Engineering Chimera and Novel Technique Based on Machine Learning

A THESIS

Submitted in partial fulfillment of the requirements for the award of the degree

of

Master of Science

by

NIRAJ KUSHWAHA

under the guidance of

Prof. SARIKA JALAN

DISCIPLINE OF PHYSICS

INDIAN INSTITUTE OF TECHNOLOGY INDORE

June 2020

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **ENGINEEERING CHIMERA AND NOVEL TECHNIQUE BASED ON MACHINE LEARNING** in the partial fulfillment of the requirements for the award of the degree of **MASTER OF SCIENCE** and submitted in the **DISCIPLINE OF PHYSICS, Indian Institute of Technology Indore**, is an authentic record of my own work carried out during the time period from JULY 2019 to JUNE 2020 under the supervision of PROF. SARIKA JALAN, Professor, Discipline of Physics, Indian Institute of Technology Indore.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

19/06/20

Signature of the student with date
(NIRAJ KUSHWAHA)

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

06.07.2020
Signature of the Supervisor of
M.Sc. thesis (with date)
(Prof. Sarika Jalan)

NIRAJ KUSHWAHA has successfully given his/her M.Sc. Oral Examination held on **23rd June 2020.**

Signature(s) of Supervisor(s) of MSc thesis
Date: 06/07/2020

Convener, DPGC
Date: 23/06/2020

Signature of PSPC Member #1
Date: 23/06/2020

Signature of PSPC Member #1
Date: 23/06/2020

ACKNOWLEDGEMENT

First and foremost I would like to express my sincere gratitude to my guide Prof. Sarika Jalan for allowing me to be a part of complex systems lab for the past 1 year and for continuously supporting me throughout my project. I have benefited immensely from her deep insight of the subject and her enthusiasm for research. Apart from the technicalities of this subject, I also learned how to perform quality research, from her. Her pursuit of relevant, original and high quality work has always been a source of inspiration for me and has helped me in pushing myself out of my comfort zone to pursue new and exciting ideas during this project. I am extremely obliged to her for everything I learnt from her during the last 1 year.

I am grateful to my PSPC members, Dr. Sudeshna Chattopadhyay and Dr. Trapti Jain for scrutinizing my work and for giving me their valuable feedback after my presentations on how I can improve my work as well as my presentation.

I would also like to thank all the faculty members of discipline of physics, IIT Indore who were always available for discussions and were ready to provide any support I needed while pursing my MSc degree. I am equally grateful to each and every faculty member but I would like to give special mention to Dr. Ankhi Roy and Dr. Sudip Chakraborty who taught me the computational techniques course.

I wish to extend my heartfelt gratitude to all my seniors at the complex systems lab. You have supported me in all my ups and downs during this project. I have learnt a lot from each and everyone from CSL. CSL felt like a family to me because of my incredible seniors and in this family I was the youngest member for the past 1 year. Thank you so much once again for allowing me to be a part of this family.

Ofcourse, how can I forget my MSc- Physics batch!! I cherish each and every memory that I have had with my batchmates in the last 2 years.

I am fortunate to be a part of such a great batch. Special gratitude to the "B-Team". The barrels of laugh that we have shared and all the fun memories we have created in the last 2 years will always stay very dear to my heart.

And now to the most special people of all, I would like to thank my mother and father for bringing me into this world and for providing me a comfortable and resourceful life which has enabled me to pursue things I am most passionate about. Without you, non of this or any other great thing in my life, would have been possible. I would also like to thank my brother for providing me with every kind of support I needed in my life.

Niraj Kushwaha

Dedicated to my parents and my brother

ABSTRACT

Recently, machine learning techniques have been put into practice in precisely characterizing various dynamical properties or phenomena. Here we make use of supervised machine learning algorithms for the model-free prediction of factors determining or controlling the intensity of symmetry breaking phenomena emergent in different network architectures. In an attempt to achieve this, chimera states (solitary states) are engineered by establishing delays in the neighboring links of a node (the interlayer links) in a 2-D lattice (multiplex network) of oscillators. Different machine learning classifiers, K-Nearest Neighbours (Knn), Support Vector Machine (SVM) and Multi-Layer Perceptron Neural Network (MLP-NN) are then employed, feeding on the data obtained from mentioned models, for the prediction of intensity of rippling chimera states and critical delay to characterize solitary states. It is revealed from our analysis that Multi-Layer Perceptron Neural Network (MLP-NN) classifier is best suited for the characterization of the engineered chimera and solitary states. We hope that our successful attempt in characterizing a class of partially synchronized states using machine learning techniques would be useful in broadening the scope of model-free machine learning techniques in characterizing other phase states as well.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Studying complex real world systems has been a challenge for us for a long time. Network science gained popularity among scientists over the years because it has proven to be an effective way of studying complex systems. The tools of network science allow us to model a complex system using simple components such as nodes and links and the interactions between these nodes and links which lead to collective behaviour in such systems. The real world interactions between the nodes can be mimicked by putting an oscillator on each of the nodes such that they interact with eachother depending on the network structure. Network science has wide applications and has helped us in understanding vast varieties of complex systems found in nature belonging to different disciplines of sciences ranging from biology, physics, sociology, economics etc[1, 2].

1.1 Networks

In read world nothing acts like an isolated system. No real world system is isolated in nature. Real world systems are interconnected and changes in one system affects the time evolution and dynamics of another system. Therefore, it is necessary to study different systems together as connected units. All the connected systems all together through their interactions show collective behaviours. This collective behavior might be very different from what we might expect from time evolution of an isolated system.

To facilitate this connected dynamical evolution of systems, networks came in forefront. While constructing a network we represent each isolated system as a node(or vertex) and the interactions between these systems is represented by links(or edges). So a single network can give us the full picture of how a collection of different systems are connected with eachother and how they affect eachother while evolving in time. Mathematically, we represent and study networks using graphs.[14] One of the most popular way to represent a graph or a network is through it's adjacency matrix(A_{ij}). The elements of this adjacency matrix is defined as a_{ij}=1 if i^{th} and j^{th} nodes are connected and a_{ij}=0 otherwise.

$$A_{ij} = \begin{cases} 1 & \text{if i-j,} \\ 0 & \text{otherwise} \end{cases}$$

1.2 Kuramoto Oscillators

To mimick and study the overall dynamics and time evolution of a complex system that we have represented by a network, we put coupled oscillators on each node of the network. The oscillators at each node are coupled with each other depending on whether two nodes have a link between them or not. If two nodes have link between them then oscillators on each node will be coupled with each other.

Kuromoto model is one of the many oscillator models that are used for this purpose[11]. Even though kuromoto model is a very simple model, it has the capability to explain very complex dynamical phenomenons. The emergence of chimera was also first reported using kuromoto model[15][16].

In kuromoto model, the collective state of a complex system is defined in terms of phase(θ) of all the nodes present in a network. So Θ=[θ_1,θ_2,θ_3,....] represents the dynamical state of a complex system at a particular time. The phases of the nodes evolve in time according to the equation,

$$\frac{d\theta_i(t)}{dt} = \omega_i + \kappa \sum_{j=1}^{N} A_{ij} \sin\left(\theta_j(t) - \theta_i(t)\right) \tag{1.1}$$

Here, N is the total number of nodes present in the network. θ_i is the phase of the i^{th} node. ω_i is the natural frequency of the i^{th} oscillators in our network. κ is the coupling constant. A_{ij} is the adjacency matrix of our network.

1.3 Synchronization

Synchronization is one of the collective behaviours that is exhibited by complex networks. A complex network is said to be synchronized when all of the oscillators in that network lock into a common frequency. A network of oscillators having similar natural frequency and a network of oscillators have different natural frequencies, both can show collective behavior of synchronization[11].

We observe synchronization in wide varieties of real world complex systems and networks. In physics we see synchronization in a arrays of lasers[27, 28], Josephson junctions[29, 30], microwave oscillators[31] etc.

Biological examples of synchronization behaviour includes, synchronous flashing of fireflies[32, 33], pacemaker cells in heart[34, 35] etc.

A system of oscillators which are synchronized with each other is also called a coherent system.

1.4 Chimera

Kuramoto and Battogtokh, in 2002, discovered a fascinating new phenomena that a system of identical Kuramoto oscillators breakup into two different groups based on their synchronous properties[3]. Later, Strogatz and Abrams named this new emergent collective behaviour of the system of oscillators as Chimera state[4].

