Machine learning applications in complex systems

M.Sc. Thesis

By

Atish Panday

Department of Physics

Indian Institute of Technology

Indore

June 2021

Machine learning applications in complex systems

A Thesis

Submitted in partial fulfillment of the requirements

for the award of the degree

of

Master of Science

By

Atish Panday

Department of Physics

Indian Institute of Technology Indore

June 2021

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **Machine learning applications in complex systems** 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 June 2019 to June 2021 under the supervision of Dr. Sarika Jalan, Professor of Physics, IIT 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.

(Atish Panday)

.....

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

(Dr. Sarika Jalan)

.....

Atish Panday has successfully given his M.Sc. Oral Examination held on

Thesis Supervisor

Dr. Sarika Jalan

Date:

Convener, DPGC

Date: 24-06-2021

PSPC Member

Dr. Manavendra Mahato

Date: 19/06/2021

PSPC Member

Dr. Dipankar Das

Date: 19/06/21

ACKNOWLEDGEMENTS

First and foremost, I want to acknowledge my project advisor Dr. Sarika Jalan for her constant support and guidance. Her encouragement and support during the project was very helpful, it helped me to come out of every difficulty I faced during this work. Her suggestions and remarks are an integral part of this work.

Also, I want to acknowledge the PSPC members Dr. Manavendra Mahato and Dr. Dipankar Das. Their remarks have played a major role in the development of this work.

A special thanks to my senior lab mates: Dr. Ajaydeep Kushwaha, Mr. Subhasanket Dutta and Mr. Ankit Mishra for their constant support and cheerful cooperation.

I would like to distinctly acknowledge Dr. Woo Seok Lee for his incredible contribution to this work. He designed the machine learning model and provided all the results generated from it. It would not have been possible to complete this work without his help.

(Atish Panday)

ABSTRACT

We observe complex interacting systems on a daily basis. The vast internet itself, the network of roads, the network of neurons comprising our nervous system, and so on. Their behaviors have been a matter of interest for decades. Much of what is understood about the dynamics of these complex systems comes from mathematical estimations and continuous observations of their time series. To the same effect, there has been recent developments in machine learning techniques to better understand and predict behaviours of these dynamical systems. The main idea behind these techniques is to observe patterns in the data generated from these complex systems, and allow the machine to learn them in order to make certain predictions about the network. In this work, we explore one such novel application of machine learning techniques to unravel some fundamental structure-to-dynamics relations that better help understanding these complex systems. We classify different types of networks based on their inherent structural differences by training a CNN model on the time-series of a few highest degree nodes. The novelty of our work lies in the observation that using only a limited time-series information of a large network, we make extremely accurate classifications, and prove that with increasing the size of the networks, the number of time-series required remains the same.

Table of Contents

Chapter 1

Introduction

1.1 Complex systems

Every system of entities can be expressed as a network of nodes connected by edges. The term "entity" here can be used to describe any type of tangible or intangible unit that can in some way interact with another equal or unequal entity. Thus, in the network of social media, every user using a particular platform to interact with other users can be considered as a node of that network. The interaction between users can be considered as an edge, which can be further quantified by the strength of interactions by assigning weights to these edges. For example, user A interacts with user B on a daily basis, but not quite as often with user C. Thus a weight of the edge between A and B would be higher than the weight between the user A and user C. This abstraction of data in a manner where the nodes and edges and their dynamics can be used to model a whole system, is called a complex system. Thus, virtually any system can be broken down into a set of units and interactions between them. The larger the system gets, the more complex their interactions and their dynamics become. Network science exists precisely to deal with this very complexity in systems by reducing their behavior down to a scale of interactions between only a pair of nodes.

1.2 Dynamics of complex systems

Nodes and edges carry valuable information about the entities and their interactions. However, these nodes often have more data associated with them that changes over time. For example, different regions in the brain output EEG signals that change with time. This information about the regions of the brain are extremely important in studying epileptic seizures. Thus it is not only important to know the connectivity structure of the networks, but also their dynamics. Nodes and edges make up the structural basis of the system, or the spatial data, and the dynamics of the nodes make up the temporal data. The most interesting question that hereby arises, is, how might they be related? This is where the concepts of predicting the dynamics of the network, given its structure, and vice versa come into play. It's an easy question to answer if we look at really small systems. For example, consider two friends who go to school together. One day, friend A decides to call up his friend B and they decide to skip school that day. On this scale, the presence or absence of B in the school is completely dependent on the presence or absence of A in the school. Thus, given the attendance record of A, it may be very easy to predict the same for B. On a larger scale however, with a group of 50 or more students, with each student having multiple friends, the prediction becomes much more complicated. Every student is influenced by his/her friends, who in turn are influenced by their other friends. This creates a complex series of uncertain values that we call dynamics of complex systems.

1.3 Different types of networks

Networks can be directed, undirected, multi-edged, or multilayered. These can be further divided based on their structural properties such as degree distributions, clustering coefficients, shortest path lengths, density, etc. The main types of networks based on structural differences are detailed below:

1.3.1 Erdős–Rényi random networks

Erdős–Rényi random networks are generated by connecting labeled nodes randomly. The existence of an edge between any two nodes has a probability p. Thus, the value of p decides the average degree of the network, and thereby the density of the network.
Networks generated in this manner tend to have a very uniform distribution of degrees, i.e., the difference in the degrees of the node with the highest degree and the node with the lowest degree is relatively low. Thus the degree distribution graph of these random networks, assuming sufficiently large network size, has the shape of a Poisson distribution.

1.3.2 Scale-free networks

Scale-free networks are the most commonly observed type of networks. The social network, the internet, the network of roads, the telephone network, all have scale-free structures. Scale-free networks are constructed by the method of preferential attachment. Thus, starting with 2 connected nodes, further nodes are attached to the existing nodes preferentially to nodes with higher degrees. This creates a network where very few nodes have very high degrees and most nodes have very low degrees. Thus, the degree distribution of these networks have the typical shape of a power-law distribution.

1.3.3 Small-world networks

Small-world networks are characterised by their large clustering coefficients and small average path lengths. Put another way, nodes in these networks tend to form clusters, with each cluster having high densities of edges between the members of the cluster, and likewise, a very low number of hops to get from one node to another. The construction of these networks is through a mechanism called the Watts-Strogatz mechanism.

