

# **A GENERAL FRAMEWORK FOR EVOLVING NETWORK ANALYSIS**

A Thesis

by

**BIN WU**

Submitted to the Office of Graduate and Professional Studies of  
Texas A&M University  
in partial fulfillment of the requirements for the degree of

**MASTER OF SCIENCE**

Chair of Committee,	Chii-Der S. Suh
Committee Members,	Srikanth Saripalli
	Xingyong Song
Head of Department,	Andreas A. Polycarpou

December 2018

Major Subject: Mechanical Engineering

Copyright 2018 Bin Wu

## **ABSTRACT**

Many efforts have been given to model physical systems involving a large number of interacting constituents. Such systems are commonly called complex networks for their complex behaviors demonstrated at the system, global level. As a network evolves its constituents (or nodes) and associated links would either increase or decrease or both. It is a challenge to extract the specifics that underlie the evolution of a network or indicate the addition and/or removal of links in time. Many evolving network analysis algorithms are available. However, the majority of these algorithms assume network evolution can be fully comprehended using a snapshot of the network as it progresses. Without retaining the time information would inevitably obscure the dynamics of the evolving network and misinterpret its behaviors. A general framework viable for describing evolving networks is developed in the thesis. The framework incorporates the two-dimensional discrete wavelet transform (2DWT) with the tensor factorization method to extract features indicative of the development of a network in the time-frequency domain. The general framework model is evaluated against six benchmark algorithms using five different real-world evolving network datasets to demonstrate its feasibility in achieving a high level of link prediction.

To my family, friends, and mentors.

## **CONTRIBUTORS AND FUNDING SOURCES**

### **Contributors**

This work was supervised by a thesis committee consisting of Professor C. Steve Suh and Professor Srikanth Saripalli of the Department of Mechanical Engineering and Professor Xingyong Song of the Department of Engineering Technology & Industrial Distribution.

All work for the thesis (or) dissertation was completed independently by the student.

### **Funding Sources**

There are no outside funding contributions to acknowledge related to the research and compilation of this document.

# TABLE OF CONTENTS

	Page
ABSTRACT.....	ii
CONTRIBUTORS AND FUNDING SOURCES .....	iv
TABLE OF CONTENTS.....	v
LIST OF FIGURES .....	vi
LIST OF TABLES .....	viii
1 INTRODUCTION .....	1
1.1 Network Analysis.....	1
1.2 Social Network Analysis.....	3
1.3 Research Need .....	7
2 METHODOLOGIES .....	11
2.1 Problem Statement.....	11
2.2 Wavelet Time-Frequency Analysis Tool.....	11
2.3 Temporal Model in the Wavelet Domain .....	25
2.4 Wavelet Coefficient Temporal Model (WCTM) for Evolving Network Analysis....	28
3 BENCHMARK DATASETS AND LINK PREDICTION ALGORITHMS .....	33
3.1 Dataset Description.....	33
3.2 SNA Methods.....	37
3.3 Evaluation Methods .....	39
4 EXPERIMENT AND DISCUSSION .....	42
5 CONCLUSION AND FUTURE WORK .....	48
5.1 Conclusion .....	48
5.2 Future Work .....	49
REFERENCES .....	51

## LIST OF FIGURES

	Page
Figure 1      Visualization of Infectious Network at Time $T = 8h$ .....	3
Figure 2      Link Prediction .....	4
Figure 3      Community Detection .....	5
Figure 4      Anomaly Detection.....	7
Figure 5      Algorithms for Network Analysis .....	9
Figure 6      Visualization of Evolving Network.....	10
<i>Figure 7      Grayscale Image of an Adjacency Matrix.....</i>	<i>13</i>
Figure 8      Grayscale Image of the Infection Network at $T = 0-1h$ .....	14
Figure 9      Infection Network in Time Decay Grayscale .....	16
Figure 10      Two Types of Networks .....	17
Figure 11      Original Image.....	19
Figure 12      2DWT of Original Image .....	19
Figure 13      2 Dimensional Separable Wavelet Transform.....	20
Figure 14      Decomposition Wavelet Filterbanks .....	22
Figure 15      Synthesis Wavelet Filterbanks .....	23
Figure 16      Multi-Level 2 Dimensional Wavelet Decomposition.....	23
Figure 17      Time-Decay Image of the Infectious Network at $T = 0-2h$ .....	24

	Page
Figure 18	Single level Decomposition of the Infectious Network at $T = 2$ .....24
Figure 19	Network Image .....30
Figure 20	2DWT of the Network Image in Figure 19 .....30
Figure 21	Network Image with More Details .....31
Figure 22	2DWT of the Network Image in Figure 21 .....31
Figure 23	General Framework for Evolving Network Analysis .....32
Figure 24	First 20 Rows of a Static Dataset .....34
Figure 25	First 20 Rows of an Evolving Dataset .....34
Figure 26	Links for Training.....41
Figure 27	Observed Links with Test Links.....41
Figure 28	AUC Score for All Algorithms in Infectious Network .....43
Figure 29	AUC Score for All Algorithms in Haggie Network .....44
Figure 30	AUC Score for All Algorithms in arXiv Network .....44
Figure 31	AUC Score for All Algorithms in DNC Network .....45
Figure 32	AUC Score for All Algorithms in UCI Message Network.....45

## LIST OF TABLES

	Page
Table 1     Details of Experimental Datasets.....	36
Table 2     Model Parameters and Experimental Setting.....	43
Table 3     Average AUC Value for All the Algorithms .....	47

# 1 INTRODUCTION

In this chapter, the need for studying networks is given, followed by considering three areas of network analysis to help identify the various issues that hamper the broader application of the current approaches. The objective of the thesis, which is to generate a general framework valid for studying evolving networks, is provided in the last section.

## 1.1 Network Analysis

The concept of social networks was first explored by John. A. Barnes [1] in a work published in 1954 where a network was used to describe human relations. A social network is a system consisting of many nodes (vertices) and links (edges). Figure 1 is a social network that depicts the face-to-face contacts of the participants in INFECTIOUS: STAY AWAY - an exhibition at the science gallery in Dublin in 2009 [2]. In this network, the nodes represent the exhibition visitors and the edges represent face-to-face contacts that were active for at least 20 seconds.