The study of emergence of a collective behaviour in a network is very important and it holds a key to understand the complex system which the network represents. The emergence of chimeric behaviour deserves a special attention since chimera states has been found in many real world complex systems such as brain[5, 6], meta-materials[7] etc.

Scientists have found many types of special chimera patterns and named them as Virtual chimera[8], Travelling chimera[9], Breathing chimera[10] and many more. In our work, we too found a unique chimera pattern and we named it Rippling Chimera.

Chimera states have been found in various dynamical models such as Stuart-Landua[37, 38], Chaotic oscillators[39], Van-der Pol oscillators[?] etc.

1.5 Machine learning

Recently, a plethora of articles have been published exploiting machine learning techniques as a tool for the prediction of possible outcomes, system properties or emergent phenomena covering a broad area of interdisciplinary research, which ranges from non-linear dynamics, quantum physics, astrophysics to bio-medics [20, 19]. The fields of complex systems and non-linear dynamics have also witnessed a recent spurt in the use of machine learning techniques in the characterization or identification of a variety of system properties or behavior or phenomena. For instance, the machine learning algorithms have successfully been implemented in community detection in networks [21], finding fixed points attractors [22], spatiotemporal chaotic systems [23], detecting phase transition [24], prediction of chaotic systems [25] and identification of chimera states [26].

We have used three different supervised machine learning algorithms in the next sections. Those machine algorithms are K-Nearest Neighbours(Knn) classifier, Support Vector Machine(SVM) classifier and Multi Layer Perceptron Neural Network(MLP-NN) classifier.

Knn classifier is a non-parametric classification algorithm which has proven to be effective in numerous cases[18]. If we represent our data in a vector space then each point in this vector space can be classified based on the classes of k nearest neighbours of this data point. The k nearest neighbours are chosen based on a distance parameter. Most commonly euclidean distance is used to chose the k nearest neighbours. So Knn divides our data's vector space in different regions corresponding to different classes. The parameter k plays a very important role in deciding how well Knn will perform while dividing the vector space in different regions and classifying the points in that vector space.

SVM classifier is a supervised machine learning model. It works by estimating the most appropriate hyperplane which can separate our training data into two different distinct classes. The hyperplane estimation is done by maximizing the distance between nearest training data point and the proposed hyperplane. This distance is also called margin. Simple SVM can only produce linear hyperplanes. In order to estimate non linear hyperplanes one can use kernels. Kernel works by transforming our training data from a lower dimensional space to a higher dimensional space and then estimating a linear hyperplane in that higher dimensional space. On transforming the higher dimensional hyperplane back to the lower dimension, we get a non-linear hyperplane which can classify each point of our data's vector space into different classes[17].

MLP-NN classifier works by creating an artificial neural network consisting of many different layers of nodes. There are three types of layers in a neural network. Those are input layer, hidden layer and the output layer. We can have any number of hidden layers and each hidden layer can have any number of nodes. The neural network takes in the input data and tries to estimate the weights of each link between the nodes of this network. A neural network can be called a trained model if the algorithm successfully estimates the weights of the links such that the model can categorize our data into their correct classes.

1.6 Organization of Thesis

This thesis and my project can be broadly divided into three parts. These three parts are divided in such a way that it can show all the work I've done in the past one year in a systematic and chronological order.

In the first part of my project I extended the earlier work done by Dr.

Saptarishi Ghosh et al.[12]. Dr. Ghosh et al showed that a chimera state can be engineered in a complex network by applying appropriate amount of delay in few nodes of the network. They showed this phenomena in a 1D ring network. I extended this work by applying the same technique on a 2D lattice network with periodic boundary condition. In doing so I discovered a new kind of chimera which we named "Rippling Chimera". All the results that I got while engineering and then studying rippling chimera is shown in chapter 2 of this thesis.

In the second part of my project I developed a completely new and unique technique to predict dynamical properties of a network using machine learning. Chapter 3 contains all the information about this technique and all the results that I got after applying this technique on a 2D lattice network is shown in this chapter. This technique is a completely novel idea and the main highlight of this thesis.

In the third part I applied this new machine learning technique on a completely different type of network(multiplex network) to see if this technique can be used in different types of network architecture or not. I performed this technique on a multiplex network to predict it's critical delay value. All the analysis and results done in this is shown in chapter 4.

Chapter 5 is for conclusions and future scope of this work.

Chapter 2

Emergence of Chimera in 2D Lattice Network

2.1 Introduction

Existence of Chimera state has been known and studied for a long time ever since it was discovered in the year 2002. [3] The behaviour and origin of Chimera states is well studied over the years. Now, the interesting question that arises is that can we engineer or induce a chimera state in a network according to our requirement. It has been shown that a chimera state can be induced/engineered by adding delay to the nodes of a 1-Dimensional periodic ring network[12]. We are using the ideas used in the previous paper and applying it to a more complicated network structure.

Engineering chimera can have great applications in future. By engineering chimera states in simple network structures and studying them we can learn a lot about how we can control chimera states and then we can extend this knowledge to more complicated and real world networks. This can have many potential real world applications in future in fields including but not limited to neuro-science and solid state physics.

2.2 Theoretical Framework

We studied the emergence of chimera using delay in a 2-Dimensional lattice network with periodic boundary condition. The coupling between two

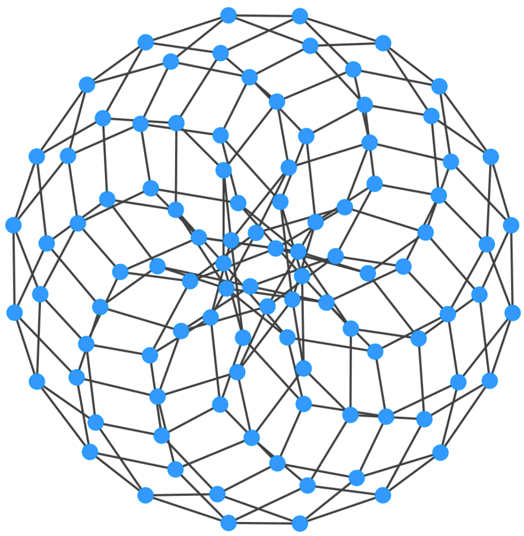

Figure 2.1: Schematic diagram of a 10x10 2-dimensional lattice network with periodic boundary condition with $N = 100$. All the nodes here are identical having exactly the same coupling architecture

connected nodes was modelled using Kuramoto oscillators. Equation 2.1 defines a Kuramoto model mathematically.

- Network description:

  The network is of locally coupled identical Kuramoto oscillators arranged in a 2-Dimensional lattice with periodic boundary conditions such that every node of the network had a degree 4. (See Fig(2.1))

  A network of size $N = 100$ was used in this analysis. (See Fig(2.1c))

- Delay:

  When a delay is introduced in a node of a network(say node $\boldsymbol{x}$) then that means that the information that $\boldsymbol{x}$ receives from $\boldsymbol{x}$'s neighbours will be delayed i.e $\boldsymbol{x}$ will receive the information from the nodes connected to $\boldsymbol{x}$ after a finite amount of time instead of instantaneously.

  The underlying equations which will determine the dynamics of this system are,

$$\frac{d\theta_i(t)}{dt} = \omega + \kappa \sum_{j=1}^{N} A_{ij} \sin\left(\theta_j(t) - \theta_i(t)\right) \tag{2.1}$$

$$\frac{d\theta_i(t)}{dt} = \omega + \kappa \sum_{j=1}^{N} A_{ij} \sin\left(\theta_j(t-\tau) - \theta_i(t)\right) \tag{2.2}$$

Eq.(2.2) describes the dynamics of those nodes which have a delay τ in them and Eq.(2.1) describers the dynamics of those nodes which don't have any delay.

Here, N is the total number of nodes present in the network. θ_i is the phase of the i^{th} node. ω is the natural frequency of all the identical oscillators in our network. κ is the coupling constant. τ is the amount of delay applied to a node. A_{ij} is the adjacency matrix of our network.

- Order Parameter:

Order parameter is a measure of how coherent or incoherent a system of oscillators are. Order parameter is used to identify if our system of oscillators is synchronized or not. The order parameter is defined as, [13]

$$re^{i\psi} = \frac{1}{N} \sum_{j=1}^{N} e^{i\theta_j} \tag{2.3}$$

A more simplified way to write this is,

$$r = \frac{1}{N} \sqrt{\left(\sum_{j=1}^{N} \cos\theta_j\right)^2 + \left(\sum_{j=1}^{N} \sin\theta_j\right)^2} \tag{2.4}$$

Here, θ_j is the phase of the j^{th} node and N is the total number of nodes present in the network.