1.4 Dynamical models

There are various mathematical models to mimic the dynamics of real world networks as closely as possible. Some of these models are Kuramoto oscillators model, Rössler attractors model, Lorentz oscillators model, and so on. In this work, I explore two of these models: the Kuramoto model and the Rössler model.

1.4.1 Kuramoto model

Consider a network of N nodes, represented by an adjacency matrix A. Then, for each node of the network, an initial phase and a frequency is assigned. The network then evolves according to the governing equations of the Kuramoto model.
The governing differential equations for this model are as below:

$$\dot{\theta}_i = \dot{\omega}_i + \lambda \sum_{j=1}^{N} A_{ij} \sin(\theta_j - \theta_i) \tag{1.1}$$

Here, ω_i is the frequency of the ith node of the network, θ_i is the phase of the ith node and λ is the coupling strength of the nodes.

1.4.2 Rössler model

The Rössler oscillator model is a system of 3 differential equations that exhibit chaotic dynamics of the nodes. Each node is assigned a vector of 3 dimensions, x, y and z. Each node is given a frequency ω_i. The equations governing these oscillators are:

$$\begin{aligned} \dot{x_i} &= -\omega_i y_i - z_i + \epsilon \sum_{j=1}^{N} A_{ij}(x_j - x_i) \\ \dot{y_i} &= \omega_i x_i + a y_i \\ \dot{z_i} &= f + z_i(x_i - b) \end{aligned} \tag{1.2}$$

Here, ϵ is the coupling constant in the x dimension. The variables x, y and z together represent the dynamic state of the nodes. Solving the

above numerically for the values of x, y and z gives us the chaotic time series of the nodes.

1.5 Problem statement

We aim to use the Kuramoto model time-series information of the nodes of the networks to classify them based on differences in their basic structural properties such as degree distribution. Consider a complex network containing N nodes and average degree m. We intend to classify the networks as Erdős–Rényi random, scale-free or small-world network, based on the time-series of only a few highest degree nodes. To achieve this, we employ a CNN machine learning model. The model takes in as input the time-series of the nodes and gives as output the label of the network, whether ER, SF or SW. The important point to note here is that the time-series supplied to the machine learning model here does not include the transient time required for the network to reach a steady state, and that the time-series is generated for such a value of the coupling strength λ, that the nodes are neither completely synchronized, nor completely non-synchronized.

To further explore our success of machine learning in predicting underlying structure from dynamics, we examine the prediction of the correlation matrices formed by the Rössler time-series of the nodes of a network, again using only a limited number of nodes time-series as input.

Chapter 2

Previous work

The study of complex networks and their dynamics has been a hot area of scientific research. Most recently, machine learning techniques have been constantly applied to study the dynamics of complex networks and have provided deeper insights into the rich information contained in these complex systems. Studying the network structure and its dynamics and finding the relationship between the two has been the center of attention in this field. State of the art machine learning techniques such as reservoir computing has been methodically applied for predictions of various dynamical parameters. For example, reservoir computing was successfully applied to estimate the lyapunov exponents of chaotic dynamics by using the reservoir to forecast long time-series[1], and in another instance, was applied to predict the spiking and bursting dynamics of globally coupled networks[2]. In predicting phase transitions of epidemic spreading dynamics of complex networks, [3] employed a framework combining supervised and unsupervised machine learning techniques. More than 96% accuracy was achieved for identifying the chimera states using machine learning in Kuramoto dynamical model applied to complex networks[4]. In a similar domain, successful forecasting of turbulent chimeras in simulated arrays of coupled superconducting quantum interference devices (SQUIDs) or lasers have been made using LSTMs[5]. [6] have shown the success of simple feed forward neural networks in detecting multiple types of order parameters from raw state configurations. In this space, convolutional neural

networks have shown great applicability specifically in classification problems such as in distinguishing various random electron states in quantum phase transitions[7, 8]. Classification of time-series based on its chaotic or non-chaotic nature [9] using CNN machine learning model has shown that deep underlying nature of the behavior of the dynamics of networks can be learnt and successfully applied to identify key features of the dynamics. Conversely, attempts have been made at forecasting the time-series of networks by studying the network structure. The hypothesis that the dynamics of a complex system largely depends on the underlying intricate structure of the network was successfully proven by [10] by estimating the outbreak size of an epidemic starting from a single node using machine learning techniques. The immense success of machine learning in studying complex systems becomes clear from the above results. In an opposite direction to that of [10], we explore a highly sought after problem of detecting the network structure from its dynamics. We employ a CNN model as well, to classify different types of networks, by using the time-series information of only the top few highest degree nodes, which prove to carry sufficient information required to make predictions on a global scale. We emphasise on the fact that indeed only a limited number of highest degree nodes is sufficient to distinguish between different classes of networks, irrespective of the size of the network. In the classical domain, this problem has been approached using the statistical similarity measures of the time-series generated from the networks, for example, a measure of the mutual information rate of the time-series for identifying the connectivity structure of the network has been applied[11]. In another attempt, a statistical similarity matrix was constructed using the time-series generated from coupled Kuramoto and Rössler oscillatos to find missing links in the networks[12], and was argued that the missing links were found with perfect accuracy under suitable controlled parameters. Other classical methods such as calculating a score function from structural variables such as common neighbors, Adamic/Adar, path lengths, density, etc., combined with supervised and unsupervised machine learning techniques have been applied to predict the

connectivity structures [13] and have shown better accuracy than previously known methods without machine learning. In a more recent attempt, time-series data of the networks has been used as input for a reservoir computing model, which is trained to replicate the dynamics of the networks from the given input for then inferring the missing links in the networks [14]. Motivated by the active research being done in the field, we explored another possible advantage of machine learning techniques in identifying the underlying network structure, by allowing the machine to learn only from a small subset of the time-series of the nodes. We established in this work that a small number of time-series of the nodes with the highest degrees could be used to identify with high accuracy, the underlying structure of the network. Even more surprisingly, the number of time-series required to achieve the same level of accuracy is independent of the size of the network, and hence can be extended to larger networks without affecting the error rates. We consider the Kuramoto oscillators for generating the time-series of the nodes, which is an excellent model for mimicing the time-series of real world networks such as the brain [15, 16], power-grid systems [17], epidemic spreading networks, transport systems, networks of pacemaker cells in the heart [18], congregations of synchronously flashing fireflies [19], chirping of crickets in unison [20], an array of lasers [21], superconducting Josephson junctions [22].