Different from the physical data of natural science, social networks data are as complex as they are intractable. People and the inter-relationships of theirs are the fundamental elements of a social network. Social network data are of two types [3]:

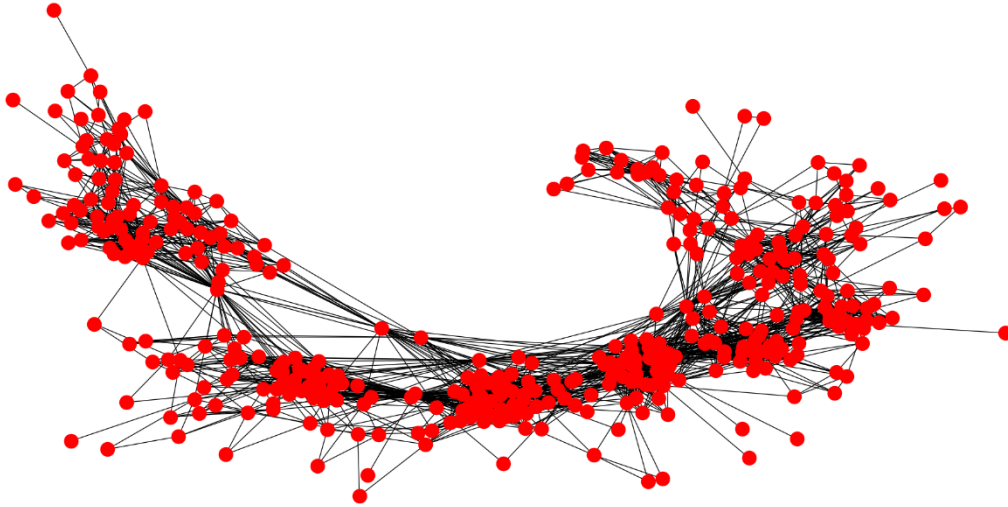
- Attribute data: The behavior of each node (or each individual) would eventually influence the behavior of the whole network. Attribute data are data that describe the property of the individuals in the network such as attitude, user opinion, or any information that can be

used to define the individual node in the network. Attribute data can be collected through questionnaire or survey.

- Relation data: Information about individual nodes can be collected from the attribute data.

However, this is not sufficient to form and define a network. Relation data are the contacts, ties, and/or connections of the nodes. Different from the attribute data, relation data focus on describing the interaction information among the nodes. For example, in Figure 1, when two persons met with each other more than 20 seconds then at the moment an edge was formed to connect the two individuals (nodes) in the network. Relation data convey when edges were generated involving which nodes.

Per the two types of network data, it can be concluded that social network analysis studies network individuals (nodes) and their relationships (edges) to have a better understanding of the network. Given that individual nodes are each rich of different properties, finding a rule for their relationship that is generally applicable to all of them is not an easy task. However, a comprehensive understanding of the network can be established by exploring the relation data of these nodes. Network analysis therefore aims at studying the relationship between nodes.



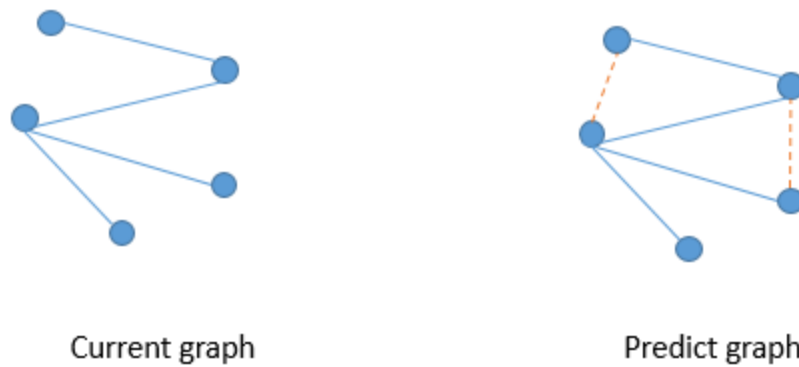
**Figure 1: Visualization of Infectious Network at Time  $T = 8h$**

## **1.2 Social Network Analysis**

Link prediction, community detection, and anomaly detection are the three areas of interest in social network analysis. Because network analysis studies the nodal relationship in a network, it is imperative that relationships are defined.

Link prediction is for predicting whether new links and new edges would appear or disappear in the future. Link prediction can be used to predict the encountering or establishing of the relationship between two nodes at the future time. As graphically conveyed in Figure 2, two new links are added (predicted) in the simple 5-node network. Lately, much interest has been given to link prediction with many algorithms developed to predict new links in social networks [4][5][6].

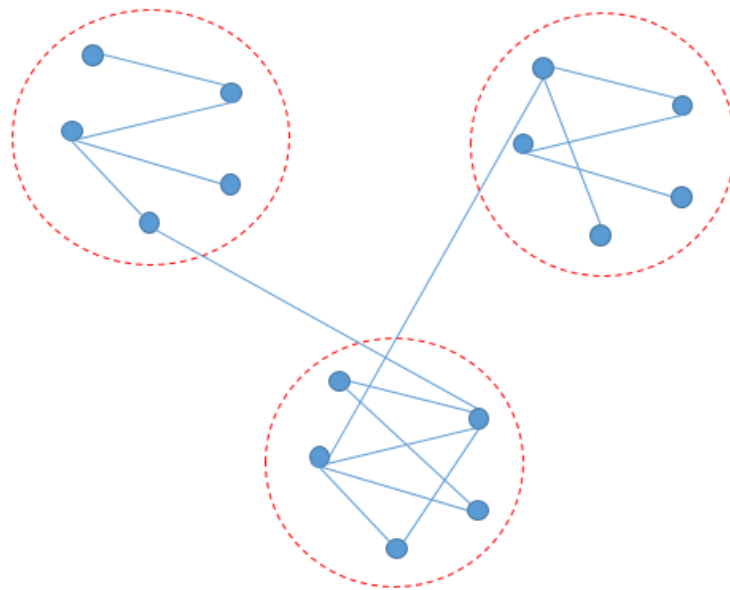
Similarity-based algorithms [7][8][9], likelihood methods [10][11][12], and probabilistic models [13][14][15] are 3 mainstream methods for link prediction.



**Figure 2: Link Prediction**

Social networks are networks of individuals connected by interpersonal relationships. A community is a feature of a social network characterized by abundant contacts, as depicted in Figure 3. Being able to detect communities in a social network promises many applications. Take Amazon the online shopping site as an example. If consumers sharing similar shopping interests can be detected, then the community of these consumers can be targeted with products determined by their purchase relationship. Many community detection algorithms have been developed [16]. Graph partition [17] divides the edges in a group of predefined size. The

algorithm by Girvan and Newman calculates the centrality of the vertices [18]. Dynamic algorithms focus on spin-spin interaction, random walks, and synchronization [19].