If all the oscillators are completely incoherent to each other then in that case the value of order parameter(r) is zero and if all the oscillators are completely synchronized with each other then the value of r is 1.

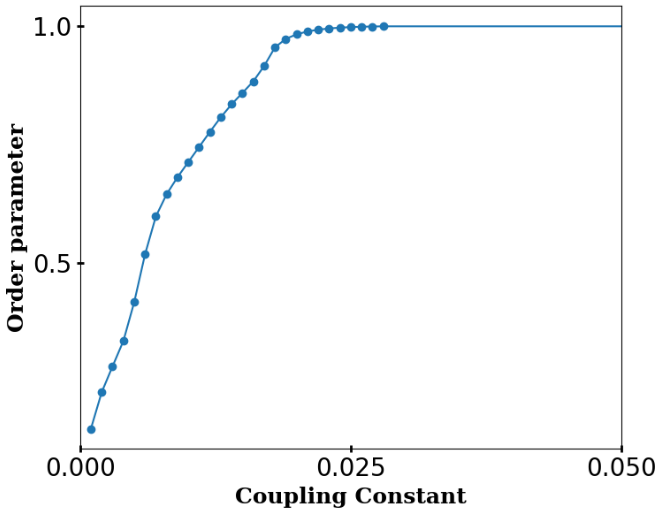

Figure 2.2: Coupling constant(κ) vs Order parameter(R) for 2-Dimensional lattice network with total nodes, $N = 100$ and $\alpha = 0$. This shows that the critical coupling constant is around $\kappa_c = 0.02$.

2.3 Results

2.3.1 Finding the critical coupling constant

While examining the overall dynamics of the network without any delay it was found that the coupling constant determines whether the whole network will get synchronized or not. On plotting order parameter with respect to coupling constant, the critical coupling constant was found. Critical coupling constant is the value of coupling constant above which the system can synchronize if it is allowed to evolve for sufficient amount of time. In all the simulations done in future, the coupling constant was always taken such that it is greater than the critical coupling constant.

In Fig(2.2) we can see that the order parameter with respect to the coupling constant shows a 2^{nd} order transition.

2.3.2 Engineering Chimera

We randomly assigned the initial phase to all the oscillators between the interval [0,2π]. We then introduced a delay in the 45^{th} node of our network and observed the change in the dynamics of the system.

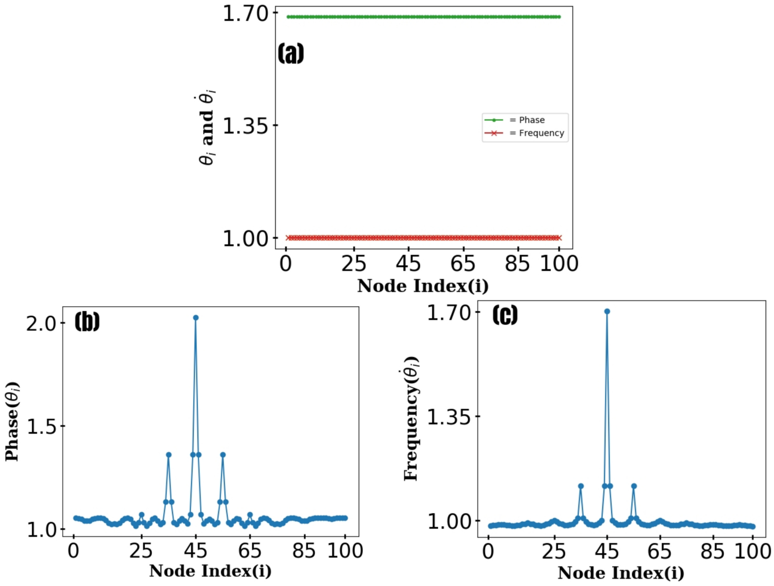

Figure 2.3: Chimera obtained by introducing delay in a 2D lattice network. (a) Shows a snapshot of Node index vs Phase(θ) and final frequency($\dot{\theta}$) for system without delay($N = 100$,$\tau = 0$,$\kappa = 1$).This plot shows that in the absence of any delay all the oscillators synchronize. (b) shows a snapshot of Node index vs Phase(θ) for system with delay in the links of the 45^{th} node ($N = 100$,$\tau = 10$,$\kappa = 1$). This plot shows that there exists two kinds of nodes in the system. Some are coherent and some are incoherent. This indicates that we can engineer a chimera state by introducing a delay in the links of one of the nodes in our network. (c) shows a snapshot of Node index vs frequency($\dot{\theta}$) for system with delay in the links of the 45^{th} node ($N = 100$,$\tau = 10$,$\kappa = 1$).

On applying delay on one of the nodes(45^{th} in our case) emergence of chimera state was observed. The comparison between the final state of the network with and without delay is shown in the Fig(2.3). The chimera pattern obtained in this system resembles a ripple of wave and so we are calling it "Rippling Chimera"(Fig(2.3b,2.3c)).

We can see in the Fig(2.4) that on introducing a delay in the 45^{th} node we get a chimera state. The nodes near the delayed node are in a incoherent state and the nodes which are far from the delayed node are coherent to each other.

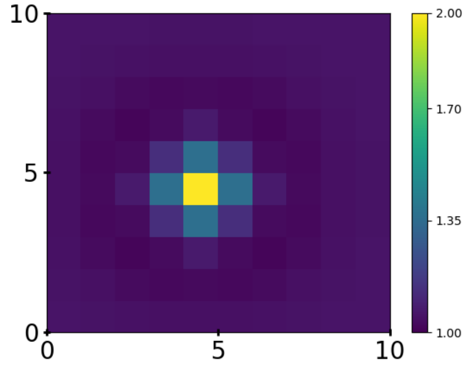

Figure 2.4: 2-Dimensional colour-grid representation of phases of a 2-Dimensional lattice network with parameters, $N = 100$, $\kappa = 1$, $\tau = 10$, $\alpha = 0$. The delay was introduced in the links of the 45^{th} node. Each block in this figure represents a node. The nodes nearer to the 45^{th} node show incoherence in phases. The nodes far from the 45^{th} node are coherent with each other. This whole system thus represents a chimera state.

2.3.3 Phase vs. Frequency diagram

Till now, we have managed to get a macroscopic overview of our system and how the overall system changes when we introduce a delay in a 2-dimensional lattice/network of oscillators. Now, we will look at the dynamics of the nodes more closely in order to get some microscopic information about what's happening with a single node in the system. One of the questions we wish to answer in this section is that, how are the nodes behaving individually and how the individual behaviour of different nodes together result in a collective phenomena of a synchronized or a chimeric state.

We plotted the phase portrait of some of the nodes to deduce the overall behaviour of that node and to find out if that node has a stable trajectory or not. If the node finally settles into a constant frequency and all the other nodes also settle to the same constant frequency, we can say that those nodes are coherent to each other. If a node's trajectory suggests that it cannot settle to a constant frequency and it follows a chaotic path then we can conclude that the node is incoherent with respect to the other nodes. If a system has a combination of such incoherent and coherent nodes then we can conclude that the system is showing chimeric behaviour and if the system has all coherent nodes then the system has achieved synchronization.

We looked at the phase portraits of the node number 8 and 45(Refer Fig(2.1) for node indexing) when there was no delay introduced in the system and when there was a delay introduced at 45^{th} node.

When delay is absent(τ=0):

We can see in the fig(2.5a,2.5c) that each of the nodes starts off from their initial random phase and after some transition period, they settle into a stable frequency line i.e their frequencies become constant. Both of the nodes settle into the same constant frequency and so we can conclude that these nodes are synchronized. Same behaviour was observed for all the nodes in this system and so concluded that the whole network gets synchronized when no delay is introduced in the network.

When delay is present($\tau \neq$0):

From the fig(2.5) we can observe that we get two categories of nodes in this system after we let the system evolve for some time. Node 8 belongs to the first category and is an example of synchronized nodes. These nodes settle into a constant frequency just like in the case of non delayed system.

The two categories of nodes i.e coherent(ex-8^{th} node) and incoherent(ex-45^{th} node) collectively form the chimera state in this system.