Chapter 3

Methodology

3.1 Data generation of the Kuramoto oscillators

In equations given in 1.1, the values of θ_i can obtained by solving them using the fourth order Runge-Kutta method. To generate the data, we first determine a suitable value of the coupling constant λ such that we make sure that the nodes are not completely synchronized or non-synchronized. The reason for this is that, after synchronization, the time-series of each node becomes completely identical. Thus no useful information can be extracted from it since the dynamical interactions between the nodes is lost. On the other extreme, in the completely non-synchronized region, the nodes have very weak interactions, and thus exhibit almost random behaviour in their dynamics. Thus, most of the information of the nodes is contained in the semi-synchronized region. That is, when after a long time, the nodes tend to synchronize only partially.

In our case, we chose 2 values of the coupling strength λ such that the nodes remain semi-synchronized after the transient time. We then used these values of λ to solve the Kuramoto differential equations using the Runge-Kutta method with an increment of $h = 0.01$.

3.1.1 Order parameter vs lambda

The order parameter of Kuramoto oscillators defines the amount of synchronization in the network. That is, it gives a measurable quantity to represent how synchronized the network is for a given time. It is defined as the following:

$$re^{i\psi} = \frac{1}{N}\sum_{j=1}^{N} e^{i\theta_j} \tag{3.1}$$

where r is the order parameter, ψ is the average phase of the oscillators, N is the number of nodes in the network, and θ_j is the phase of the jth node. Taking the time average of the order parameter values for a sufficiently long time-series length, just after the transient time gives the degree of synchronization in the system for a given value of λ. r becomes 0 when the nodes are completely incoherent, and 1 when the nodes become completely synchronized. Thus, in order to find the semi-synchronized region, we choose the value of λ such that the value of r lies somewhere between 0 and 1. Thus, by varying λ and calculating the values of order parameter, we can plot r vs. λ graph. From this graph, we can find the value of λ which gives us the semi-synchronized region.

3.1.2 Generation of networks

We generate 3 types of networks outlined in 1.3. We denote these three networks as ER, for Erdős–Rényi random network, SF, for scale-free networks, and SW, for small-world networks. The ER networks are generated by randomly making connections between nodes, with a probability p, such that $N * p$ determines the average degree of the network. We generate the SF networks by the Barabasi-Albert model, which follows the method of preferential attachment. Each new node is added to the network by connecting to $\langle k \rangle$ nodes, $\langle k \rangle$ being the average degree of the network. The probability that the new node connects to the ith node is $\langle k_i \rangle$ / $\sum_{j=1}^{M} \langle k_j \rangle$, where M is the number of nodes already included in the network. The SW networks are generated by the Watts Strogatz algorithm. First, a regu-

lar ring network of N nodes is created with each node having $\langle k \rangle$ degree, connected to $\langle k \rangle/2$ nodes on either side. The nodes are then rewired with a probability p_r, where p_r lies between 0 and 1, 0 corresponding to the original ring lattice, and 1 corresponding to a completely random network.

3.1.3 Generation of time-series

We generate the time-series of the Kuramoto oscillators by solving the Kuramoto differential equations numerically using the fourth order Runge-Kutta method. We take the step size to be $h = 0.01$ and run the simulation for a total number of 50,000 iterations, i.e., for a total time of 500. Then, we flush out the initial transient of 300 and store the remaining 200 phase values of the nodes. We then create two sets of data: (i) by taking the actual time-series of the nodes for a total of 200 after transient, and (ii) by taking the symbolic time-series, $s(t)$, of the nodes by course-graining the actual time-series using the following logic: $s(t)$ is taken as 1 when $\theta(t) < \theta(t + 0.01)$, and otherwise 0.

3.2 Convolutional neural networks (CNN)

We employ a convolutional neural network model to classify the networks using the time-series. The structure of the CNN model used is as follows: the first layer is the input layer, having the shape of the input time-series. Then we have 3 convolutional layers, having 80, 80 and 40 filters consecutively. After the convolutional layers, there is an average graph pooling layer, which is followed by 2 fully connected layers with 80 neurons each. Finally, we have the output layer with the shape of the desired output, which in our case is the number of classes in which the networks are getting classified. A rough overall structure of the CNN model is given diagrammatically below (Fig. 3.1).

We divide the input into the training and testing set, in the ratio of 4:1. Thus, generating the time-series data for 500 networks, we train the model on 400 networks and test for the remaining 100 networks. The

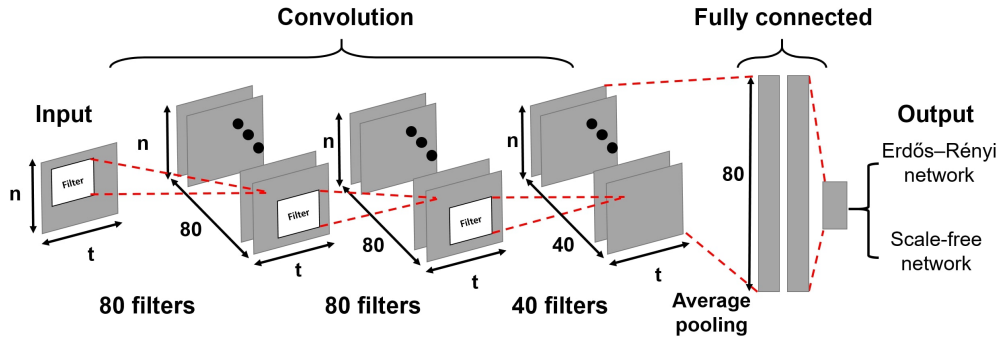

Figure 3.1: Outline of the machine learning model used

results provided are an average of the test data sets. Each sample provided to the machine learning model has the shape of (t, n), where t is the length of the time-series (i.e. the time-series recorded after transient time) and n is the number of nodes for which the time-series is being provided. We do not provide the input for one shape of the sample, but instead vary the sample shape by increasing the number of nodes and the length of time-series used. Thus, we can analyze the change in error of classification with different sample shapes.