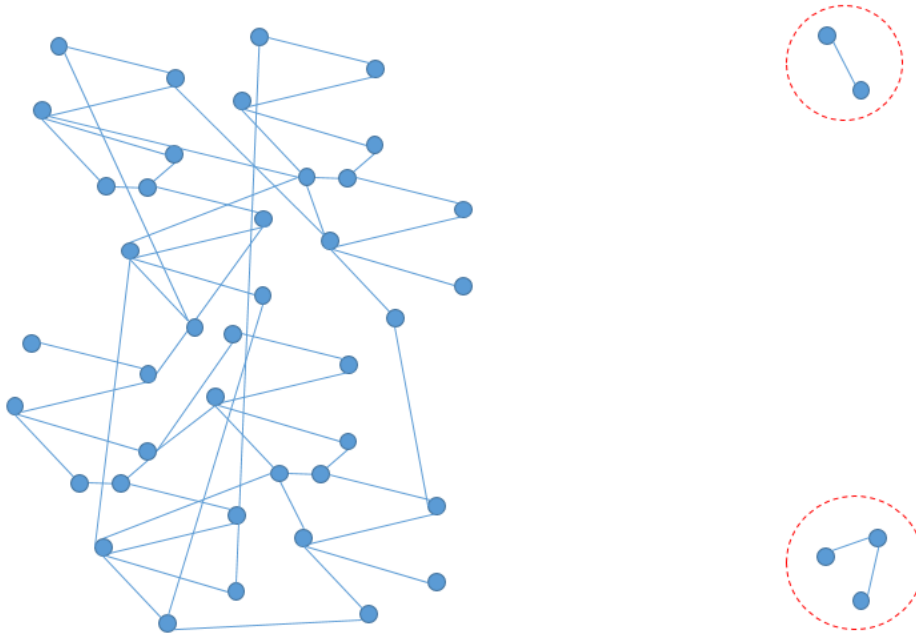


**Figure 3: Community Detection**

Unlike link prediction and community detection, anomaly detection focuses on finding patterns that do not conform to the expected behaviors as indicated by the 2 out-of-the-ordinary anomalies seen in Figure 4. Anomalies can be outliers, surprised, peculiarities, or exceptions. Anomaly detection extracts the nonconforming patterns of the anomalies. The nonconforming nodes usually are expected to appear in a network. If these nodes can be detected and be removed

from a network, a better understanding of the characteristics of the network becomes probable. Anomaly detection has been applied to identify anomalous traffic information in a computer network, spiteful credit card user register, and terrorists. The process of anomaly detection is straightforward. First is to establish the normal behavior pattern and then detect the behavior that does not conform to the normal pattern. Many algorithms have been developed [20] including classification based anomaly detection techniques [21], nearest neighbor-based anomaly detection techniques [22], clustering-based anomaly detection techniques [23], and statistical anomaly detection techniques [24], among many others.

The above three are common in social network analysis. As the summary given in Figure 5 suggests, they handle the relationships of the nodes in the network with applications in different situations.



**Figure 4: Anomaly Detection**

### 1.3 Research Need

The objective of the MS thesis is to develop a general approach viable for investigating evolving networks. Many network analysis approaches and algorithms are either reviewed or evaluated in references [20], [25], and [26]. However, are they feasible for addressing network problems? To answer the question, one needs to re-examine as to what evolving networks are. The graph in Figure 1 conveys a network. However, is it representative of the networks found in the real-world? And, does a snapshot carry all the information of the network at all time? The answers are both negative.

In the real-world, a social network evolves in time because the attributes (peoples) of the network are not stationary nor static. Figure 1 is a snapshot of the infectious network at the end of the meeting. (The infectious dataset records visitors meeting over a 7.8-hour time span.) From the snapshot, we only know connections were generated but not the time information as to when they were generated. In the real world, networks evolve. The time at which a link is established is an essential part of the network. Figure 6 gives the evolution of a 5-node network over a time window of 6 hours. Using a time-step of one hour, for example, there would be six snapshots in total to represent the evolving network. And, smaller the time step, more information would be obtained and larger the dataset size would be. Knowing at which hour visitors met each other is essential.

The differences between evolving network analysis and static network analysis are as follows:

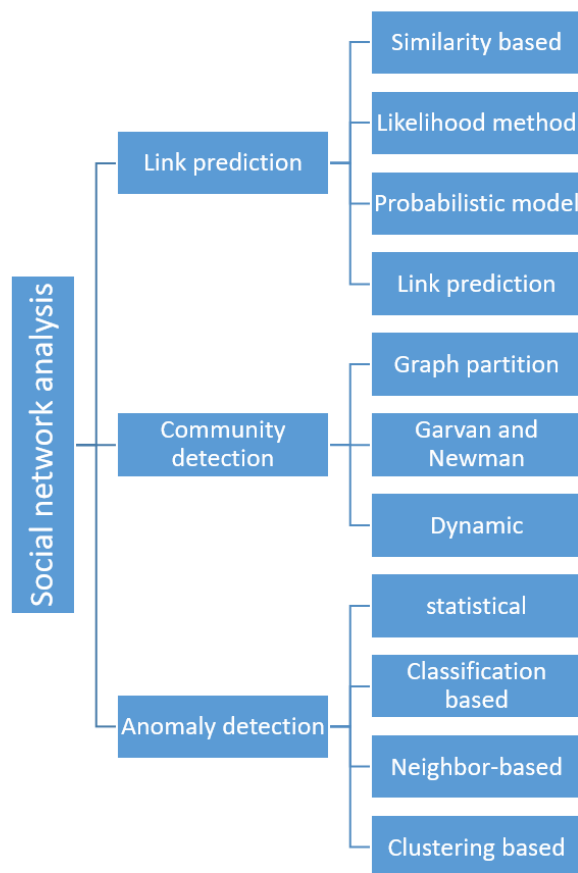
- Static network analysis (SNA) analyzes a network based on a snapshot of the network where the time information is missing.
- Evolving network analysis (ENA) uses appropriate approaches to extract the evolving features of a network. Differ from SNA which uses a snapshot, ENA employs time series data which retain time information about the network.

ENA is more complicated than SNA because the dataset size is larger and the processing time is longer. ENA provides many benefits,

