about selecting the appropriate channel for data transmission. Some of the protocol tend to use single radio per node [8].

Sometimes during the transmission of data there is a difference between the frequency assigned to the receiving and transmitting end. The spectral bands at the receiving end and transmitting end do not match, leading to power dissipation. This power dissipation is usually caused by the switching of frequencies. This frequency switching leads to delays [9]. There is a potential chance that these delays can cause loss of data. In terms of data transmission data loss is crucial as it has a huge effect on the system performance. So, in case of data loss there will be loss in the performance of system as well. To overcome data loss packet-based channel assignment is dismissed and replaced by stream-based channel assignment protocols [10].

A femtocell works through the internet and it provides a small area of coverage. The user signal is transmitted over the internet to a ‘femto’ gateway which sends it out to the base station. Femtocell require internet connection and they have usually small overage and usually allow 4 people to use it simultaneously. Femtocells are also carrier specific. There is research that proposes a possible game perspective for joint resources and power grant that increases the femtocell capacity achievement by reducing the intrusion factor and it tries to maintain microcell performance [11].

There has been addition in the numbers of multichannel protocol during the past years. Some of these protocols have a unique function which is known as a utility function. They also have a performance matrix to keep the record of the performance of different channels. The performance matrix uses the previous details to estimate the topological information of a network. This technique assigns a channel to nodes based on previous instructions which are not useful. This technique is not useful because it does not focus on the channel quality indicator as the response of a channel can easily vary with respect to time [12].

Most the protocol mentioned are not self-aware. They are unable to learn about the changing network metrics such as network density. Most of the work is done on formulating a better channel quality indicator metric measuring protocol for better channel assignment. Instead there is a need for the development of a fully automated channel partition protocol which has the ability to divide the channel into different time based and frequency-based slots according to the need of network.

This automated protocol is not only able to partition the channel, but it also has the tendency to allocate robust spectrum to each node for coherent data transmission in multiple user territory [1]. The partition of the channel is done in two ways:

1. Time based partitioning
2. Spectral based partitioning

If the amount the data that has to be transmitted over the channel is small, then time-based partition is preferred. While in case of large amount of data to be transmitted over the channel then it is better to go with frequency-based transmission. The proposed protocol provides multi hybrid resolution channels which means it can allocate some devices time-based slots according to their need while at the same time some devices will be allotted frequency-based slots as per their need.

Several machine learning algorithms are currently being used for different kind of classification problem. Our proposed protocol uses nonlinear support vector machine. It uses Gaussian radial basis function as a kernel function to achieve better accuracy. In other research papers authors have used k-nearest neighbor, hidden Markov model and linear SVM. A linear support vector machine has no kernel function. The proposed protocol which uses nonlinear support vector machine achieves up to 50% better classification accuracy than the existing techniques being used in the term of classification of channels for robust spectrum allocation [1].

**Chapter # 3**

**Requirement Specifications**

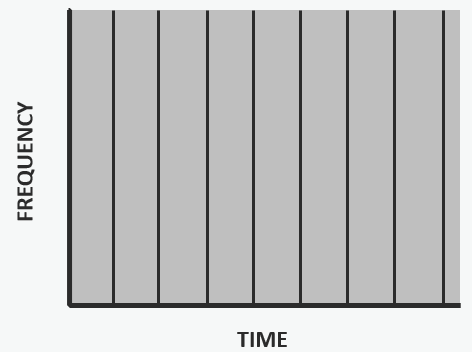
This project is simulation based. Most of coding work is done on MATLAB. MATLAB houses one of the most powerful tools for coding and simulation purposes. We have coded purpose-specific blocks to perform specific tasks.

* TDM Block
* FDM Block
* Communication Block
* Machine Learning Block
* Data Set

**3.1 TDM Block:**

Multiplexing is a technique of sending many signals over a single channel. In time division multiplexing the channel is partitioned with respect to time. It is method of transmitting and receiving signals over a common channel. Every user is assigned some time in which it has the access to send and receive the data over a particular period. Once the allotted time for a user is over then the second user can send and receive data for a respective amount of time. Synchronization plays a key role in time division multiplexing. The transmitter and receiver must be synchronized in order to carry out a successful data transmission.

If synchronization is not working properly it can lead towards ambiguity in the data received. Time Division Multiplexing is effective when we have a smaller number of nodes. For better result the data must have smaller size. Large data leads towards slow data transmission rate in case of Time Division Multiplexing. TDM can be further extended into ‘Time Division Multiple Access’ (TDMA). In TDMA channel is divided into multiple slots based on time division multiplexing. Multiple users can access the channel as we have more than one slot available for transmission. Time Division Multiplexing has several advantages [13]. The problem of cross talk is not severe as each user access the slot at his own time interval. Bandwidth is properly utilized.



**Figure 3.1: Time Slots**

**3.2 FDM Block:**

In ‘Frequency Division Multiplexing’ (FDM) channel is partitioned on spectral basis. Total bandwidth will be divided into different non overlapping spectral bands. In frequency division multiplexing each of the divided spectral band carry a separate signal. Once a spectral band is assigned to a user, he will be the only one to send data over the allotted frequency. So, the single transmission medium will be shared by many users at a same time, but each node will have its own frequency where data transmission is carried out. FDM can also be further extended into frequency division multiple access which has a same concept as time division multiple access but the bottom technique for data transmission is FDM.

FDM is generally more effective when each node has to transmit large amount of data. In case of a greater number of user’s, frequency division multiplexing comes in play. Frequency division multiplexing has its own advantages [14]. There is no need for the synchronization in FDM as each node has a slot specially created for its own use. Further to protect the data from overlapping guard bands are introduced. A small amount of bandwidth is left out between two slots to avoid overlapping of the signal. Cross talk is prominent in FDM.