Chapter 4

Experimentation and results

4.1 Classification of ER and SF networks

We begin by finding the values of λ for which the nodes are in the semi-synchronized region. Thus, we plot the r vs λ graphs (Fig. 4.1) for networks of size $N = 100$, average degree $\langle k \rangle = 20$ and $N = 500$, $\langle k \rangle = 10$. From the plots, 2 values of λ are chosen for each network size. For the case of $N = 100$, these values are 0.085 and 0.09, and for $N = 500$, the values are 0.15 and 0.17. We generate 250 realisations of the ER and SF networks. We take the frequency values ω_i of the nodes from a Gaussian distribution with mean 0 and variance 1, unique for each network.

For each network, we assign an initial uniformly selected random phase value to each node from a range of [-π, π].

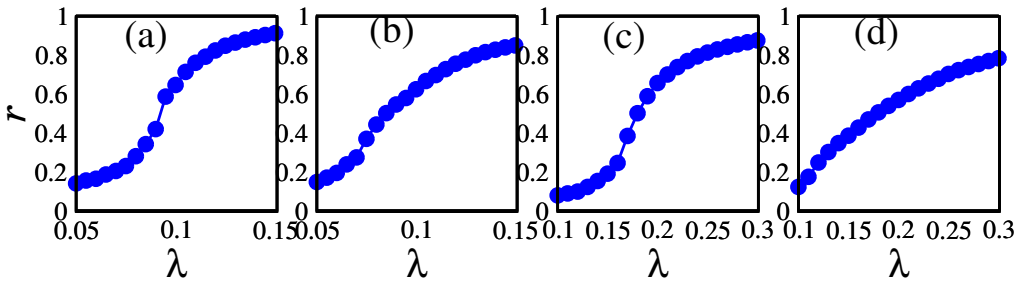

Figure 4.1: Order parameter (r) (Eq. 3.1) as a function of coupling strength λ. (a) ER random network $N = 100, \langle k \rangle = 20$, (b) SF network $N = 100, \langle k \rangle = 20$ (c) ER network $N = 500, \langle k \rangle = 10$ (d) SF network $N = 500, \langle k \rangle = 10$

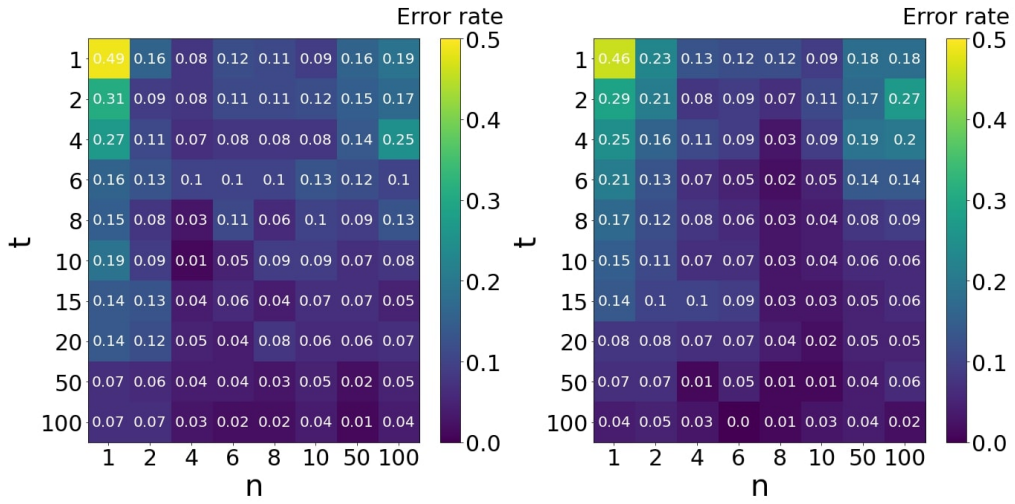

Figure 4.2: Classification error of ER and SF using actual time-series for $N = 100, \langle k \rangle = 20$ and (left) $\lambda = 0.085$ and (right) $\lambda = 0.09$

We then run the simulations using the 4^{th} order Runge-Kutta method for all the networks. For the networks of size $N = 100$, we store 100 θ values after an initial transient of 300, and for the networks of size $N = 500$, we store 200 θ values after an initial transient of 300. The reason for this is to speculate whether a longer length of time-series improves accuracy. We also store the symbolic time-series for the above networks. We take the time-series as input in decreasing order of the degree of the nodes. Thus, the first time-series provided corresponds to the highest degree node, and consecutive time-series belong to nodes with lesser degree. We then classify the networks of size $N = 100$ using our CNN machine learning model.

Fig. 4.2 clearly indicates the decrease in error with increasing the length of the time-series. However, the accuracy does not improve significantly with the increase in the number of nodes used for the classification. This proves that the length of time-series is far more important than the number of time-series provided. Thus, we require clearly, only the time-series of the top few highest degree nodes. Next, we classify the networks using the symbolic time-series for the networks of size $N = 100$.

Surprisingly, even with such coarse-grained data, the classification results are just as good as when the actual time-series were used. The trends of decrease in the error remains the same, i.e., the error decreases

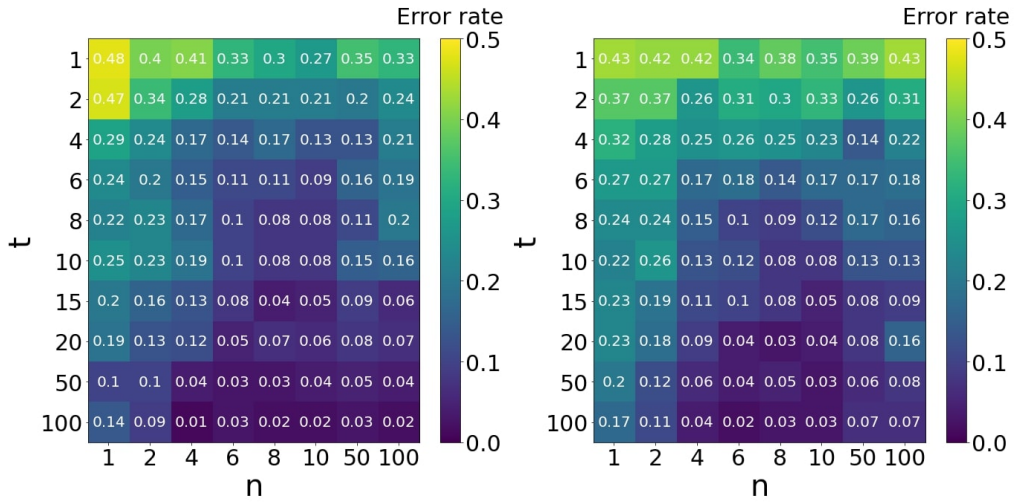

Figure 4.3: Classification error of ER and SF using symbolic time-series for $N = 100, \langle k \rangle = 20$ and (left) $\lambda = 0.085$ and (right) $\lambda = 0.09$

significantly with increase in the length of time-series used, but remains almost same with the increase in the number of nodes.