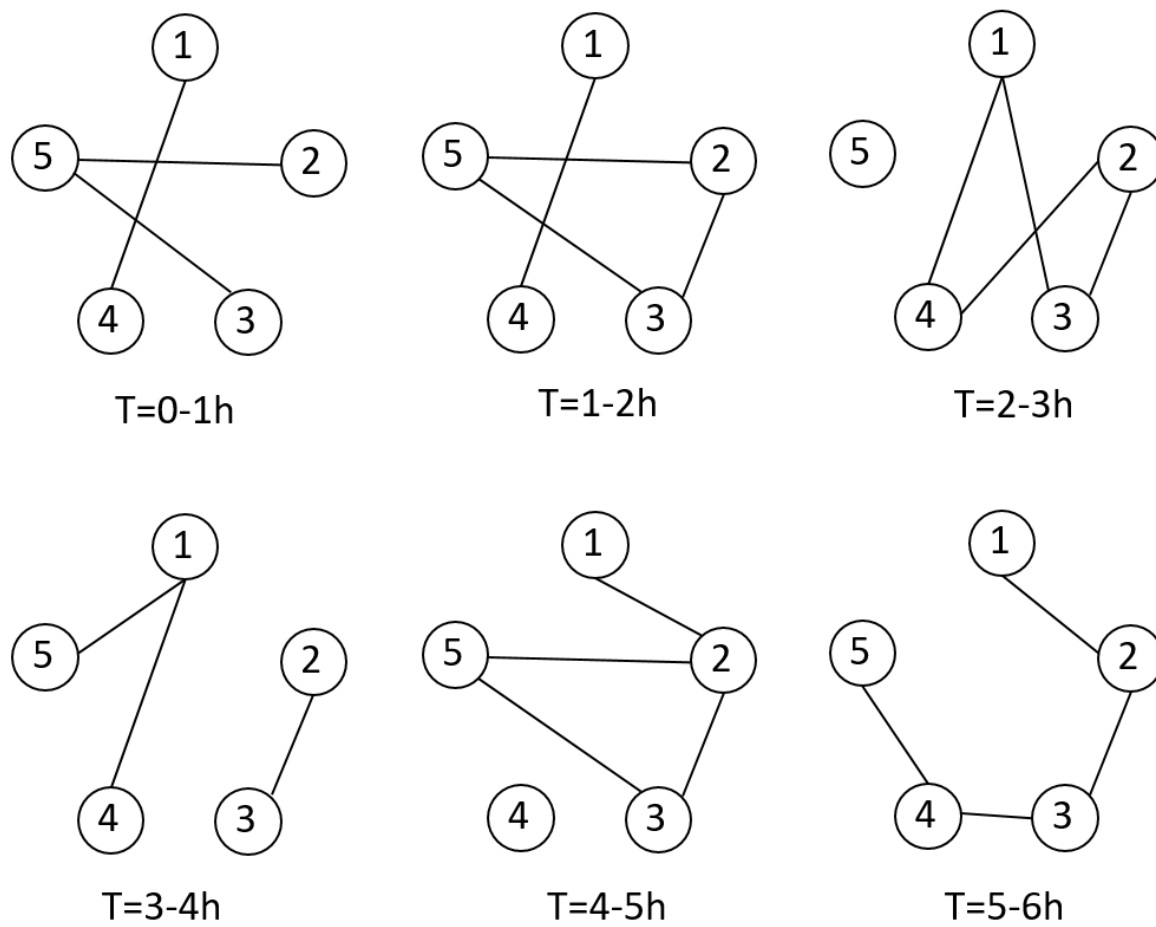
- Information processed by ENA pertaining to the state of an evolving network in time is intact.

- ENA retains and does not obscure the inherent characteristics of the network. As the network evolves in time, its evolution pattern is to be extracted from the corresponding time-series data. Not from a particular snapshot.

ENA can extract more information from a network than SNA. Unfortunately, the literature review shows that the majority of algorithms developed for network analysis are SNAs. The objective of this M.S. research is to develop a general framework for ENA.



**Figure 5: Algorithms for Network Analysis**



**Figure 6: Visualization of Evolving Network**

## 2 METHODOLOGIES

### 2.1 Problem Statement

Link prediction, which is a primary area of interest in ENA, is the focus of the thesis. In this chapter link prediction is first introduced as an SNA problem, followed by generalizing it into an ENA link prediction problem.

Given a network  $G(V, E)$ , where  $V$  is the set of nodes and  $E$  is the set of links. For simplicity, only undirected networks are considered. The universal set containing all  $\frac{|V| \cdot (|V| - 1)}{2}$  of the possible links is denoted as  $U$ , where  $|V|$  is the number of elements in set  $V$ . The nonexistent link set is  $U - E$ . In real-world networks, links may be lost and new links may appear in the future. Link prediction is to determine whether these links would appear in a future time. In the real world, networks are time-evolving which means  $V$  and  $E$  are also time-evolving, thus  $G$  is a time-varying network,  $G(V(t), E(t))$ . Link prediction in evolving networks can be defined as determining the missing and newly established links based on a collection of  $G(V(t), E(t))$  in time, where  $t \in [1, T]$

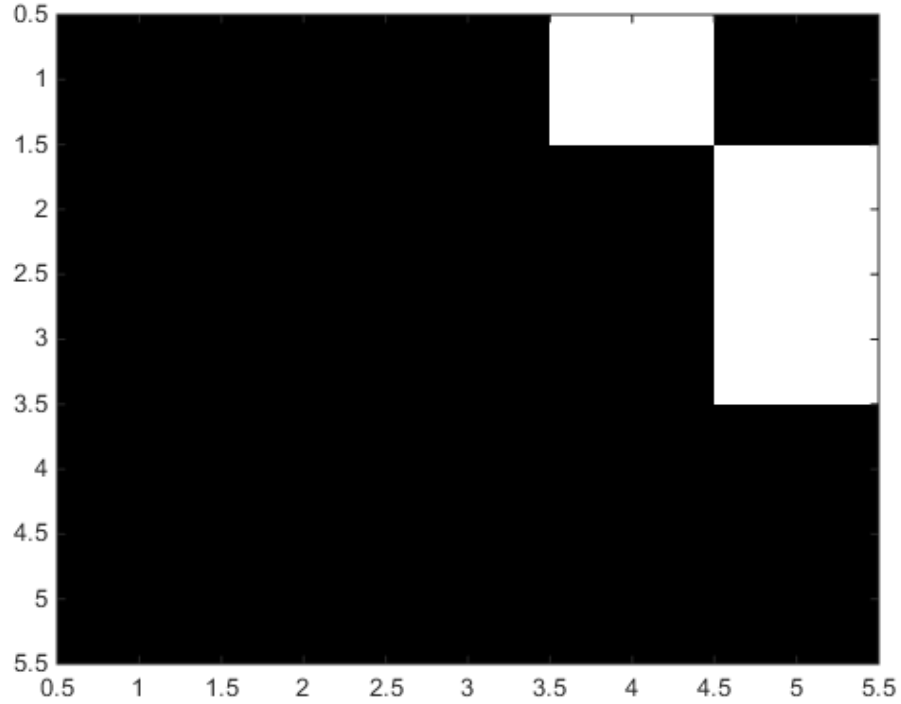
### 2.2 Wavelet Time-Frequency Analysis Tool

An adjacency matrix as defined in the graph theory is a symmetric matrix used to represent a finite graph. The element  $a_{ij}$  of the matrix indicate a connection from node  $j$  to node  $i$ . Whether a pair of vertices (nodes) are connected or not in the graph is explicitly implied by the corresponding matrix element using the scheme:  $a_{ij} = 0$  if there is a link between the  $i$ -th and  $j$ -th nodes. Otherwise  $a_{ij} = 1$ . A network at a specific time instance can be represented using an

adjacency matrix. For example, when  $T=0-1h$ , the network in Figure 6 has the following adjacency matrix representation:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

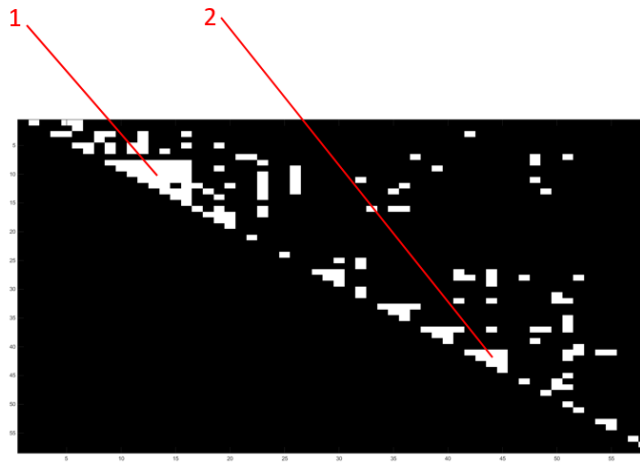
The row and column of the matrix correspond to the index of each node. Since only undirected networks are considered in the thesis, the corresponding adjacency matrix is therefore symmetrical about the diagonal. Only the entries in the upper triangular matrix need be stored and processed. A matrix can also be represented as a grayscale image. The undirected matrix above, which contains binary elements, can be alternatively represented using Figure 7. Note that the lower triangular portion of the square matrix has been set to ‘zero.’