So we observed how a single delayed node can lead to a chimera state. This shows that we can engineer a rippling chimera in a network of oscillators arranged in a 2D lattice by introducing a delay in one of the nodes.

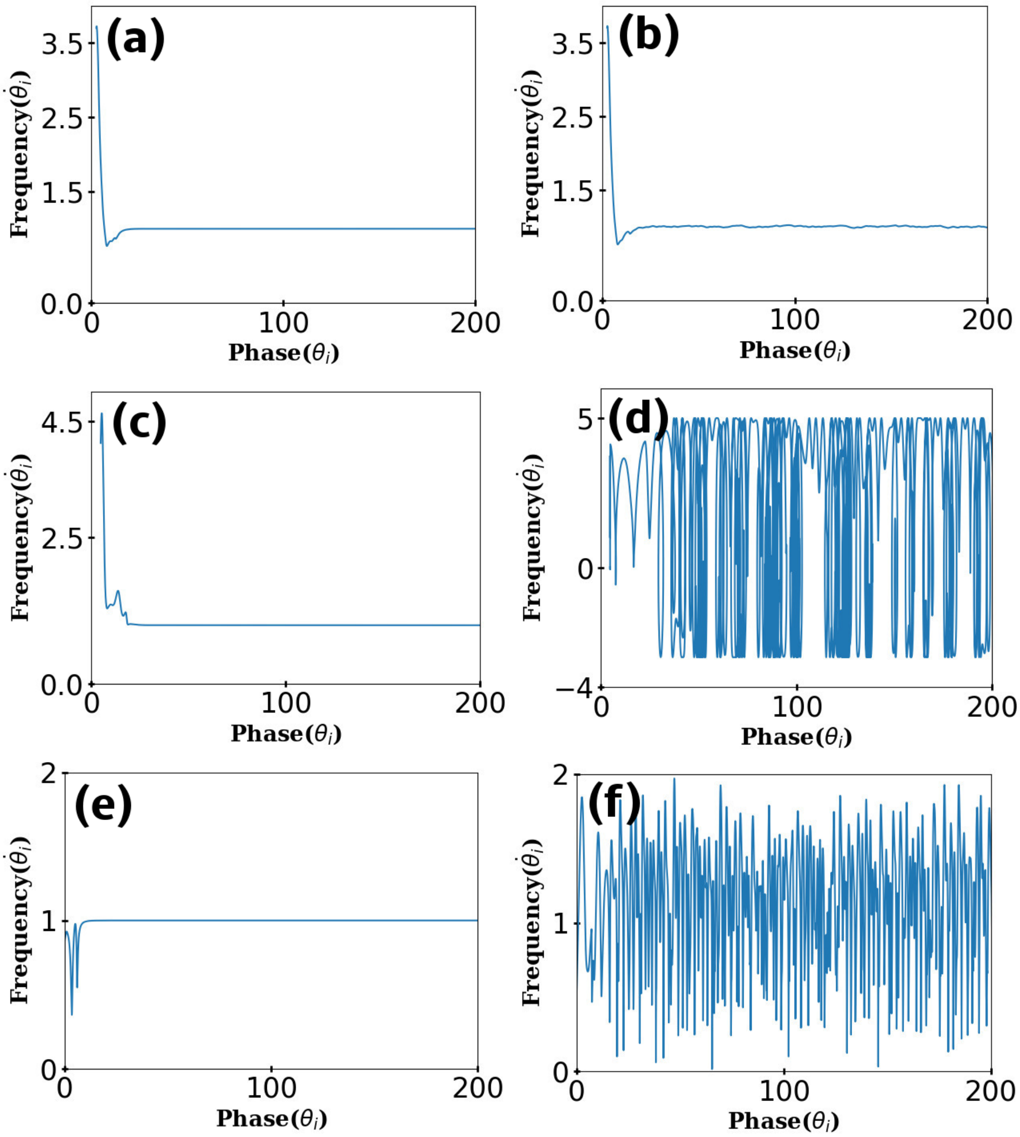

Figure 2.5: θ vs $\dot{\theta}$ for a single node in 2D lattice network. (a) θ vs $\dot{\theta}$ for 8^{th} node when no delay was applied in the network (i=8, $\mu = 1$, $N = 100$, $\tau = 0$). (b) θ vs $\dot{\theta}$ for 8^{th} node when delay was applied in the links of the 45^{th} node (i=8, $\mu = 1$, $N = 100$, $\tau = 10$). (c) θ vs $\dot{\theta}$ for 45^{th} node when no delay was applied. (i=45, $\mu = 1$, $N = 100$, $\tau = 0$) (d) θ vs $\dot{\theta}$ for 45^{th} node when delay was applied in the links of the 45^{th} node (i=45, $\mu = 1$, $N = 100$, $\tau = 10$). (e) θ vs $\dot{\theta}$ for 35^{th} node when no delay was applied (i=35, $\mu = 1$, $N = 100$, $\tau = 0$). (f) θ vs $\dot{\theta}$ for 35^{th} node when delay was applied in the links of the 45^{th} node (i=35, $\mu = 1$, $N = 100$, $\tau = 10$). (a), (c) and (e) shows that when no delay is applied then all the nodes settle into a common frequency(here the common final frequency is 1) and we can conclude that the system is synchronized. (b), (d) and (f) shows that on introducing a delay in the links of the 45^{th} node, the 45^{th} node and the nodes near to the 45^{th} node for example node number 35, 46 etc. move incoherently and the nodes which are far from the 45^{th} node for example node number 8, 87 etc move coherently(Refer Fig. 2.4). This shows that two domains exist in this system i.e incoherent and coherent domains. This concludes that introducing delay in the links of one of the nodes in a 2D lattice network results in the emergence of Chimera state.

Chapter 3

Machine Learning for Predicting Complex Dynamics

3.1 Introduction

The network was allowed to evolve in time for many different values of coupling constant and delay. In total 800 such simulations were carried out and number of drifting oscillators at the end of each simulation was noted. Phase diagram drawn using the raw data generated after the simulations is shown in Fig(3.1).

Fig(3.1) shows that the data contains a lot of noise. This noise is due to inaccuracies in the numerical simulations. By looking at the data, only one boundary can be drawn with certainty as shown in Fig(3.2). This diagram gives a good idea about synchronized and chimera state but it ignores a lot of useful information that our data contains. The exact number of drifting oscillators or the "intensity" of chimera state can't be identified by looking at this diagram.

To construct a more detailed phase diagram machine learning techniques were used.

3.2 Theoretical Framework

The algorithms of each machine learning algorithm that was used in this analysis is shown in this section. Refer Table 3.1,3.2 and 3.3 for the algo-

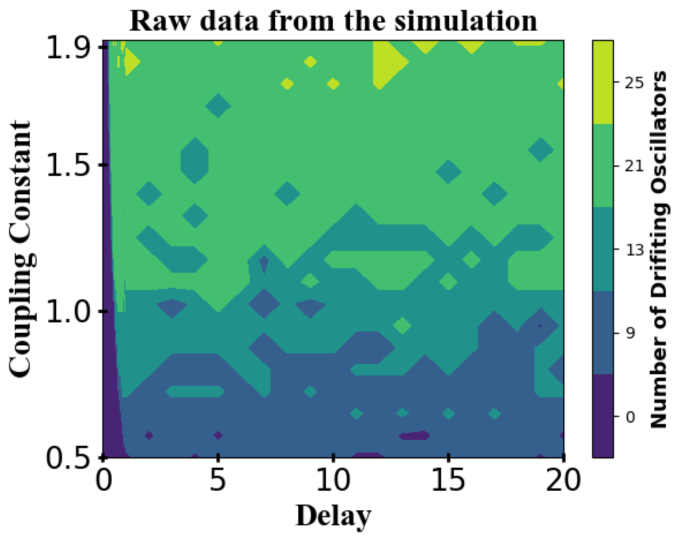

Figure 3.1: Phase diagram for 2D lattice network in two parameter space of delay and coupling constant. This phase diagram was plotted using the data that was directly obtained from the simulations. This is a filled contour plot where each colour represents a different value of "number of drifting oscillators" found in the engineered chimera state in a 2D lattice network. The colourbar at the right hand side shows number of drifting oscillators corresponding to each colour in the diagram.