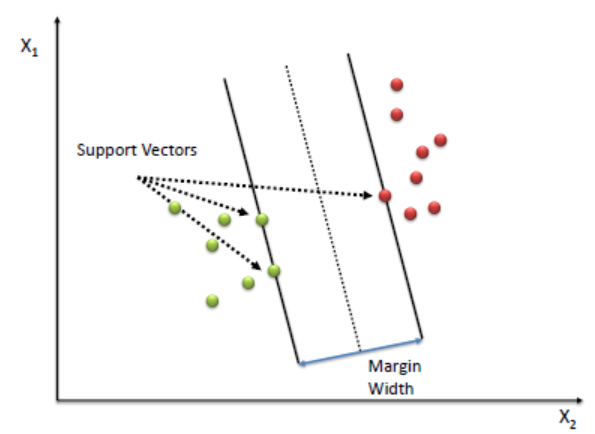
**Figure 3.2: Spectral Partitioning**

**3.3 Communication Block:**

MATLAB is used for the purpose of communication. One laptop plays the role of a transmitter while the second laptop acts as a receiver. Each laptop has MATLAB 2018b installed. For the ease of purpose, communication is carried over the wi-fi. For communication TCP IP protocol is used in the MATLAB. The concept is to wrap the data into TCP IP protocol, but the technique used for transmission is FDM and TDM. Basically, it is not possible to transmit data directly using TDM and FDM by a laptop because there is no TDM and FDM chip in a laptop. Further the work is being done on MATLAB, so it also creates another barrier. First the communication is carried between two different instances of MATLAB using a single laptop. Once that was successful, we moved toward two different devices. One device acts as a client while other acts as server during the process of communication using TCP IP [15]. It is necessary to define buffer size which carries the data. The size of the buffer must be greater than the number of bits being transmitted during the communication to overcome any kind of data loss.

**3.4 Machine learning Block:**

In machine learning block we used ‘Support Vector Machine’ (SVM) algorithm. A model is trained using SVM. To achieve better accuracy Gaussian Basis Radial is used as kernel function. SVM draws a decision boundary between two classes [20]. Output of the SVM is in binary. TDM is represented by 1 and FDM is represented by 0. First of all, data set is divided into train set and test set. Test set is used at the end to find out the trained model accuracy. While the trainset is used to train the machine learning model. K-fold cross validation method is used during the training process of the model.

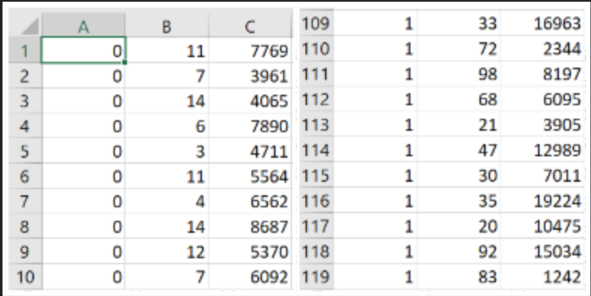


**Figure 3.3: Support Vector Machine**

Different numbers of fold are used K-fold cross validation and best accuracy is achieved at 5 folds. In cross validation test data set is divided into k number of folds [21]. After the model is trained it is tested on the test data set to find out the accuracy of trained model. SVM is a powerful tool when the training data have high dimension. As obvious from its name SVM finds out the support vector inside the data set. SVM tries to find many decision boundaries and at the end it stops when it finds an optimum decision boundary where we have the maximum distance between the two classes. TDM is defined as class 1 and FDM is defined as class 0.

**3.5 Data Set:**

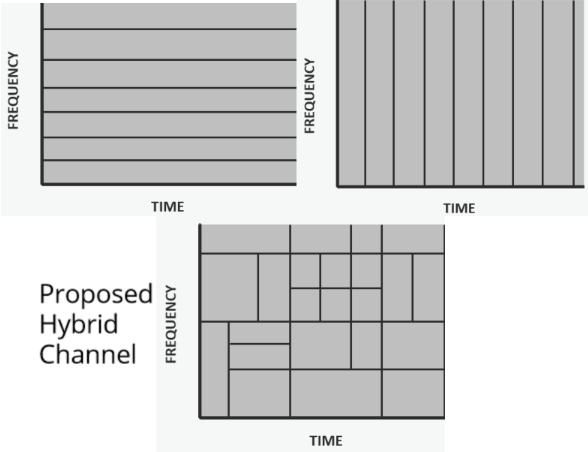
Primary data set is obtained from IEEE [16]. Data set contains 2 features. First feature (f1) is the number of nodes and second feature (f2) represents the number of messages. Data set has a dimension of 200 x 3. First column represents the classes and second column represents number of nodes while the third column represent number of messages. Number of messages are measured as samples per second.