**Figure 7: Grayscale Image of an Adjacency Matrix**

The grayscale image is a visualization of the matrix. Consider again the infectious network as an example. There were 401 nodes. A 401-by-401 adjacency matrix,  $A(t)$ , can be used to represent the network. Because the infectious network is an unweighted network, it can be represented using the adjacency matrix seen in Figure 8 with binary entries. At  $T = 0\text{-}1\text{h}$ , some elements in the image, such as clusters 1 and 2, generate more links than others. In link prediction, it is hard to recognize the pattern of the entire network. Therefore heuristic approaches are popular for predicting future links. Heuristic approaches are considered valid for static network

analysis. They are not proper approaches for extracting features from evolving networks. More discussions are given in Chapter 3.



**Figure 8: Grayscale Image of the Infection Network at T = 0-1h**

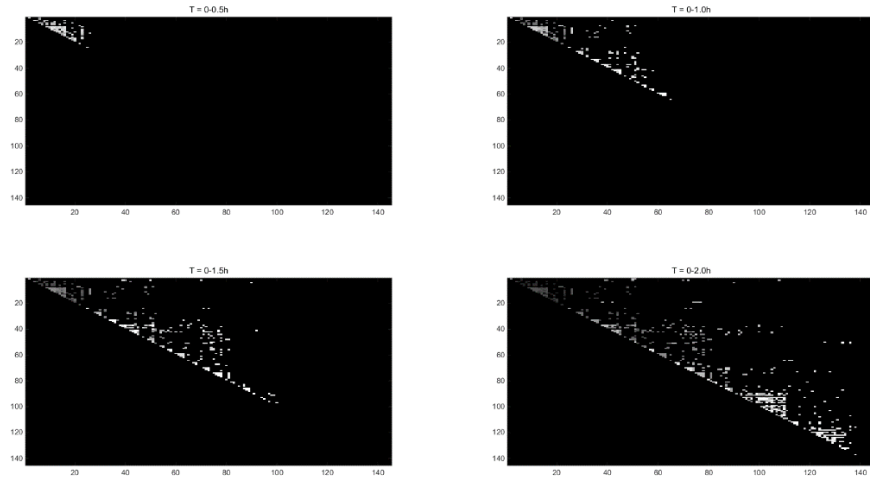
Figure 8 indicates that there are features hidden in the clusters where more links were generated. If these features can be extracted, they could be used to better predict link appearance and disappearance in future time. As the clusters are subsets of the entire image, salient information of the evolving infectious network can be extracted by processing all the individual subsets. Prediction is to foretell future using past data. For example, new links for the infectious network at  $T = 7-8h$  can be predicted based on the features extracted from the data collected at  $T = 0-7h$ . In other words, past data have an influence on future results. However, the past data at  $T_1 = 0-1h$  and  $T_2 = 6-7h$  do not impact the future result at  $T_3 = 7-8h$  the same

way. In this thesis, it is assumed that the data collected closer to a future time at which a prediction is attempted to have a greater influence.

The adjacency matrix for an undirected network is a binary matrix. When two nodes are connected, the corresponding element  $a_{ij}$  is set to 1. If not,  $a_{ij}$  is set to 0. The  $a_{ij}$  for each pair of nodes in the adjacency matrix are the scores of the pairs indicating their connection. Larger  $a_{ij}$  indicates a higher probability that the node pair will connect in the future. The following exponential decay function is feasible for describing the decreasing impact of a past incident on the future event:

$$D(t) = e^{-\theta(t-T)} \quad (1)$$

where  $\theta > 0$  is a tuning parameter which is network dependent,  $T$  is the final time, and  $t$  is the time when the data is collected. The decay function is incorporated into the infectious network matrix to generate the corresponding grayscale images seen in Figure 9 that correspond to  $T = 0-0.5h$ ,  $T = 0-1h$ ,  $T = 0-1.5h$ , and  $T = 0-2h$ . Note that the parameters employed are  $\theta=0.1$  and  $T = 2$  using a 0.1h time step.



**Figure 9: Infection Network in Time Decay Grayscale**

That salient information pertaining to the attributes of an evolving network can be extracted from individual subsets and that past incidents have an exponentially decayed impact on future events are the two bases for resolving ENA problems using the following steps:

Step 1: Decompose the network graph image into subsets without loss of information or aliasing

Step 2: Extract salient features from each image subset

Step 3: Generate a new subset using the extracted features

Step 4: Assemble processed subsets to present a prediction for the future event

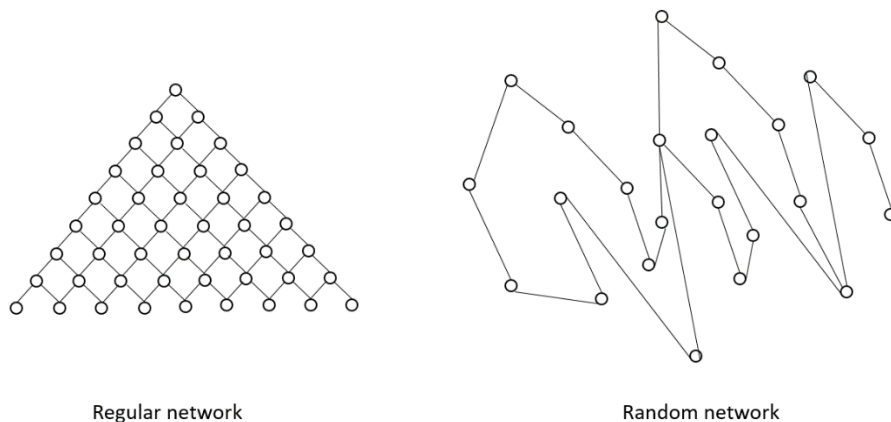
To implement the above, it is essential that a proper tool for partitioning network graph images be employed, one that ensures that

- Decomposition of the network graph is lossless (Step 1)
- Reconstruction of the network graph using processed subsets is lossless (Step 4)

How to decompose a network graph image depends on the characteristics of the network.