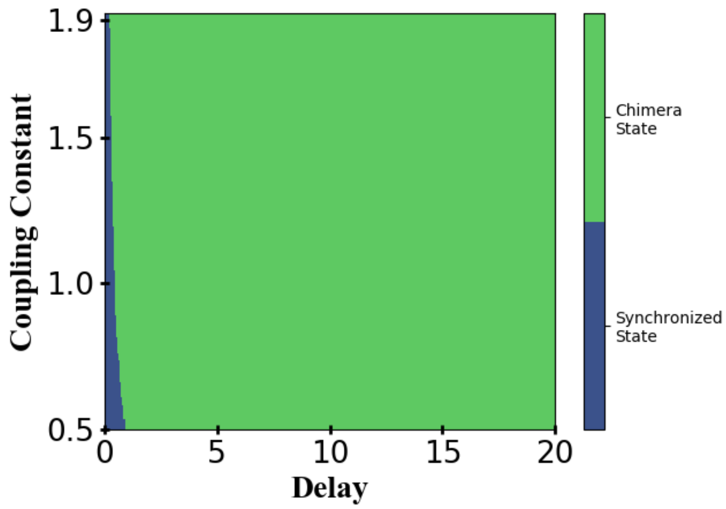

Figure 3.2: Phase diagram for 2D lattice network in two parameter space of delay and coupling constant. This is a filtered version of Fig(3.1). Any non-zero value of "number of drifting oscillators" will mean that the network is in a chimera state. So all the regions in Fig(3.1) with non-zero values of "number of drifting oscillators" was combined in this diagram. Number of drifting oscillators in blue region is zero and so it represents synchronized state. Number of drifting oscillators in green region is non zero and so it represents chimera state. This diagram is good at differentiating between synchronization region and chimera region but it lacks information about intensity of chimera for a given pair of delay and coupling constant.

Table 3.1: k nearest neighbours

Algorithm for classification using Knn Model
X = M x n is a matrix which represents the features of data where M is the total number of samples each of dimension n Y = M x 1 is a column matrix representing the labels of the data K represents the number of nearest neighbours to consider during the training phase of the model. 1) Find the optimum value of K by using validation curve analysis. While plotting validation curves use cross-validation of data and vary K from 1 to some appropriately higher value of K. 2) Once the optimum K is known, split the data into two groups randomly. The bigger chunk of data should be used as training dataset and the smaller chunk as the testing dataset. 3) Train a Knn model with optimum value of K using the training data and test this model using the testing data. 4) Note down the training and testing accuracy of the trained model. 5) Predict the label of an unclassified data point using this trained model. 6) Repeat steps 3,4 and 5 multiple times. 7) The mean of the accuracies of all the trained models will be our final accuracy of the Knn model. 8) The most frequently predicted label for the unclassified data point will be the final predicted label of that data point using Knn model.

rithms.

3.3 Results

3.3.1 Data

The data was arranged in three columns where column one contained coupling constant values, column two contained delay values and column three contained number of drifting oscillators corresponding to the pair of delay and coupling constant in that row. The data structure looked like Table 1,

There were in total 800 rows in this table.

3.3.2 Machine learning algorithms used

The data was randomly split into training and testing set in the ratio of 4:1. Parameter for Knn- Validation curve was plotted for Knn(Fig (3.3)) and it

Table 3.2: Support Vector Machine

Algorithm for classification using SVM Model
X = M x n is a matrix which represents the features of data where M is the total number of samples each of dimension n Y = M x 1 is a column matrix representing the labels of the data. 1) Find the optimum value of the regularization parameter and gamma by using grid search analysis for SVM when using rectified linear unit(ReLU) kernel. 2) Once the optimum values of the two parameters are known, split the data into two groups randomly. The bigger chunk of data should be used as training dataset and the smaller chunk as the testing dataset. 3) Train the SVM model with the optimum parameters using the training dataset and test this model using the testing dataset. 4) Note down the training and testing accuracy of the trained model. 5) Predict the label of an unclassified data point using this trained model. 6) Repeat steps 3,4 and 5 multiple times. 7) The mean of the accuracies of all the trained models will be our final accuracy of the SVM model. 8) The most frequently predicted label for the unclassified data point will be the final predicted label of that data point using SVM model.

Table 3.3: Multi Layer Perceptron Neural Network

Algorithm for classification using MLP-NN Method
X = M x n is a matrix which represents the features of data where M is the total number of samples each of dimension n Y = M x 1 is a column matrix representing the labels of the data. 1) Split the data into two groups randomly. The bigger chunk of data should be used as training dataset and the smaller chunk as the testing dataset. 2) Train the MLP-NN model using the training dataset and test this model using the testing dataset. 3) Note down the training and testing accuracy of the trained model. 4) Using experimentation determine the number of hidden layers and number of nodes in each hidden layer which gives the best accuracy for our dataset. 5) Once that is determined, train a model with the optimum number of hidden layers and number of nodes. 6) Predict the label of an unclassified data point using this trained model. 7) Note down the training and testing accuracy of the trained model. 8) Repeat steps 6,7 and 8 multiple times. 9) The mean of the accuracies of all the trained models will be our final accuracy of the MLP-NN model. 10) The most frequently predicted label for the unclassified data point will be the final predicted label of that data point using MLP-NN model.

Table 3.4: Data structure for Rippling Chimera data

Coupling Constant	Delay	No. of Drifting Oscillators
:	:	:
:	:	:
0.875	0.5	0
0.875	0.6	9
0.875	0.7	13
:	:	:
1.1	1	21
1.1	2	21
:	:	:
:	:	:

showed that for the given data, k=5 will give the best possible results that Knn is capable of giving us. So in the end, we created 1000 Knn models with a different randomly chosen training set in each iteration and then based our final prediction of number of drifting oscillators on the aggregate of all these models.

Parameters for SVM- Using grid search module available in sklearn package for python, we selected the following parameters for SVM. Kernel=Radial Basis Function(rbf), Regularization parameter('C' in sklearn module)=10, Gamma for rbf kernel=0.5 We then created 1000 SVM models with a different randomly chosen training set in each iteration and then based our prediction of number of drifting oscillators on the aggregate of all these models.

Parameters for the neural network are- Number of hidden layers=2, Number of nodes in each hidden layer=30, Activation function=Rectified Linear Unit(ReLU). We created 50 neural net models with a different randomly chosen training set in each iteration and then based our prediction of number of drifting oscillators on the aggregate of all these models.

It was observed that multi layer perceptron neural network algorithm was the best in estimating a clear phase boundary for our system and thus it was best in predicting the final intensity of the chimera state of the system.

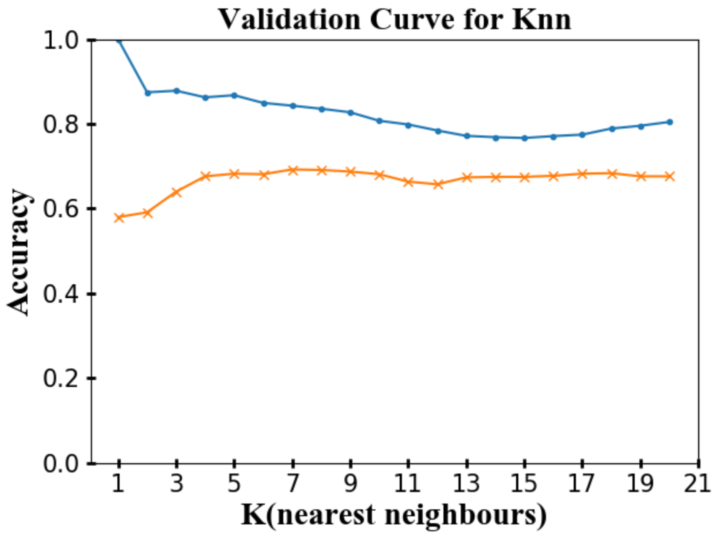

Figure 3.3: Validation curve for Knn. 5 fold cross validation of data was used while plotting these curves. The parameter K was varied from 1 to 20. Value of K at which a Knn model gives high training accuracy as well as high validation accuracy is the optimum value of K for the dataset. k=1 is not a suitable fit because in that case the training accuracy is 1 but the validation accuracy is very low. So k=1 will lead to overfitting. Using this validation curve we can see that the best possible Knn model that we can obtain for our dataset is at K=5. Blue line represents training set accuracy and orange line represents validation set accuracy.