We learn from the above results that our model is capable of classifying the networks based on only a limited number of time-series of the network. To further investigate the effectiveness of our model, we classify the networks of size $N = 500$ for only the top 10 nodes using 200 length of time-series (Fig. 4.4)

4.2 Classification of ER, SF and SW networks

We have established that we only require a small number of time-series of the top 10 nodes to classify two types of networks, even for a larger size of the network. In this section, we extend our results to the case of three different types of networks, namely, ER, SF and SW networks. Thus, we take 250 realisations of each type, ER, SF and SW. For SW networks, we take the rewiring probability to be 0.1 and λ values as 0.28 and 0.3. Thus, we train the CNN model for 200 networks each of ER, SF and SW and test for 50 networks each. Fig. 4.5 shows the error rate of the classification. Clearly, the results are just as good as for the classification of two different

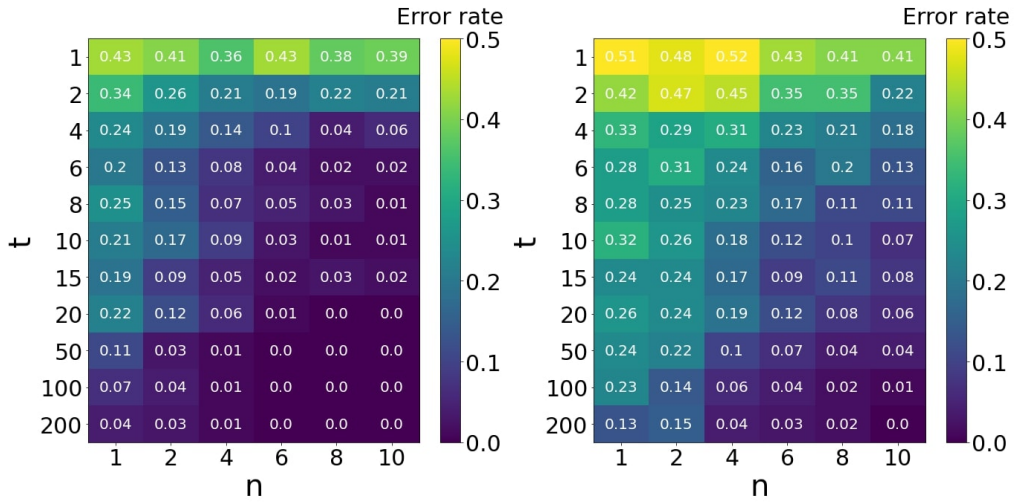

Figure 4.4: Classification error for ER and SF networks using symbolic time-series for $N = 500, \langle k \rangle = 10$, for (left) $\lambda = 0.15$ and (right) $\lambda = 0.17$

types. Thus, our model has successfully derived a clear correlation between the time-series and the structure of the networks based on observations of only a few highest degree nodes.

4.3 Classification of larger network

As we have seen, our results remained just as good for $N = 500$ as for $N = 100$, without any requirement for additional information of the number of nodes. We emphasise on this result by classifying ER and SF networks of size $N = 1000, \langle k \rangle = 10$.

We can see from Fig. 4.6, the classification results for larger network size is remarkably just as good, without the need for increasing the number of nodes. Even for the same length of time-series, the error drops to nearly 0, when top 10 nodes time-series is provided. This result reinforces the fact that our model is able to capture the underlying structure of the network based solely on the time-series of the top few nodes without requiring additional information even as the size of the network increases significantly.

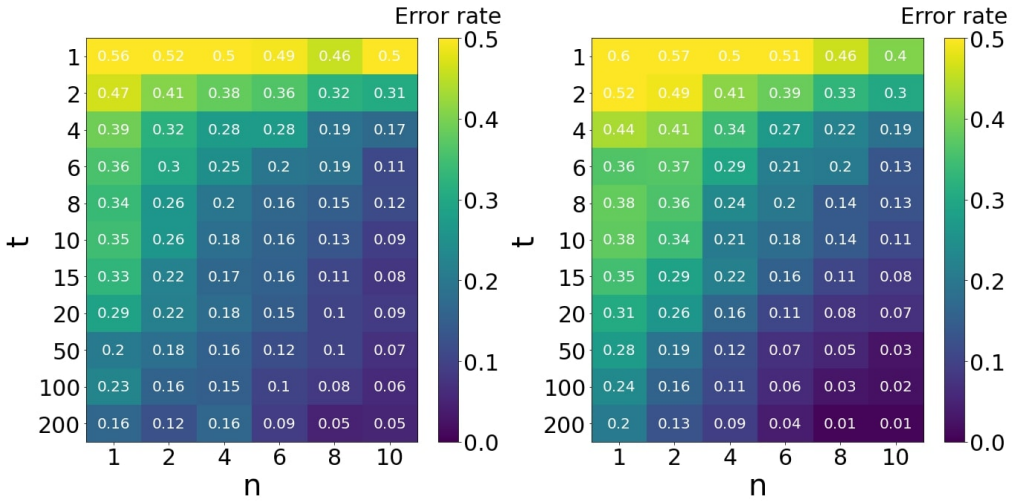

Figure 4.5: Classification error of ER, SF and SW networks using symbolic time-series for $N = 500, \langle k \rangle = 10$ and (left) $\lambda = 0.15$ for ER and SF, and 0.28 for SW, and (right) $\lambda = 0.17$ for ER and SF, and 0.3 for SW