In graph theory, there are two types of networks: regular networks (regular lattices and trees) and random networks (random graph), as seen in Figure 10.

- Regular networks usually are man-made networks of low heterogeneity and low randomness. They tend to have long average paths and high clustering.
- Random networks (RN) are generated by the random process where graph theory intersects with probability theory. A random graph can be generated by setting a series of nodes and then add edges between the nodes randomly. Because randomness is common in the real world, there are many random graph models [27][28][29]. Different random graph models result in the very different probability distribution of the graph.



**Figure 10: Two Types of Networks**

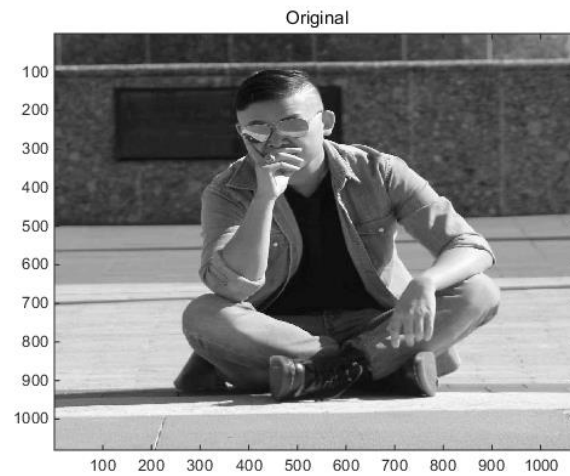
Regular networks and random networks are extreme cases. Real-world networks usually fall somewhere in between. Watts and Strogatz proposed a small world network model [30] in 1998. Barabasi and Albert proposed a scale-free network model [31]. Real-world networks come with a variety of different types than can be described by the small-world or scale-free world models. The infectious network shown in Figure 8, for example, is a network positioned somewhere between the random and regular networks, neither a small-world nor a scale-free network.

Networks that are in between random and regular networks are partial pattern networks (PPNs). Being ‘partial pattern’ means the network obey certain rules. However, differing from regular networks, the ‘pattern’ is temporal and easily influenced by initial conditions, making it hard to find a precise mathematical representation for it.

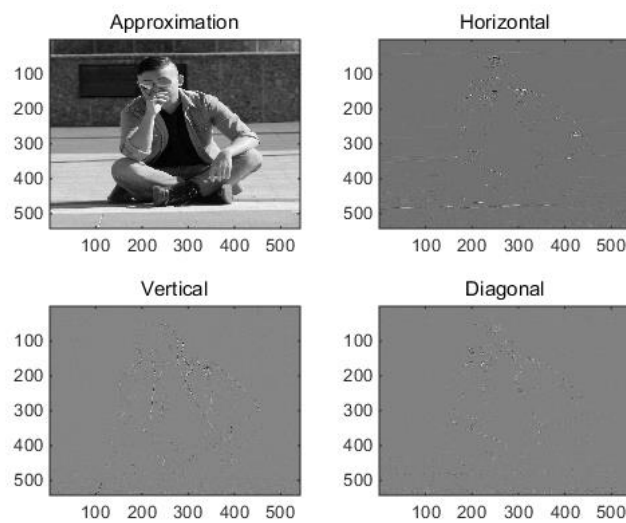
Evidently, infectious networks are PPNs. As can be seen in Figure 9, the network does have definitive features that can be extracted. Links adjacent to the diagonal appear to form clusters. As the graph evolves in time, new links are more likely to join these clusters. Identifying these gradually evolving clusters to allow for the extraction of the underlying pattern (or feature) indicative of the network behavior is the core operation of ENA.

Two-dimensional Discrete Wavelet Transform (2DWT) is popular for image segmentation and image fusion. It is the choice tool for decomposing and synthesizing the network graph

images considered in the following sections. Figure 12 presents the transformation results of applying 2DWT to the grayscale image shown in Figure 11.

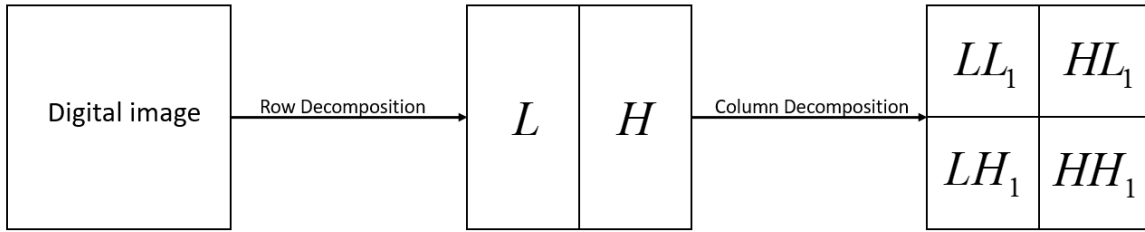


**Figure 11: Original Image**



**Figure 12: 2DWT of Original Image**

2DWT is implemented by applying 1-D DWT (which is a filterbank) to each row of the pixel image to get a high-frequency part simultaneously along with a low-frequency part of the image.[32][33][34]. The 1D DWT decomposition operation is then repeated to process the columns of each of the decomposed channel as indicated in Figure 13. Eventually, the operation generates three high-pass channels that correspond to the vertical, horizontal, and diagonal details, and one low-pass channel that corresponds to the coarse approximation, of the image.



**Figure 13: 2 Dimensional Separable Wavelet Transform**

Similar to the one-dimensional wavelet transform, 2DWT requires a two-dimensional scaling function  $\varphi(x, y)$ , and three two-dimensional wavelet functions  $\Psi^H(x, y)$ ,  $\Psi^V(x, y)$ ,  $\Psi^D(x, y)$ , with each corresponding to the horizontal, vertical, and diagonal channels, respectively. Excluding the products that are one dimensional, such as  $\varphi(x)\psi(y)$ , the four remaining products have the separable scaling functions (2)

$$\varphi(x, y) = \varphi(x)\varphi(y) \quad (2)$$

And separable, “directionally sensitive” wavelets, Equations (3), (4) and (5) which are measures of variations along the columns, rows, and diagonals, respectively.

$$\psi^H = \psi(x)\varphi(y) \quad (3)$$

$$\psi^V = \varphi(x)\psi(y) \quad (4)$$

$$\psi^D = \psi(x)\psi(y) \quad (5)$$

The scaled and translated basis functions are (6) and (7):

$$\varphi_{j,m,n}(x, y) = 2^{\frac{j}{2}} \varphi(2^j x - m, 2^j y - n) \quad (6)$$

$$\psi_{j,m,n}^i(x, y) = 2^{\frac{j}{2}} \psi(2^j x - m, 2^j y - n) \quad (7)$$

where  $i$  is the subscript that assumes the values H, V, and D

The DWT of the function  $f(x, y)$  of size  $M \times N$  is presented in Equations (8) and (9).