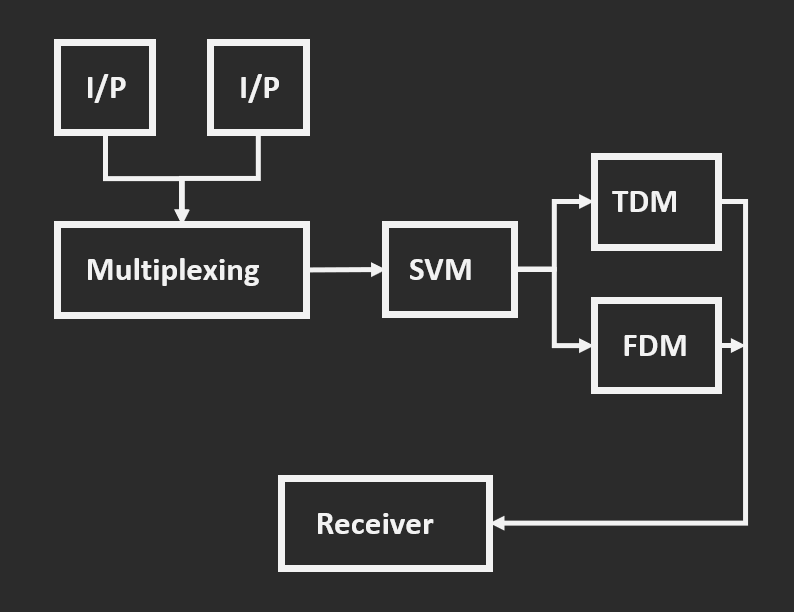
**Table 3.1: Data Set**

**Chapter # 4**

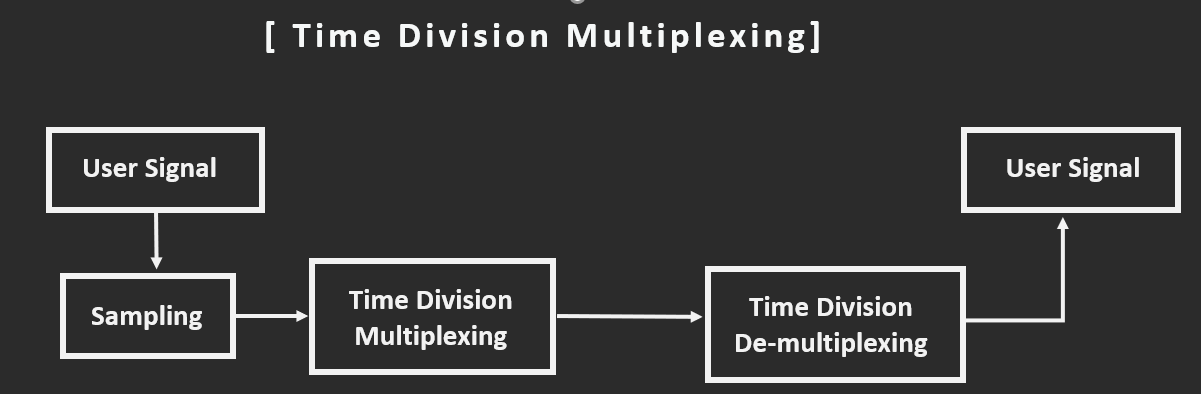
**System Design**



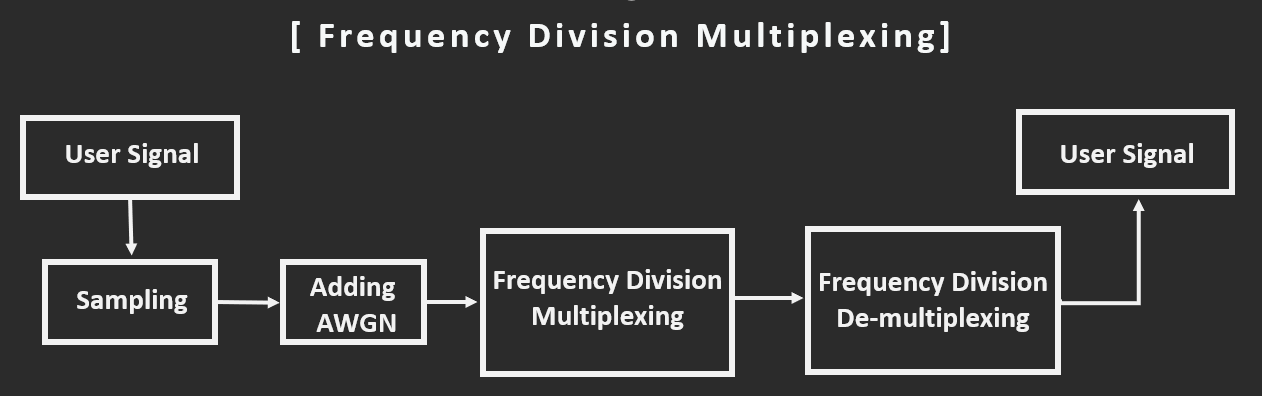
**Figure 4.1: Proposed Multi-resolution Channels**