Table 3.5: Accuracy of different algorithms in percent

Algorithm	Training Accuracy	Testing Accuracy
Knn	85.449	77.66
SVM	85.455	80.821
Neural Network	83.012	82.725

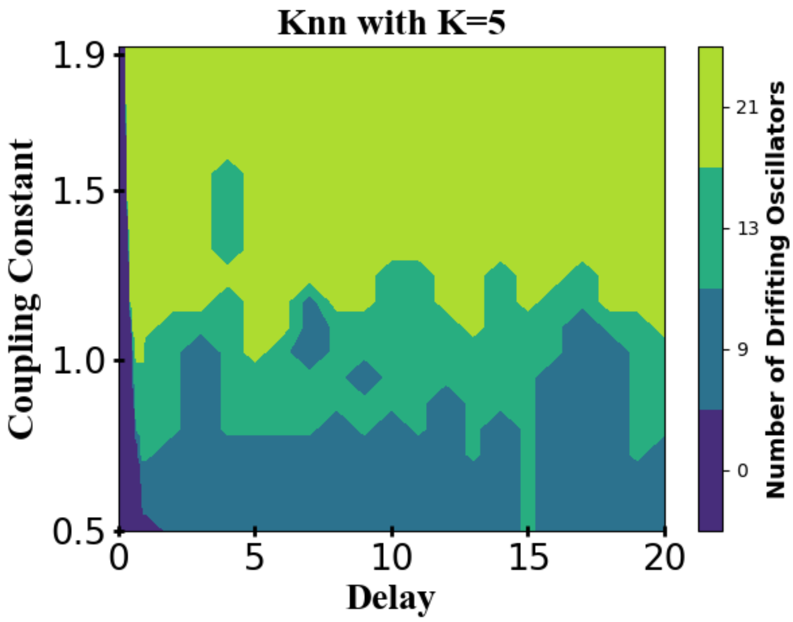

Figure 3.4: Phase diagram for 2D lattice network in two parameter space of delay and coupling constant. Region boundaries in this diagram was obtained using a trained Knn machine learning model. The value of parameter K=5 for training this model was obtained using validation curve analysis(See Fig(3.3)).This is a filled contour plot where each colour represents a different value of "number of drifting oscillators" found in the engineered chimera state in a 2D lattice network. The colourbar at the right hand side shows number of drifting oscillators corresponding to each colour in the diagram.

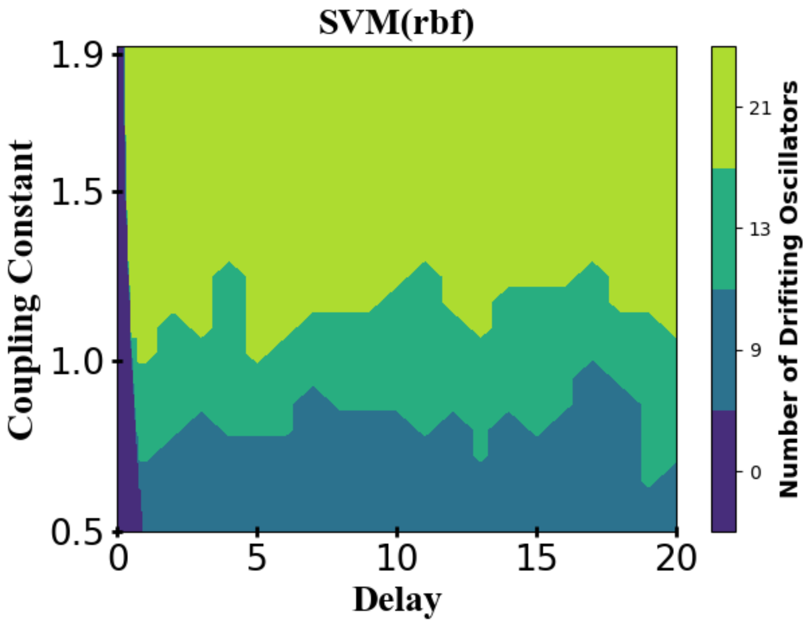

Figure 3.5: Phase diagram for 2D lattice network in two parameter space of delay and coupling constant. Region boundaries in this diagram was obtained using a trained SVM machine learning model. RBF kernal was used while training this SVM model. The value of parameters C=10 and gamma=0.5 was obtained used grid search analysis. This is a filled contour plot where each colour represents a different value of "number of drifting oscillators" found in the engineered chimera state in a 2D lattice network. The colourbar at the right hand side shows number of drifting oscillators corresponding to each colour in the diagram.

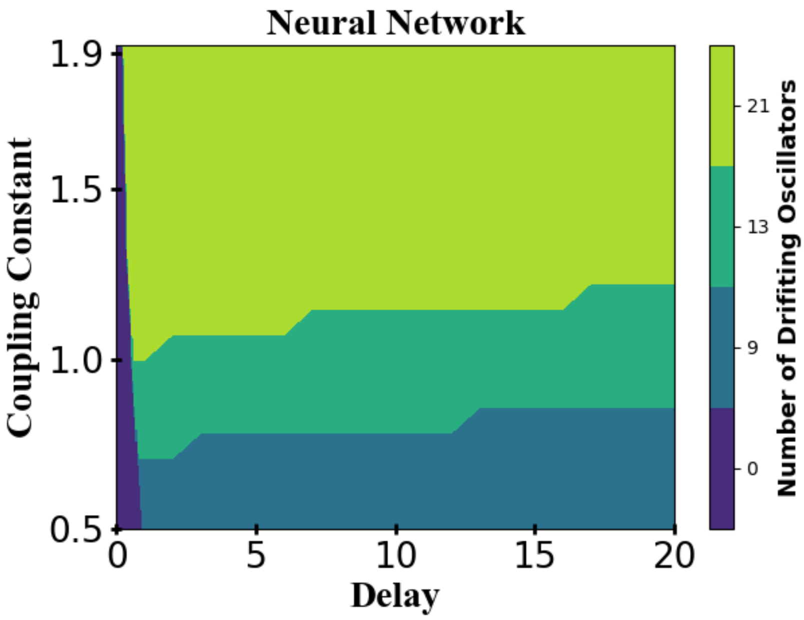

Figure 3.6: Phase diagram for 2D lattice network in two parameter space of delay and coupling constant. Region boundaries in this diagram was obtained using a trained MLP-NN machine learning model. ReLU activation function was used while training this MLP-NN model. Apart from the input and output layer, the artificial neural network used in this model contains 2 hidden layers with 30 nodes each. This is a filled contour plot where each colour represents a different value of "number of drifting oscillators" found in the engineered chimera state in a 2D lattice network. The colourbar at the right hand side shows number of drifting oscillators corresponding to each colour in the diagram.

Chapter 4

Machine learning Technique Applied on Multiplex Network

4.1 Introduction

To check the validity and the applicability of the machine learning developed in the previous chapter, it is important to test this technique on a network which is structurally very different to the 2D lattice.

In this chapter, the same machine learning technique as previous chapter was applied on a multiplex network of kuramoto oscillators. This network is different from the previous 2D lattice network in many ways. First and foremost the network structure of the two networks is completely different. Secondly on applying delay multiplex networks show solitary state instead of chimera state as observed in the case of 2D lattice. Third, the number of parameters which determine whether a multiplex network will be in synchronized state or in a solitary state is different to that of a 2D lattice network. In 2D lattice network coupling constant and delay i.e two parameters determined whether the network will be in a synchronized state or chimera state but in case of multiplex network, interlayer coupling constant, intralayer coupling constant and delay i.e three parameters determines whether the network will be in a synchronized state or solitary state.

Multi layer perceptron neural network was used to train a machine

Table 4.1: Data structure for Solitary data

Interlayer Coupling	Intralayer Coupling	Delay	State
:	:	:	:
:	:	:	:
3.60	0.86	0.50	0
3.60	0.88	0.50	0
3.70	0.1	0.50	1
:	:	:	:
2.00	0.20	2.00	1
2.00	0.22	2.00	0
:	:	:	:
:	:	:	:

learning model. Using this trained model exact value of the critical delay can be calculated for any pair of inter and intra layer coupling constant.

4.2 Results

4.2.1 Data

The network was allowed to evolve in time for many different values of inter and intra layer coupling constant. In total 1600 such simulations were performed and the frequency difference between delayed node and rest of the synchronized nodes was noted at the end of each simulation. This was done for 5 different values of delay i.e 0,0.5,1,2,4. So the total number of simulations that were performed was 8000.