$$W_\varphi(j_0, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \varphi_{j_0, m, n}(x, y) \quad (8)$$

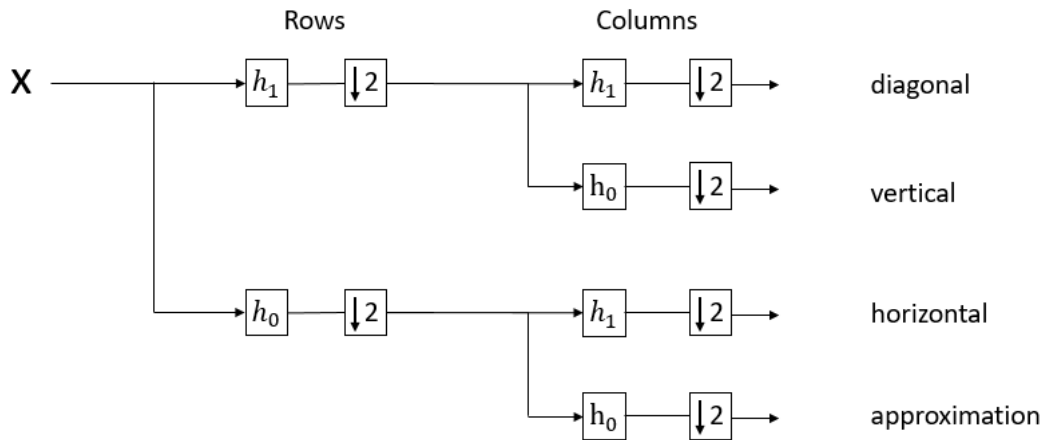
$$W_\psi^i(j, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \psi_{j, m, n}^i(x, y) \quad (9)$$

where  $i = \{H, V, D\}$ ,  $j_0$  is an arbitrary starting scale,  $W_\varphi(j_0, m, m)$  defines the “coarse” information of  $f(x, y)$  and  $W_\psi^i(j, m, m)$  are the wavelet transform coefficients indicating the horizontal, vertical and diagonal details for the scale  $j_0 \leq j$ .

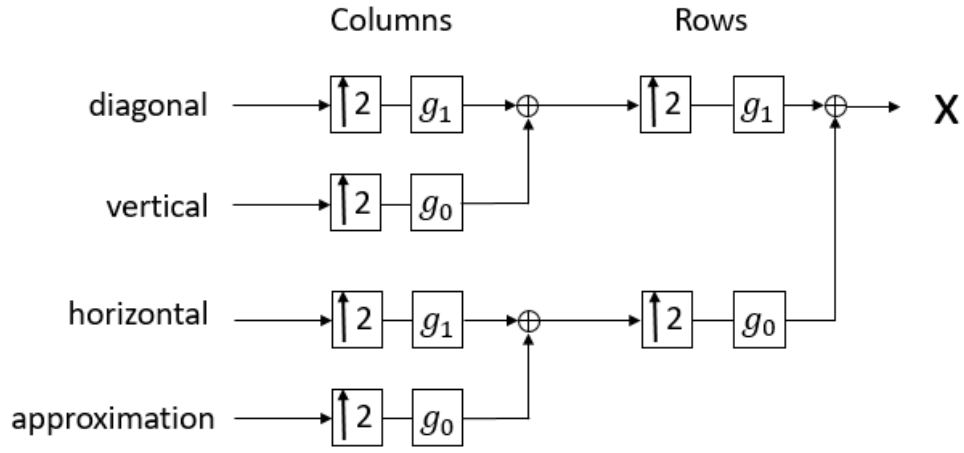
One prominent property of 2DWT is that signals can be fully reconstructed using  $W_\varphi(j_0, m, m)$  and  $W_\psi^i(j, m, m)$  without losing time information. Given  $W_\varphi, W_\psi^i$ ,  $f(x, y)$  can be reconstructed via the inverse discrete wavelet transform (IDWT), Equation (10).

$$f(x, y) = \frac{1}{\sqrt{MN}} \sum_m \sum_n W_\varphi(W_\varphi, m, n) \varphi_{j_0, m, n}(x, y) + \frac{1}{\sqrt{MN}} \sum_{i=H,V,D} \sum_{j=j_0}^{\infty} \sum_m \sum_n W_\psi^i(j, m, n) \psi_{j, m, n}^i(x, y) \quad (10)$$

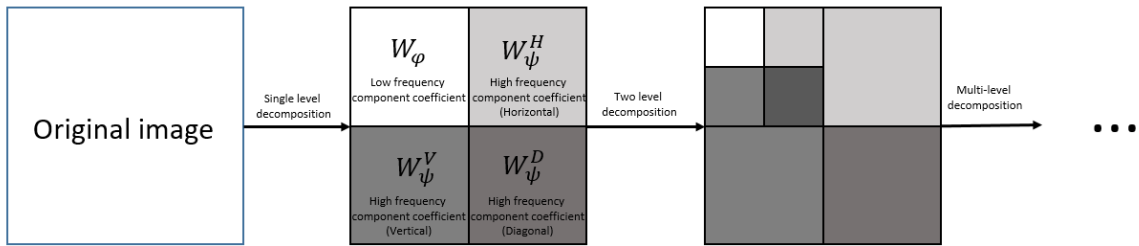
In practice, the process is implemented as a 2D filter operation as shown in Figure 14. where image X is the input to the wavelet filter banks. The operation decomposes the image and generates four wavelet coefficients as outputs providing the diagonal, vertical, and horizontal details and the coarse approximation of the image. The synthesis filtering operation that implements the inverse 2DWT is given in Figure 15. Figure 14 is a single level decomposition. The decomposition can be repeated to achieve multiple-level decomposition as illustrated in Figure 16. A network graph image can be decomposed to multiple levels to allow for the extraction of salient features at different scales.



**Figure 14: Decomposition Wavelet Filterbanks**



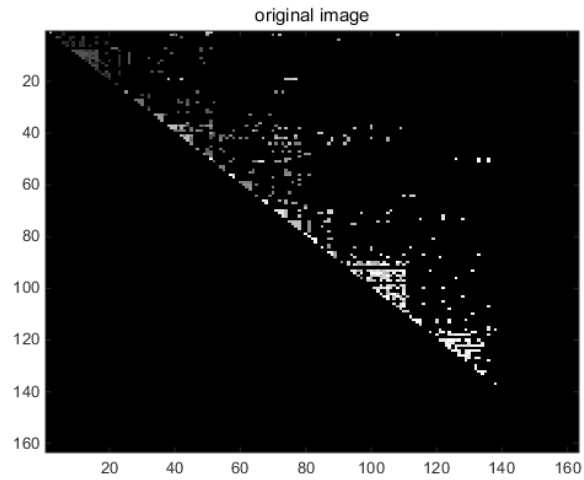
**Figure 15: Synthesis Wavelet Filterbanks**