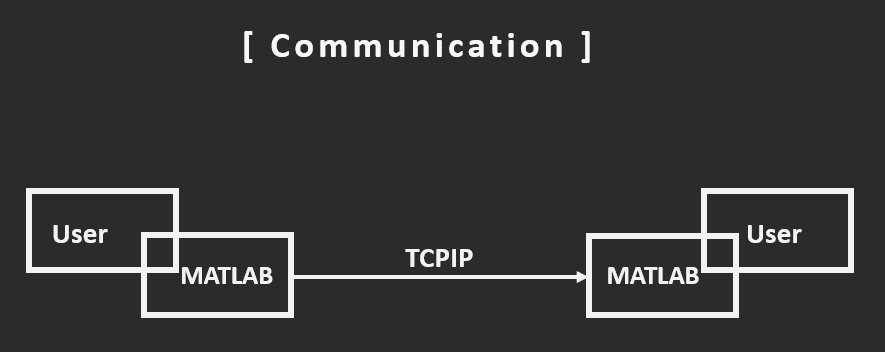
**Figure 4.2: Block Diagram**



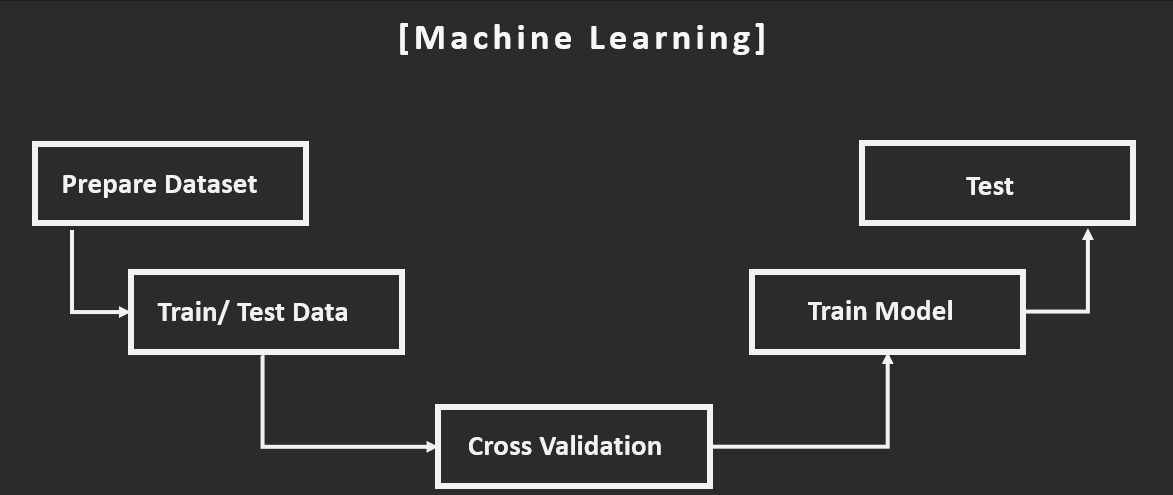
**Figure 4.3: TDM Block Diagram**



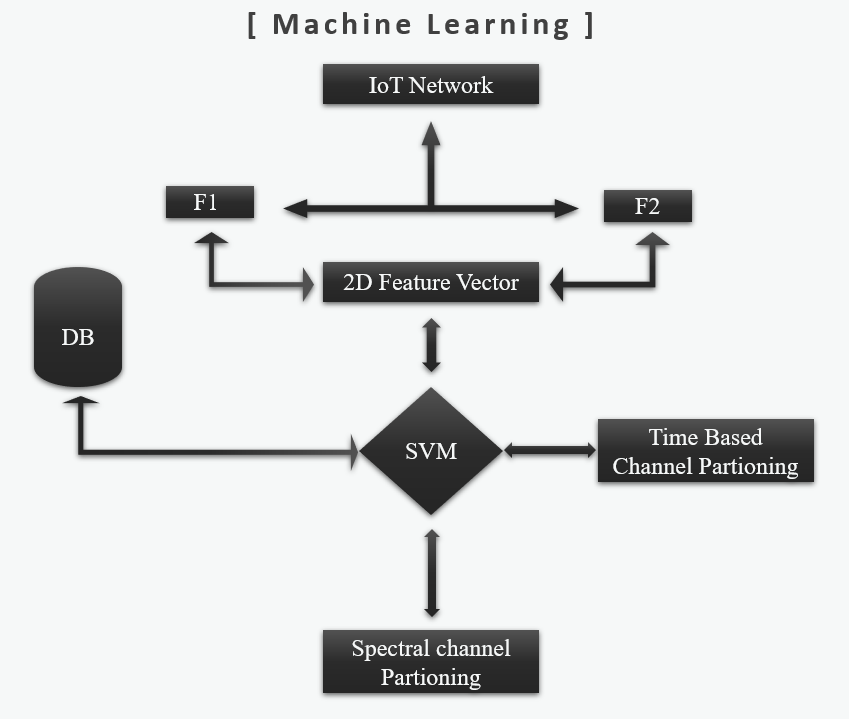
**Figure 4.4: FDM Block Diagram**

****

**Figure 4.5: Communication Block Diagram**



**Figure 4.6: Machine Learning Block Diagram**



**Figure 4.7: Proposed Classification System**

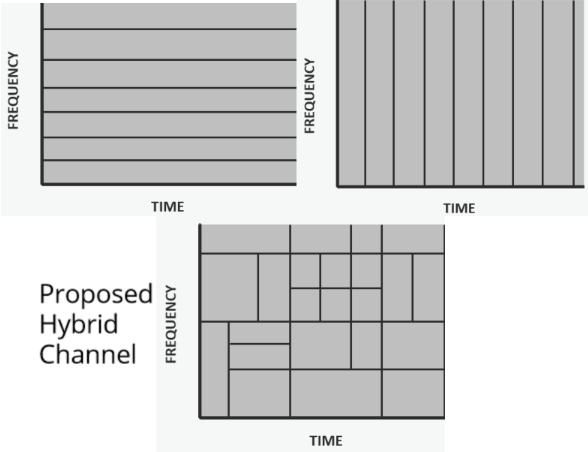
**Chapter # 5**

**System Implementation**

* 1. **System Model**

**5.1.1 Network Model:**

We propose a sensor network which is based on IoT. The network consists of **N** number of sensor nodes. Nodes can be any devices it could be any sensor. We have **K** number of multiresolution channels as shown in figure 5.1. To keep things simple and avoid any useless complexity we modeled the multi-resolution channel by introducing ‘Additive White Gaussian Noise’ (AWGN).



**Figure 5.1: Proposed Multi-resolution Channels**

**5.1.2 Hybrid Channel Partitioning Protocol:**

The key attribute of the suggested protocol is that it is able to enhance data transmission by acquiring channel qualities. All IoT nodes within the network is furnished with Gaussian Radial Basis Function. The RBF function works as the kernel for the Support Vector Machines. It helps to classify the data. As we have a kernel function for the SVM, so it is now called a non-linear SVM. If we have no kernel function then SVM is called linear SVM. The input function of this non-linear support vector machine is a two-dimensional feature vector f = [f1.f2]. The algorithm gives the decision whether to use channel time characteristics or channel frequency features.

1. **Number of Nodes (f1):**

This feature represents the total nodes inside a wireless IoT sensor network. Basically it is the sum of all devices inside the network which is computed using the equation 1 and 2 [1]:

***N = {n1, n2, n3, …., nk}* (1)**

**(2)**

1. **Number of Messages (f2):**

This attribute represents the number of messages transmitted through the respective channel. It is obtained by calculating equation 3. Its unit is samples per second.

**(3)**

The purpose of extracting these features from all IoT devices is to decide the use of frequency frame or time frame for data transfer. In a tightly packed IoT network where all nodes have to transmit large data, spectral frame is used. When the amount of data for transmission is less and the number of nodes is also less, then it is efficient to use time frame.

**5.2 Tools and Technology**

**MATLAB:**

MATLAB is an environment for multi-paradigm numerical computing. It has a proprietary language for programming. It is developed by MathWorks [17]. It has many useful features including plotting of functions, matrix manipulation, plotting of data, implementation of algorithm such as machine learning algorithms and creation of user interface etc.

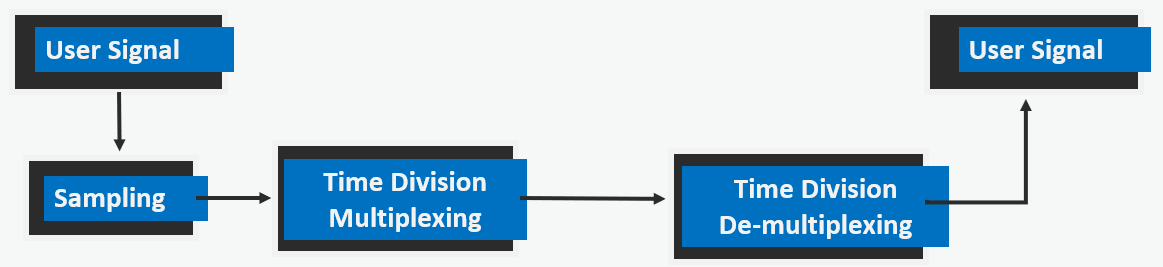
We applied Nonlinear support vector machine on a data set on MATLAB to obtain a trained machine learning model. All the coding blocks including time division multiplexing and frequency division multiplexing are completed on MATLAB.