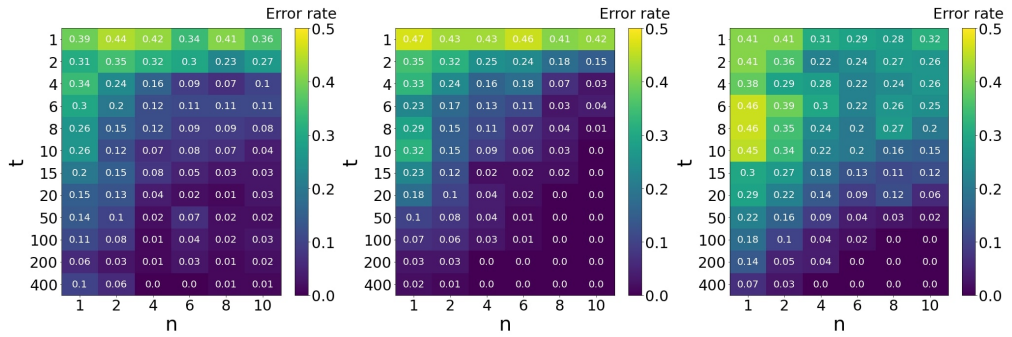

Figure 4.6: Classification of ER and SF networks using symbolic time-series of maximum length 400 for (left) $N = 100, \langle k \rangle = 20, \lambda = 0.085$, (middle) $N = 500, \langle k \rangle = 10, \lambda = 0.15$ and (right) $N = 1000, \langle k \rangle = 10, \lambda = 0.18$

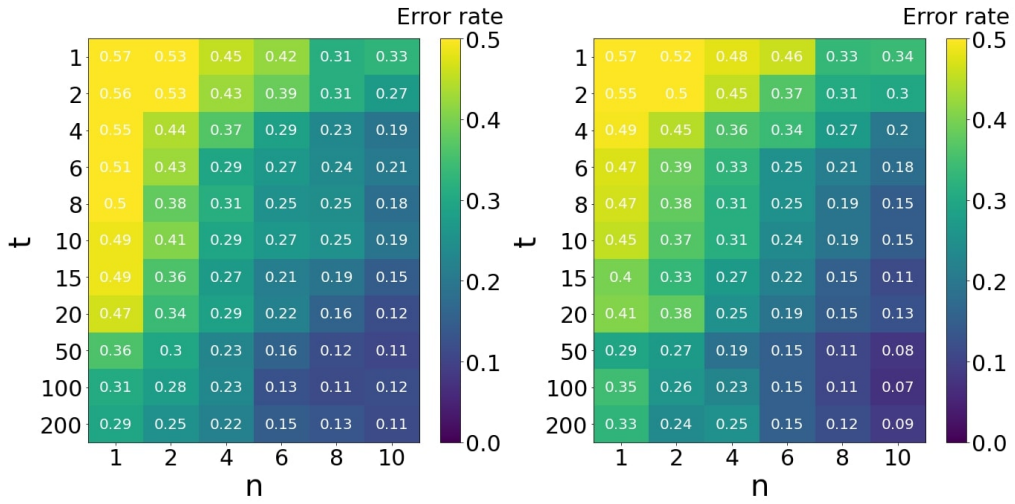

Figure 4.7: Classification error for SW networks generated for three different values of rewiring probabilities, $p_r = 0.05, p_r = 0.1$, and $p_r = 1$, and $N = 500, \langle k \rangle = 10$. (left) $\lambda = 0.4$ for $p_r = 0.05$, 0.28 for $p_r = 0.1$ and 0.15 for $p_r = 1$, (right) $\lambda = 0.45$ for $p_r = 0.05$, 0.3 for $p_r = 0.1$ and 0.17 for $p_r = 1$

4.4 Classification of SW networks with different rewiring probabilities

We now classify three SW networks with different rewiring probabilities, $p_r = 0.05$, 0.1 and 1. As rewiring probability increases, the randomness in the networks increase. Thus, a completely regular network is observed at $p_r = 0$ and a completely random network at $p_r = 1$, with a small-world transition at $p_r = 0.1$. Thus, in this section we test our model for the classification between regular, small-world and random networks. In a similar manner as previously, we generate 250 networks of each type and train the model. The length of time-series remains same. The size of the networks is $N = 500$, with average degree $\langle k \rangle = 10$. The maximum number of nodes taken for the classification is again 10, having the highest degrees.

The results in Fig. 4.7 show slightly poorer error rate. This can be attributed to the fact that the SW networks with different p_r values have the same shape of their degree distributions, and hence are harder to distinguish. The model, as we have evidently observed, learns the underlying

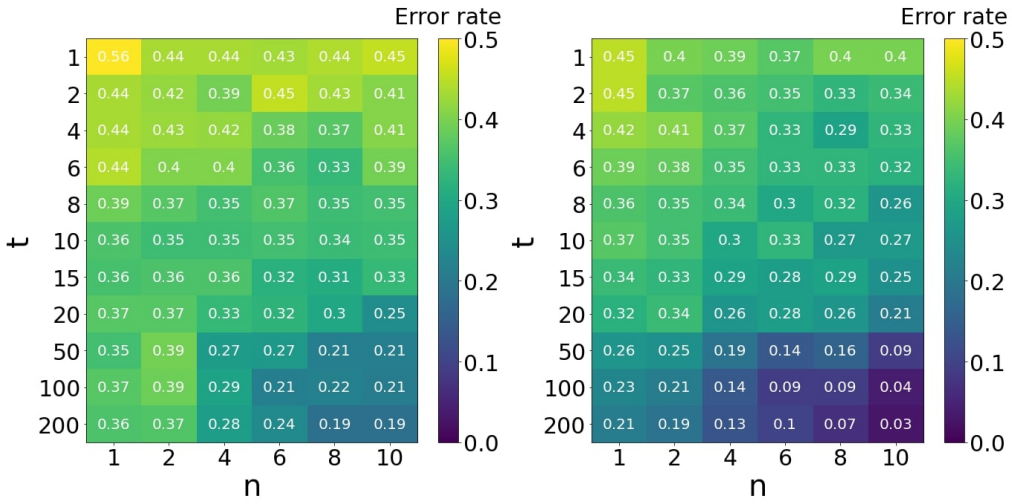

Figure 4.8: (left) Classification of ER networks of size $N = 500$ and average degrees $\langle k \rangle = 10$ and 50, using symbolic time-series, (right) Classification of ER and SF networks using symbolic time-series for $N = 500, \langle k \rangle = 10$, for randomly selected 10 nodes

structure from the time-series of the highest degree nodes, and makes the classification based on the structural differences. Thus, a poorer error can be expected for networks having their underlying structure similar across the different types. We substantiate this speculation by classifying ER networks with different average degrees in the next section.