The data was arranged in four columns where column one contained interlayer coupling constant values, column two contained intralayer coupling constant values, column three contained delay value and column four contained 0 for synchronized state and 1 for solitary state. A system is solitary or not was decided by looking at the frequency difference between the excited(delayed) node and the synchronized node. If the frequency difference was more than 0.01 then the system's state was classified to be in solitary state. Here, 0.01 is called the threshold value for solitary state. The data structure looked like Table 3,

There were 8000 rows in total in this table.

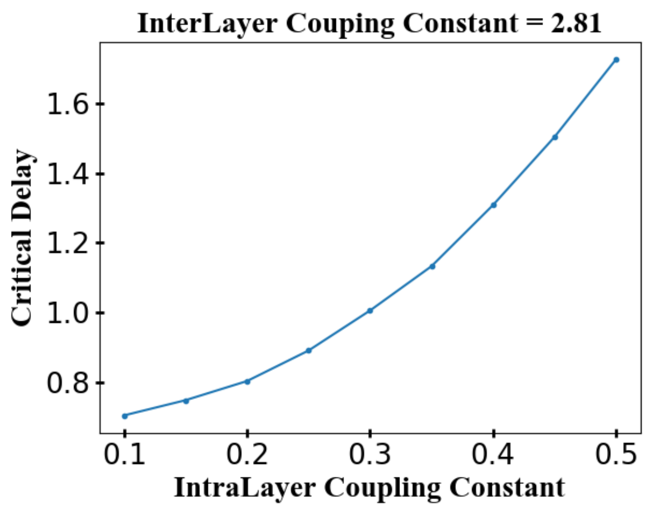

Figure 4.1: This figure shows the critical delay as a function of intralayer coupling constant for a multiplex network. Interlayer coupling constant=2.81 and is a constant here. The values of critical delay has been calculated using a trained MLP-NN machine learning model.

4.2.2 Machine learning model

Multi layer perceptron neural network was used for creating a prediction model using our data. The data was randomly split into training and testing set in the ratio of 4:1. Parameters for the neural network are- Number of hidden layers=2, Number of nodes in each hidden layer=30, Activation function=Rectified Linear Unit(ReLU). We created 50 neural net models with a different randomly chosen training set in each iteration and then averaged over the results of each model to get a final value of the critical delay.

4.2.3 Critical delay for solitary state

4.2.4 Impact of threshold parameter

The impact of the frequency difference threshold set for differentiating solitary state and synchronized state was also studied.(See Fig(4.4)) It was observed that if the value of threshold frequency difference is low then changing the threshold value doesnot have any significant effect on the prediction of the critical delay but as soon as the threshold is changed to a larger value such as value greater than 0.03 then the machine learning

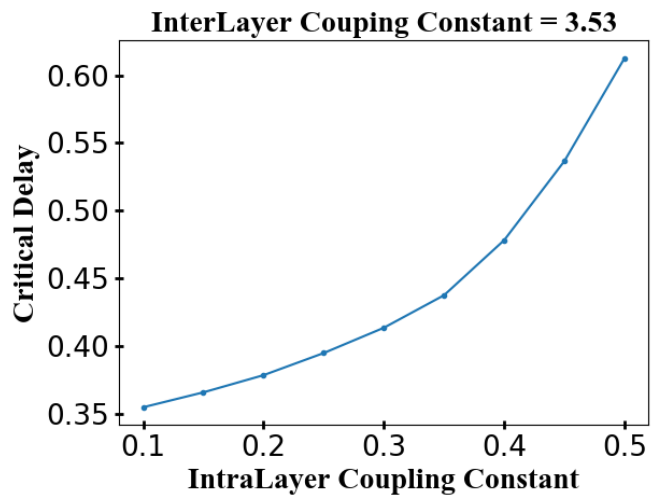

Figure 4.2: This figure shows the critical delay as a function of intralayer coupling constant for a multiplex network. Interlayer coupling constant=3.53 and is a constant here. The values of critical delay has been calculated using a trained MLP-NN machine learning model..

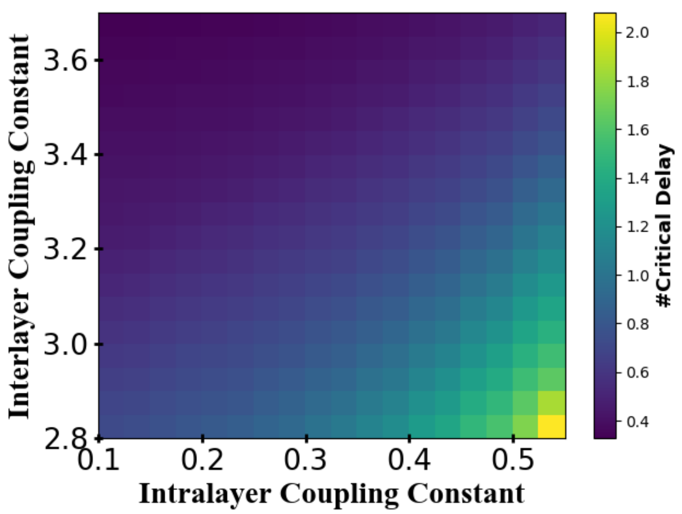

Figure 4.3: Heatmap of intralayer coupling constant vs interlayer coupling constant for a multiplex network. Colour of each block represents the value of critical delay corresponding to inter and intra layer coupling constant of that block. Here the critical delay values are calculated using a trained MLP-NN machine learning model. The heatmap shows how the critical delay varies with respect to an increase or decrease in the value of inter and intra layer coupling constant. The colorbar on right represents the value of critical delay corresponding to the colours in heatmap..

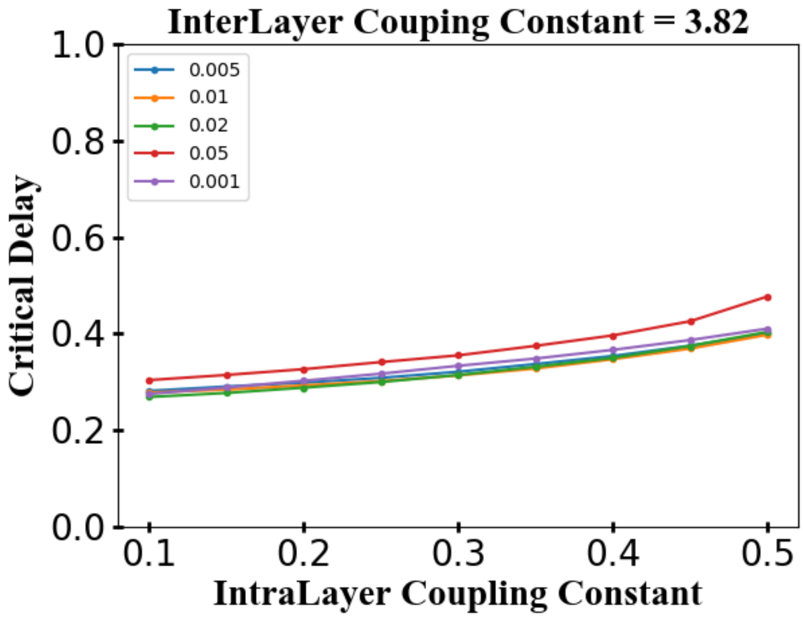

Figure 4.4: This figure shows the critical delay as a function of intralayer coupling constant for a multiplex network. Interlayer coupling constant=3.82 and is a constant here. The different lines in the plot corresponds to different threshold values of frequency difference between the excited node and the synchronized nodes which is used to identify if a given state is solitary or not. The plot suggests that changing the threshold value between 0.001 and 0.02 doesnot have any significant change in the predicted value of the critical delay using machine learning model. Once the threshold value exceeds 0.03, we start observing bad predictions by the machine learning model.

model starts giving wrong predictions.

Chapter 5

Conclusion and Future Scope

Applying delay on one of the nodes in a 2D lattice network can lead to emergence of chimera state. This tells us that the technique used by Dr. Ghosh et al[12] is valid not only for 1D ring network but also for higher order network structures. We also observed that we can control the spread of incoherent region in a rippling chimera by changing two parameters i.e delay and coupling constant.

A new and novel machine learning technique to predict dynamical features of a complex network was developed in this project. First it was tested on 2D lattice network to predict intensity of chimera and to generate more detailed phase space diagram for the network.