* 1. **Processing Logic:**

The coding of each block (FDM, TDM, and Machine Learning) is carried out separately. Each block is tested to make sure that they work as intended. All the coding blocks are integrated into a single MATLAB session. In order to fine tune the integrated coding block, further errors and bugs are also removed.

* + 1. **TDM:**

To depict the nodes, we created a variable called number of users denoted by **“nou”**. Time Division Multiplexing coding blocked is generic. The number of signals generated at the transmitter side is equal to the total number of nodes. First node signal is sinusoidal while the second node signal is triangular.So, every node represented by odd number will have a sinusoidal signal while every node represented by even number will have a triangular signal.

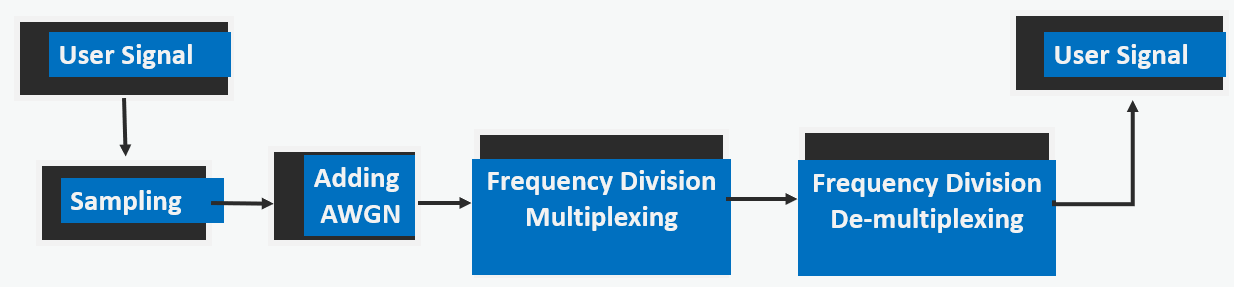


**Figure 5.2: TDM Logic Block**

User signal is sampled according to the sampled rate Nyquist Frequency [18]. After sampling signals are multiplexed. The time frame is first used by a single node for limited period. Constant time period is same for every node. Signals are converted in form of bits for data transmission. In fact, bits are being transmitted between transmitter and receiver. At the receiver side the multiplexed signals are demultiplexed. Hence the original signal is received successfully.

* + 1. **FDM:**

In Frequency Division Multiplexing block, number of users represents nodes. Different channels are created at different frequencies. Every signal has a carrier frequency which is ten time its own frequency. Frequency deviation is also introduced. Once the modulated signal is created, it passes through the channel.



**Figure 5.3: FDM Logic Block**

To depict real time noise interference in the signal additive white gaussian noise is also introduced. Each node signal is sampled. After sampling the signals are multiplexed at the transmitter end, and then demultiplexed at the receiving end. In a tightly packed IoT network where all devices have to transmit sizeable data, spectral frame is more efficient than time frame.

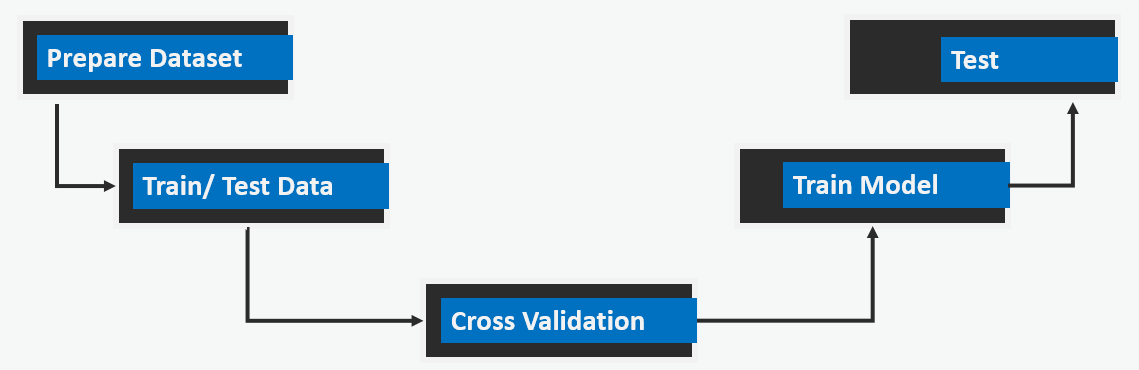
* + 1. **Machine Learning:**

SVM is used the train the model. Built in MATLAB command “Fitcsvm” is used [19]. Kernel function is also added. Data set is divided into train data and test data then the training data is used by the Support Vector Machine. During the training process the data set is cross validated to achieve better accuracy.



**Figure 5.4: Cross Validation**

Once the model is trained then it is tested on a sperate data set called test set. Trained model is called in the main coding block to give real time decision. Then time frame or spectral frame is used according to the predicted decision by the trained model.



**Figure 5.5: Machine Learning Logic Block**

**Chapter # 6**

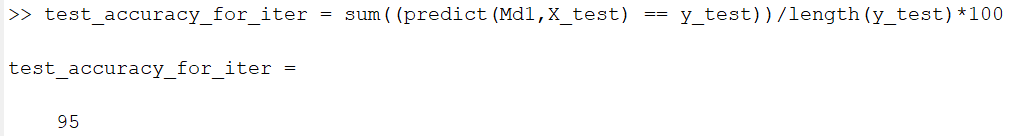
**System Testing and Evaluation**

**6.1**  **Result and Evaluation:**

After spending endless hours on MATLAB, we achieved splendid results with a tremendous accuracy of our trained model and successful data transmission. Designed protocol is prototyped on a laptop with Core i7 processor and 12GB DDR4 RAM. Simulation are obtained using MATLAB 2018b. SVM based decision support system using Gaussian Basis Radial Function has been deployed across all the nodes.

During the classification process cross validation process is used to compute decision boundary. This procedure is carried on with different iterations to attain accuracy of more than 90%. represents the classification accuracy and it is calculate using equation 4. Figure 6.1 shows the classification. Figure 6.8 to 6.10 shows logic from final integrated block.