4.5 Classification of ER networks with different degrees

In this section, we provide results (Fig. 4.8 (left)) for the classification of ER networks of size $N = 500$, and two different averages degrees $\langle k \rangle = 10$ and 50. Thus, we test our model on networks with different densities. For this, we again take 250 networks for each average degree and train the model.

4.6 Classification using random nodes

We now speculate our model's accuracy when the time-series of randomly selected 10 nodes is given. We find the accuracy of our model to be slightly worse, as the highest degree nodes contain richer information of the underlying structure of the network compared to randomly selected nodes. The networks used for this purpose are ER and SF of size $N = 500, \langle k \rangle = 10$. The results can be seen in Fig. 4.8 (right).

4.7 Classification of ER and SF for same range of r

Our results so far have required time-series data of networks for a constant value of λ for each network. However, for different classes of networks, for example ER and SF, the amount of synchronization can be quite different for the same value of λ. One might question the fact that our model learns the underlying structure, independent of the differences in the level of synchronization in the networks. Thus, to argue the fact that our model indeed learns the difference in structure of the networks and does not depend on the difference in the r values, we generate two sets of data for ER and SF networks, this time taking the same range of r values for both classes. We set the values of r to be between 0.45 and 0.55, and generate the time-series for the networks for this range of r. We again take 250 networks of each type and 200 length of the time-series. This way, we ensure that every network in our input data set has the same level of synchronization.

Fig. 4.9 nicely proves our hypothesis. Even for the same level of synchronization in the networks, our model successfully classifies the different types of networks solely based on the structural differences, with no additional information required.

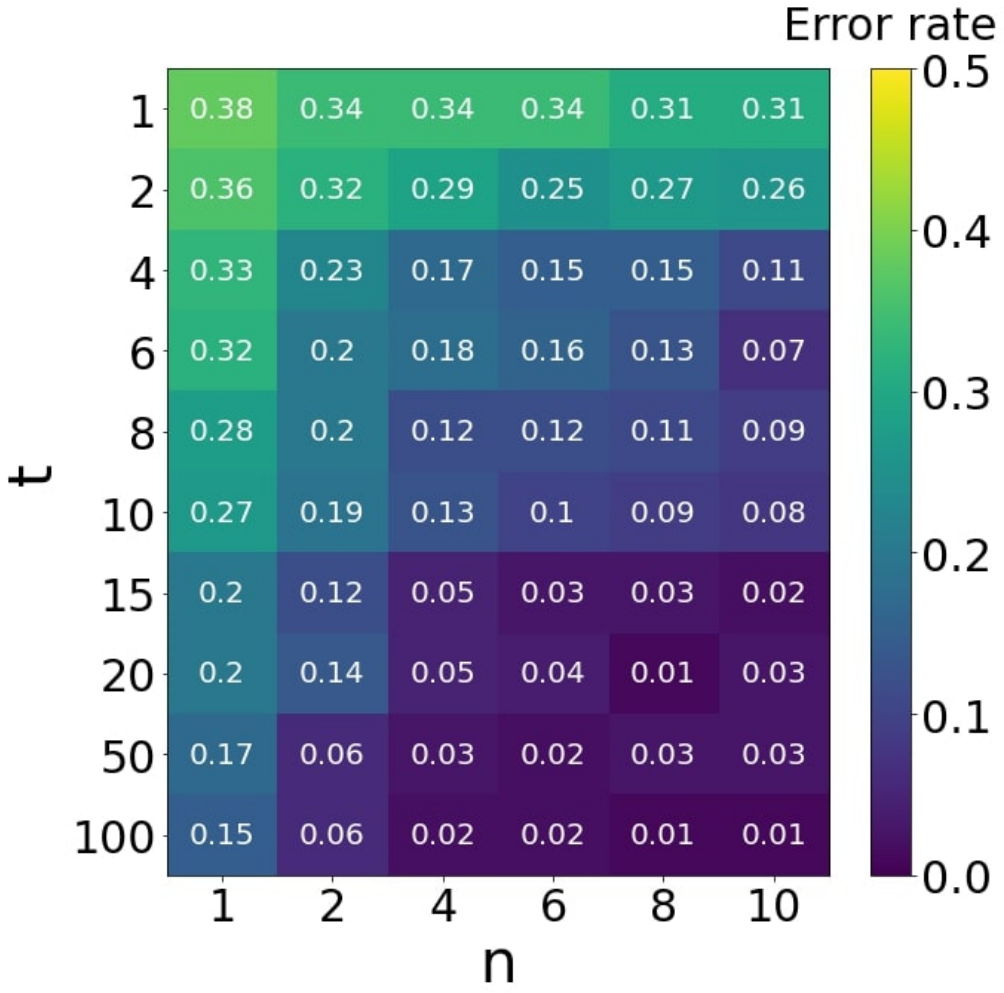

Figure 4.9: Classification of ER and SF networks having the same range of r values for $N = 500, \langle k \rangle = 10$, using symbolic time-series. λ varies for each network.