Then, the same machine learning technique was tested on a totally different network structure i.e multiplex network. Here, the technique was used to predict the exact value of critical coupling constant for a given value of inter and intra layer coupling constant. It was observed that this technique worked very well on this network structure too.

Getting good results in two completely different types of network system gives a nod towards the validity and the applicability of this new technique in diverse network systems.

It was observed that for predicting dynamical properties using dynamical parameter, MLP-NN is the best suited algorithm when compared with Knn and SVM.

5.1 Future Direction

It has been observed that in real world systems it is fairly uncommon to find complete synchronization. Partial synchronization is much more common than full synchronization in real world systems. For example, neurons in brain network are more likely to show partial synchronized behaviour than fully synchronized behaviour. Chimera has also been associated with cognitive process in human brain network[41] and uni-hemispheric sleep found in other mammals[42, 43]. So studying chimera states is of great importance.

The technique used in this project to engineer a chimera state can be applied to various other network structures. This particular project can also be extended by performing the analysis shown in chapter 2 on more complicated network structures such as 3D lattice network. Rippling chimera can also be engineered on a 2D lattice network of bigger size and any difference in properties due to change in network size can be studied by that.

Since the machine learning technique shown in chapter 3 and 4 is a completely new idea, it has a lot of potential to be explored in the future. This technique can be used in many different types of network with different network architecture. It can also be explored on systems with completely different models such as Fitzhugh Nagumo model, coupled maps etc. This method can potentially be used to predict numerous other dynamical properties of a system which were not shown in this thesis. The possibilities with this technique are tremendous.

Apart from the applications, in future one can also work on improving this technique and try other more complicated machine learning algorithms to perform the same analysis that was performed in this thesis using Knn, SVM and MLP-NN.

Bibliography

[1] Jalan, Sarika, and Camellia Sarkar. "Complex Networks: an emerging branch of science." Phys. News 47, 42-51 (2017).

[2] Jalan, Sarika, et al. "Uncovering randomness and success in society." PloS one 9.2 (2014): e88249.

[3] Kuramoto, Yoshiki, and Dorjsuren Battogtokh. "Coexistence of coherence and incoherence in nonlocally coupled phase oscillators." arXiv preprint cond-mat/0210694 (2002).

[4] Abrams, Daniel M., and Steven H. Strogatz. "Chimera states for coupled oscillators." Physical review letters 93.17 (2004): 174102.

[5] Andrzejak, Ralph G., Giulia Ruzzene, and Irene Malvestio. "Generalized synchronization between chimera states." Chaos: An Interdisciplinary Journal of Nonlinear Science 27.5 (2017): 053114.

[6] Santos, M. S., et al. "Chimera-like states in a neuronal network model of the cat brain." Chaos, Solitons Fractals 101 (2017): 86-91.

[7] Lazarides, N., G. Neofotistos, and G. P. Tsironis. "Chimeras in SQUID metamaterials." Physical Review B 91.5 (2015): 054303.

[8] Larger, Laurent, Bogdan Penkovsky, and Yuri Maistrenko. "Virtual chimera states for delayed-feedback systems." Physical review letters 111.5 (2013): 054103.

[9] Bera, Bidesh K., Dibakar Ghosh, and Tanmoy Banerjee. "Imperfect traveling chimera states induced by local synaptic gradient coupling." Physical Review E 94.1 (2016): 012215.

[10] Abrams, Daniel M., et al. "Solvable model for chimera states of coupled oscillators." Physical review letters 101.8 (2008): 084103.

[11] Strogatz, Steven H. "From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators." Physica D: Nonlinear Phenomena 143.1-4 (2000): 1-20.

[12] Ghosh, Saptarshi, and Sarika Jalan. "Engineering chimera patterns in networks using heterogeneous delays." Chaos: An Interdisciplinary Journal of Nonlinear Science 28.7 (2018): 071103.

[13] Kuramoto, Yoshiki. "Chemical Turbulence." Chemical Oscillations, Waves, and Turbulence. Springer, Berlin, Heidelberg, 1984. 111-140.

[14] Newman, M. (2018). Networks. Oxford university press (DOI: 10.1093/oso/9780198805090.001.0001)

[15] Y. Kuramoto, D. Battogtokh, Nonlinear Phenomena in Complex Systems, 5(4), 380385 (2002).

[16] D.M. Abrams, S. H. Strogatz, Physical review letters 93(17), 174102 (2004).

[17] Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20.3 (1995): 273-297.

[18] Guo, Gongde, et al. "KNN model-based approach in classification." OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". Springer, Berlin, Heidelberg, 2003.

[19] Burkov, Andriy. The hundred-page machine learning book. Quebec City, Can.: Andriy Burkov, 2019.

[20] Mitchell, Thomas M. "Machine learning." (1997).

[21] Ghasemian, Amir, et al. "Detectability thresholds and optimal algorithms for community structure in dynamic networks." Physical Review X 6.3 (2016): 031005.

[22] Rivkind, Alexander, and Omri Barak. "Local dynamics in trained recurrent neural networks." Physical review letters 118.25 (2017): 258101.

[23] Pathak, Jaideep, et al. "Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach." Physical review letters 120.2 (2018): 024102.

[24] Ni, Qi, et al. "Machine learning dynamical phase transitions in complex networks." Physical Review E 100.5 (2019): 052312.

[25] Fan, Huawei, et al. "Long-term prediction of chaotic systems with machine learning." Physical Review Research 2.1 (2020): 012080.

[26] Ganaie, M. A., et al. "Identification of Chimera using Machine Learning." arXiv preprint arXiv:2001.08985 (2020).

[27] Jiang, Ziping, and Martin McCall. "Numerical simulation of a large number of coupled lasers." JOSA B 10.1 (1993): 155-163.

[28] Kourtchatov, S. Yu, et al. "Theory of phase locking of globally coupled laser arrays." Physical Review A 52.5 (1995): 4089.

[29] Wiesenfeld, Kurt, Pere Colet, and Steven H. Strogatz. "Synchronization transitions in a disordered Josephson series array." Physical review letters 76.3 (1996): 404.

[30] Wiesenfeld, Kurt, Pere Colet, and Steven H. Strogatz. "Frequency locking in Josephson arrays: Connection with the Kuramoto model." Physical Review E 57.2 (1998): 1563.

[31] York, Robert A., and Richard C. Compton. "Quasi-optical power combining using mutually synchronized oscillator arrays." IEEE Transactions on Microwave Theory and Techniques 39.6 (1991): 1000-1009.

[32] Buck, John. "Synchronous rhythmic flashing of fireflies. II." The Quarterly review of biology 63.3 (1988): 265-289.

[33] Buck, John, and Elisabeth Buck. "Synchronous fireflies." Scientific American 234.5 (1976): 74-85.

[34] Peskin, Charles S. "Mathematical aspects of heart physiology." Courant Inst. Math (1975).

[35] Michaels, Donald C., Edward P. Matyas, and Jose Jalife. "Mechanisms of sinoatrial pacemaker synchronization: a new hypothesis." Circulation research 61.5 (1987): 704-714.

[36] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12 (2011): 2825-2830.

[37] Zakharova, Anna, Marie Kapeller, and Eckehard Schöll. "Chimera death: Symmetry breaking in dynamical networks." Physical review letters 112.15 (2014): 154101.

[38] Sethia, Gautam C., and Abhijit Sen. "Chimera states: The existence criteria revisited." Physical review letters 112.14 (2014): 144101.

[39] Ujjwal, S. R., Punetha, N., Prasad, A., and Ramaswamy, R. (2017). Emergence of chimeras through induced multistability. Physical Review E, 95(3), 032203 (DOI: 10.1103/PhysRevE.95.032203)

[40] Ulonska, Stefan, et al. "Chimera states in networks of Van der Pol oscillators with hierarchical connectivities." Chaos: An Interdisciplinary Journal of Nonlinear Science 26.9 (2016): 094825.

[41] Bansal, Kanika, et al. "Cognitive chimera states in human brain networks." Science advances 5.4 (2019): eaau8535.

[42] Panaggio, Mark J., and Daniel M. Abrams. "Chimera states: coexistence of coherence and incoherence in networks of coupled oscillators." Nonlinearity 28.3 (2015): R67.

[43] Ramlow, Lukas, et al. "Partial synchronization in empirical brain networks as a model for unihemispheric sleep." EPL (Europhysics Letters) 126.5 (2019): 50007.