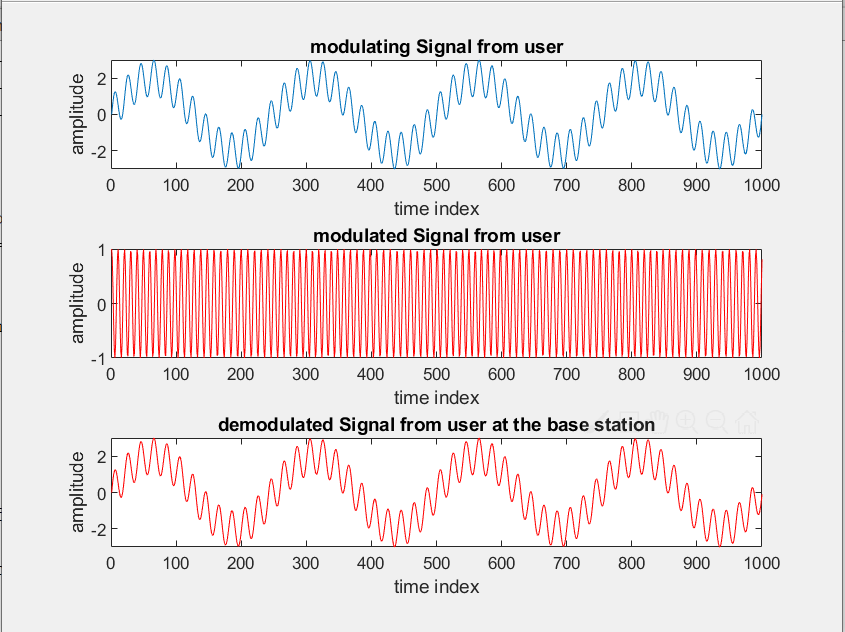
**Classification Accuracy:**



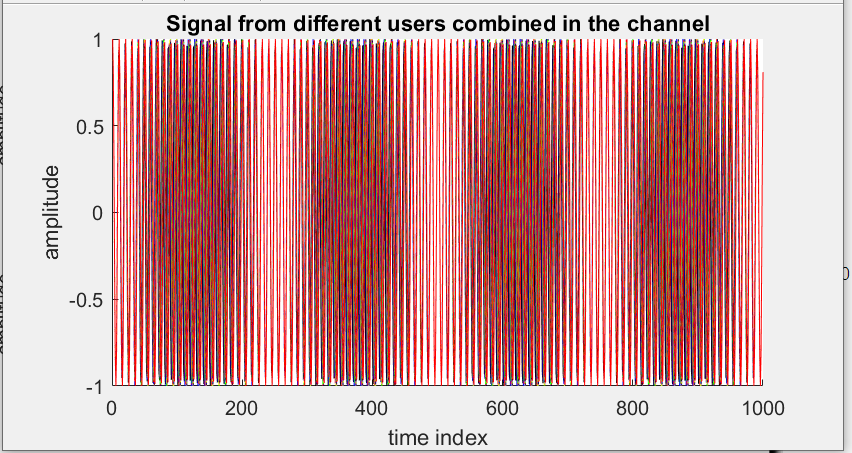
**Figure 6.1: Classification Accuracy**

**(4)**

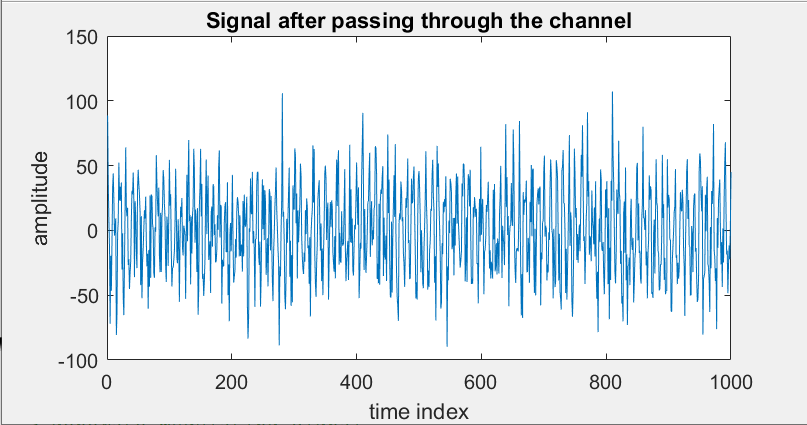
**FDM:**



**Figure 6.2: Signals from Sender and Receiver**

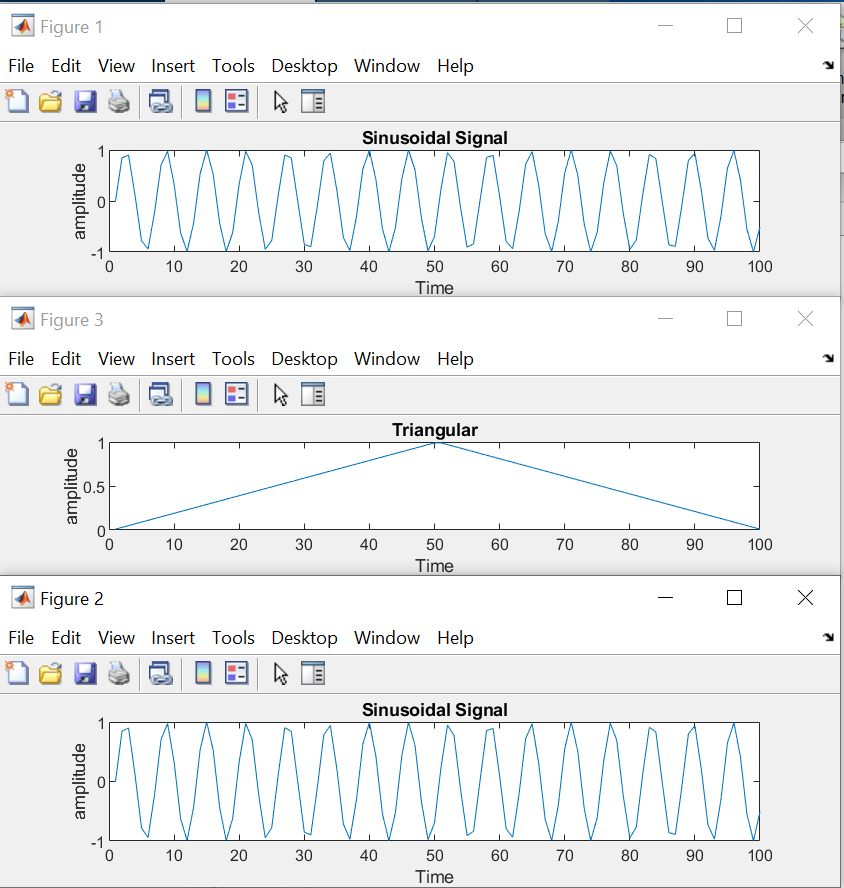


**Figure 6.3: Multiplexed Signal**

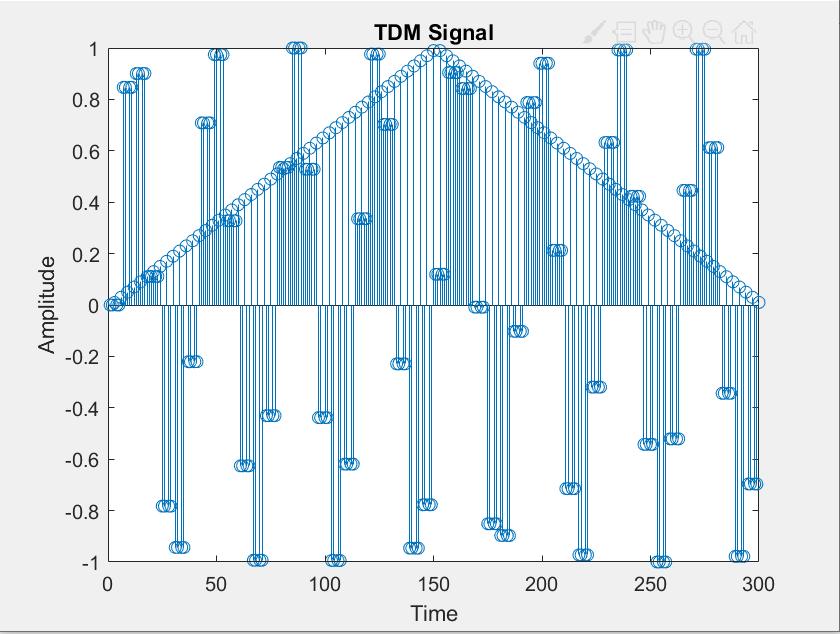


**Figure 6.4: Signal with AWGN**

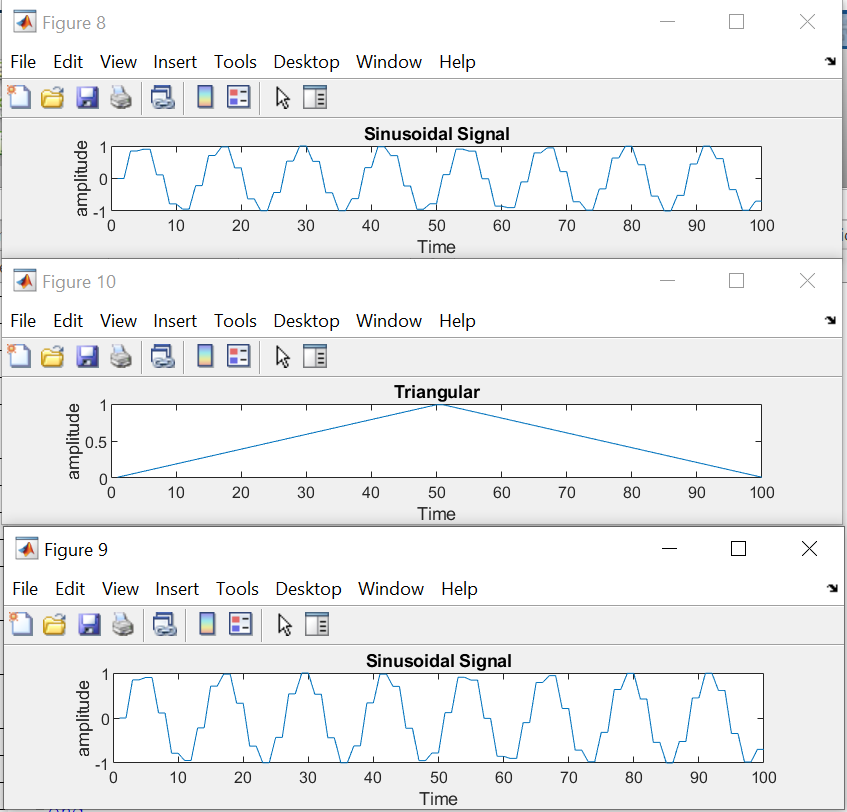
**TDM:**



**Figure 6.5: Signals from Nodes**



**Figure 6.6: TDM Signal**

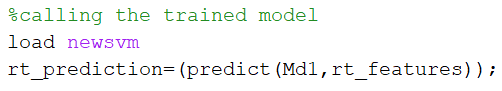


**Figure 6.7: Signal at Receiver end**

**Integration:**



**Figure 6.8: Real Time Feature Extraction**



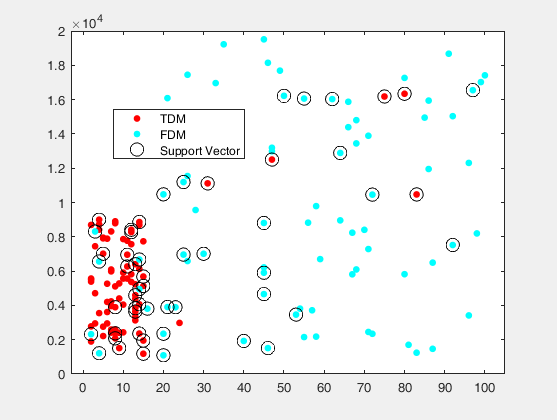
**Figure 6.9: Real Time Prediction Logic**



**Figure 6.10: Real Time Decision Logic**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Correctly Classified Samples | **Accuracy** |  | ***T****p* | ***T****n* | ***F****p* | ***F****n* |
| 38/40 | 95.0 % |  | 19 | 19 | 1 | 1 |
|  |  |  |  |  |  |  |

**Table 3.2: Proposed System Performance**



**Figure 6.11: Dataset Plot**

**Chapter # 7**

**Conclusion**

#### This report presents a fully automated apprised network for channel allocation protocol for high speed transfer in an IoT sensor nets. For defining the mode of transmission IoT devices dynamically grasp from their load features. The purposed protocol gives the surety for optimal selection of channel for data transmission.