Chapter 5

Conclusion and discussions

We have shown, with high accuracy, that our machine learning model classified different types of networks by learning the underlying structure from the dynamics of a few highest degree nodes, based solely on the structural differences in the networks. Our model was able to distinguish between different types of network not only based on different degree distributions, but also based on other structural properties such as different average degrees and different degrees of randomness in the structure. The fact that the size of the networks is irrelevant to the accuracy of our model, is remarkable in itself and goes to show that a few high degree nodes contain sufficient information about the underlying network structure in their time-series data. In addition to this, we have shown that simply the trends in the change of the phase values in the time-series is enough information to trace back the original network structure. Using only the symbolic time-series, our model made classifications with equal accuracies as for when the actual time-series were provided. The symbolic time-series provides significant advantage over actual time-series. The training time for the machine learning model is greatly reduced in using the binary information of the symbolic time-series as opposed to using the actual time-series. In addition, the symbolic time-series is a more applicable form of input data in the real world, where collection of time-series introduces a lot of noise in the data, making the classification tasks prone to errors due to the noise in the input. Eliminating this by taking only the difference in the phase values of

the nodes and converting them to a symbolic form, greatly reduces our model's susceptibility to errors in the data collected. From a theoretical standpoint, successful classification of networks from symbolic time-series signifies the richness of structural information contained in the dynamics of the networks. Extending our results to time-series generated from same range of r values have discarded any doubts regarding the suspicion that our model classified the networks based on the differences in the levels of synchronization.

As we have shown, unquestionably, that our model can classify different types of networks based solely on the difference in their connectivity structures, we have proven the massive success of machine learning in the study of complex systems. Network science, as an evolving field in almost all areas of science, has been an excellent abstraction of data, capturing the pairwise interactions of real world entities and providing immense dynamical information. Having machine learning techniques successfully apprehend the underlying correlations of the network structure and its dynamics not only provides a great foundation of further research in the network science space, but also in other areas of scientific research.

Chapter 6

Future work

In view of the immense success of our machine learning model in classifying the networks based on limited number of time-series of the highest degree nodes, we explore further the applicability of machine learning in predicting the correlation matrices of the time-series of chaotic oscillators. For this purpose, we model ER, SF and SW networks on the Rössler oscillators (eq. 1.2). We generate the time-series of these oscillators in a similar manner and provide the time-series in the x-dimension of the top 5 highest degree nodes to our new supervised machine learning model. We train the model on a few networks with varying coupling constants and network structures by supplying the entire correlation matrices as output labels. With only a few networks provided, our model is able to reconstruct the entire correlation matrix with high accuracy. We intend to further investigate these results by testing our model against various parameters such as degree, rewiring probabilities, size of the networks, and so on.

Bibliography

[1] J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, E. Ott, Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data, Chaos 27 (2017) 121102.

[2] S. Saha, A. Mishra, S. Ghosh, S. K. Dana, C. Hens, Predicting bursting in a complete graph of mixed population through reservoir computing, Phys. Rev. Res. 2 (2020) 033338.

[3] Q. Ni, M. Tang, Y. Liu, Y. C. Lai, Machine learning dynamical phase transitions in complex networks, Phys. Rev. E 100 (2019) 052312.

[4] M. Ganaie, S. Ghosh, N. Mendola, M. Tanveer, S. Jalan, Identification of chimera using machine learning, Chaos 30 (2020) 063128.

[5] G. Neofotistos, M. Mattheakis, D. Georgios, J. Barmparis, Hizanidis, G. P.Tsironis, E.Kaxiras, Machine learning with observers predicts complex spatiotemporal behavior, Front. Phys. 7 (2019) 24. doi:10.3389/fphy.2019.00024.
URL https://www.frontiersin.org/article/10.3389/fphy.2019.00024

[6] J. Carrasquilla, R. G. Melko, Machine learning phases of matter, Nature Physics 13 (5) (2017) 431.

[7] T. Ohtsuki, T. Ohtsuki, Deep learning the quantum phase transitions in random two-dimensional electron systems, Journal of the Physical Society of Japan 85 (12) (2016) 123706.

[8] T. Ohtsuki, T. Ohtsuki, Deep learning the quantum phase transitions in random electron systems: Applications to three dimensions, Journal of the Physical Society of Japan 86 (4) (2017) 044708.

[9] N. Boullé, V. Dallas, Y. Nakatsukasa, D. Samaddar, Classification of chaotic time series with deep learning, Physica D: Nonlinear Phenomena 403 (2020) 132261.

[10] F. Rodrigues, T. Peron, J. Kurths, A machine learning approach to predicting dynamical observables from network structure, arXiv:1910.00544.

[11] E. Bianco-Martinez, N. Rubido, C. G. Antonopoulos, M. S. Baptista, Successful network inference from time-series data using mutual information rate, Chaos 26 (2016) 043102.

[12] G. Tirabassi, R. Sevilla-Escoboza, J. M. Buldú, C. Masoller, Inferring the connectivity of coupled oscillators from time-series statistical similarity analysis, Sci. Rep. 05 (2015) 10829.

[13] P. R. da Silva Soares, R. B. C. Prudêncio, Time series based link prediction, in: The 2012 International Joint Conference on Neural Networks (IJCNN), 2012, p. 1.

[14] A. Banerjee, J. D. Hart, R. Roy, E. Ott, Link inference of noisy delay-coupled networks: Machine learning and opto-electronic experimental tests, arXiv:2010.15289.

[15] M. Breakspear, S. Heitmann, A. Daffertshofer, Generative models of cortical oscillations: neurobiological implications of the kuramoto model, Frontiers in Human Neuroscience 4 (2010) 190.

[16] R. Schmidt, K. J. R. LaFleur, M. A. de Reus, L. H. van den Berg, M. P. van den Heuvel, Kuramoto model simulation of neural hubs and dynamic synchrony in the human cerebral connectome, BMC Neurosci. 16 (2015) 54.

[17] A. H. N. G. Filatrella, N. F. Pedersen, Analysis of a power grid using a kuramoto-like model, Phys. Cond. Matt. 61 (2008) 485.

[18] C. S. Peskin, Mathematical Aspects of Heart Physiology, Courant Institute of Mathematical Science Publication, New York, 1975.

[19] J. Buck, Synchronous rhythmic flashing of fireflies. ii., Quart. Rev. Biol. 63 (1988) 265.

[20] T. J. Walker, Acoustic synchrony: two mechanisms in the snowy tree cricket, Science 166 (1969) 891.

[21] Z. Jiang, M. McCall, Numerical simulation of a large number of coupled lasers, J. Opt. Soc. Am. 10 (1993) 155.

[22] K. Wiesenfeld, P. Colet, S. H. Strogatz, Synchronization transitions in a disordered josephson series array, Phys. Rev. Lett. 76 (1996) 404.