#### Purposed protocol is novel as it learns about the network traffic density. It also tracks the number of nodes and messages to make high speed and efficient data transmission possible. Spectral bandwidth is used efficiently. The purposed protocol lies in the category of hybrid channel partitioning protocol.

MATLAB provides a robust platform for applying machine learning. As we can see all kinds of variables in the MATLAB workspace, so it makes easy to debug the program. Also, SVM built in library is magnificent. MATLAB provides different optimization function to optimize the performance of support vector machines. Cross validation and kernel functions can be applied easily. Cross validation can make the model training process better.

Kernel function plays a key role in determining the optimal decision boundary. Box constraint and alpha values could be set manually but MATLAB provides a key feature through which we can select the optimal values. Same goes in case of kernel scale is can be set manually and automatically as well.

SVM takes array as its input function but the data is normally available in the form of table to it is necessary to convert the table into array. It is useless to train the model every time while executing a real time program. In MATLAB once a model has been trained. It can be easily called into another MATLAB session. One of the key and difficult task is communication between two devices. MATLAB provides a wireless mode of communication between two different MATLAB sessions through TCP IP. It is important to tune buffer size and select right port number to avoid noise and loss of bits during wireless communication.

**Future Work:**

Proposed protocol can be applied over the wide range of applications in wide area cellular IoT network. One of its key applications is ‘Telemedicine System’ where high speed and reliable transmission is required to send different diagnostic reports between remote unit and hospital. Proposed protocol has application in military and defense system where reliable transmission is crucial between distinct remote troops. It also has applications in the agriculture field especially in forest fire detection system where sensor nets have to communicate over a large area. Proposed protocol can be extended by introducing a second stage decision system where channel switching can be established if neighboring channel have less congestion.

**References**

[1] Hassan, T., Aslam, S. and Jang, J.W., 2018. Fully automated multi-resolution channels and multithreaded spectrum allocation protocol for IoT based sensor nets. *IEEE Access*, *6*, pp.22545-22556.

[2] D. Oh, D. Kim, and W. W. Ro, ``A malicious pattern detection engine for

embedded security systems in the Internet of Things,'' *Sensors*, vol. 14,

no. 12, pp. 24188\_24211, 2014.

[3] F. Li, Y. Han, and C. Jin, ``Practical access control for sensor networks in the context of the Internet of Things,'' *Computer. Communication.*, vols. 89\_90,

pp. 154\_164, Sep. 2016.

[4] S. Aslam, N. U. Hasan, J. W. Jang, and K.-G. Lee, ``Optimized energy harvesting, cluster-head selection and channel allocation for IoTs in smart cities,'' *Sensors*, vol. 16, no. 12, p. E2046, 2016.

[5] P.-J. Wu and C.-N. Lee, ``Connection-oriented multi-channel MAC protocol

for ad-hoc networks,'' *Computer. Communication.*, vol. 32, pp. 169\_178, Jan. 2009.

[6] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, ``A survey on wireless multimedia sensor networks,'' *Computer. Network.*, vol. 51, pp. 921\_960,

Mar. 2007.

[7] W. Rehan, S. Fischer, and M. Rehan, ``Machine-learning based channel quality and stability estimation for stream-based multichannel wireless sensor networks,'' *Sensors*, vol. 16, no. 9, p. 1476, 2016.

[8] M. A. Hamid, M. M. Alam, and C. S. Hong, ``Design of a QoS-aware routing mechanism for wireless multimedia sensor networks,'' in *Proc.* *IEEE Global Telecommunication. Conf. (GLOBECOM)*, New Orleans, LA, USA, Dec. 2008, pp. 1\_6, doi: 10.1109/.GLOCOM.2008.ECP.159.

[9] A. Pal and A. Nasipuri, ``DRCS: A distributed routing and channel selection

scheme for multi-channel wireless sensor networks,'' in *Proc. IEEE Int. Conf. Pervas. Computer. Communication. Workshops (PERCOM)*, San Diego,CA, USA, Mar. 2013, pp. 602\_608.

[10] P.-J. Wu and C.-N. Lee, ``Connection-oriented multi-channel MAC protocol

for ad-hoc networks,'' *Computer. Communication.*, vol. 32, pp. 169\_178, Jan. 2009.

[11] A. Shahid, S. Aslam, and K.-G. Lee, ``Distributed joint resource and power allocation in self-organized femtocell networks: A potential game approach,'' *J. Netw. Computer. Appl.*, vol. 46, pp. 280\_292, Nov. 2014.

[12] Q. Yu, J. Chen, Y. Sun, Y. Fan, and X. Shen, ``Regret matching based channel assignment for wireless sensor networks,'' in *Proc. IEEE* *Int. Conf. Communication. (ICC)*, Cape Town, South Africa, May 2010, pp. 23\_27.

[13] <https://www.polytechnichub.com/advantages-disadvantages-applications-time-division-multiplexing-tdm/>

[14] <https://electricalvoice.com/frequency-division-multiplexing-working/>

[15] <https://blogs.mathworks.com/loren/2011/05/27/transferring-data-between-two-computers-using-matlab/>

[16] Vince Glacier, "Wsn-multiplexing-data-set", IEEE Dataport, 2020. [Online]. Available: http://dx.doi.org/10.21227/1znc-v703. Accessed: Apr. 16, 2020.

[17] <https://en.wikipedia.org/wiki/MATLAB>

[18] <https://www.sweetwater.com/insync/nyquist-frequency/>

[19] <https://www.mathworks.com/help/stats/fitcsvm.html>

[20] <https://www.mathworks.com/discovery/support-vector-machine.html>

[21] <https://machinelearningmastery.com/k-fold-cross-validation/>

[22] https://securitytoday.com/articles/2020/01/13/the-iot-rundown-for-2020.aspx

**Appendices**

**Appendix A.**

SVM (support vector machine)

IoT (internet of things)

WSN (wireless sensor network)

FDM (frequency division multiplexing)

FDMA (frequency division multiple access)

TDM (time division multiplexing)

TDMA (time division multiple access)

CV (cross validation)

**Appendix B.**

For machine learning training, conversion of table to matrix for data set, wireless communication and frequency division multiplexing built in MATLAB libraries are used respectively.

[fitcsvm] [table2array] [tcpipServer = tcpip('0.0.0.0',55000,'NetworkRole','Server');]

[fmmod]