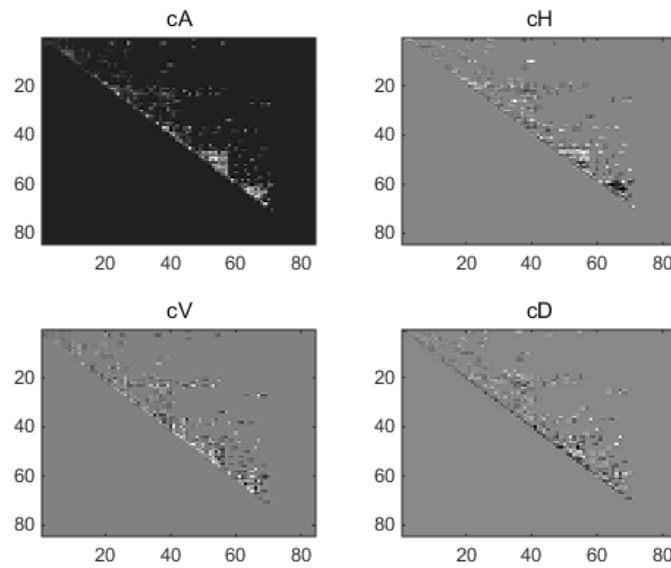
**Figure 16: Multi-Level 2 Dimensional Wavelet Decomposition**

To demonstrate how this is performed the time-decay image of the infectious network in Figure 17 is used as an example. 2DWT is applied to the image using the db3 wavelet to generate the four sub-images in Figure 18. Image cA corresponds to the low-frequency channel

(approximation image) while images cH, cV, and cD correspond to the three high-frequency channel; namely, horizontal, vertical, and diagonal, respectively.



**Figure 17: Time-Decay Image of the infectious network at  $T = 0-2h$**



**Figure 18: Single Level Decomposition of the Infectious Network at  $T = 2$**

### 2.3 Temporal Model in the Wavelet Domain

Recall that in Section 2.2 there are 4 progressive steps in the ENA algorithm that is being developed in the thesis. 2DWT and inverse 2DWT are employed to execute Step 1 and Step 4, respectively. How to implement Step 2 and Step 3 is elaborated in the followings.

The adjacency matrix of the (original) image is first decomposed into four matrices (cA, cH, cV, and cD) that each carries the corresponding wavelet coefficients as results of the 2DWT operation. Again, cA is the "coarse" information (approximation) of the original image and the rest of the three matrices are the "detail" information of the original images. Consequently, the problem is transformed into the wavelet domain with no information loss. Note that the original adjacency matrix can be fully reconstructed using cA, cH, cV, and cD also with no losses.

Because the objective is to extract the underlying pattern using cA, cH, cV, and cD as the network evolves in time, a temporal model of these matrices is therefore needed. The temporal evolution of a time matrix  $A(t)$  can be modeled using tensor factorizations [35][36], an approach that is conceptually the same as fitting discrete data points using polynomials. An  $m$ -by- $n$  time matrix  $A(t)$  can be factorized as follows

$$A(t) = UV(t)^T \quad (11)$$

where  $U$  is a  $m \times k$  scalar matrix and  $V(t)$  is a  $m \times k$  time-dependent matrix.  $V(t)$  can be modeled using the following polynomial function

$$V(t) = W^{(0)} + W^{(1)}t + \dots + W^{(d)}t^d = \sum W^{(i)}t^i \quad (12)$$

where  $n \times k$  matrices  $\{W^{(i)}\}_{i=0}^d$  are to be learned from past data. With the availability of  $W^{(i)}$  the evolving pattern of  $A(t)$  can be extracted. To find a proper  $UV(t)$  that approximates  $A(t)$  with the least amount of error, a loss function needs to be defined. At this point the problem becomes a least square optimization problem (LSO); that is, minimizing the sum of the Frobenius norms of the matrix  $A(t) - UV(t)^T$  over the entire time event,

$$\min_{U,V} J(U,V) = \sum_{t=1}^T \frac{1}{2} \|A(t) - UV(t)^T\|_F^2 \quad (13)$$

The Frobenius norm is the matrix norm of a  $m \times n$  matrix  $A$  defined as the square root of the sum of the absolute squares of its elements as follows

$$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{i,j}|^2} \quad (14)$$

After adding regularization terms along with Eq. (12), the loss function  $J(U, W)$  can be represented as

$$J(U, W) = \sum_{i=1}^T \frac{1}{2} \|A(t) - U(\sum_{i=1}^d W^{(i)} t^i)\|_F^2 + \alpha \|U\|_F^2 + \sum_{i=0}^d \beta_i \|W^{(i)}\|_F^2 \quad (15)$$

where the second and third terms are the sums of all the weights scaled by the regularization parameters  $\alpha$  and  $\beta_i$ .

To optimize the LSO problem is equivalent to minimizing the loss function as

$$\min_{U,V} J(U, W) = \sum_{i=1}^T \frac{1}{2} \|A(t) - U(\sum_{i=1}^d W^{(i)} t^i)^T\|_F^2 + \alpha \|U\|_F^2 + \sum_{i=0}^d \beta_i \|W^{(i)}\|_F^2 \quad (16)$$

The error  $\xi(t)$  at each time instance is

$$\xi(t) = A(t) - U \left( \sum_{j=0}^d W^{(j)} t^j \right)^T \quad (17)$$

Gradient descent method is employed to update  $\{W^{(i)}\}_{i=0}^d$  and to calculate the two partial derivatives below,

$$\frac{\partial J(U, W)}{\partial U} = \sum_{t=1}^T \xi(t) \left( -\sum_{i=0}^d W^{(i)} t^i \right) + 2\alpha U \quad (18)$$

$$\frac{\partial J(U, W)}{\partial W^{(i)}} = \sum_{t=1}^T \xi(t)^T (-U t^i) + 2\beta_i W^{(i)} \quad (19)$$

The model learning algorithm developed for the thesis is summarized as follows

Input: time series adjacent matrices  $\{A(t)\}_{t=1}^d$ , the order d of V(t), and dimension k

Output: matrices U and  $\{W^{(i)}\}_{i=0}^d$  for A(t+1)

1. Set the value of k and d

2. Randomly initialize U and  $\{W^{(i)}\}_{i=0}^d$

3. Set a threshold

4. While not reach the threshold do

5. Compute the  $\xi(t)$  for each time stamp

6. Compute partial derivatives  $\frac{\partial J(U, W)}{\partial U}$ ,  $\frac{\partial J(U, W)}{\partial W^{(i)}}$  using equation (18) (19)

7. Determine the learning rate  $\lambda$

8. Update  $U = U - \lambda \frac{\partial J(U, W)}{\partial U}$

9. for i = 1...d do

10.           Update  $W^{(i)} = W^{(i)} - \lambda \frac{\partial J(U, W)}{\partial W^{(i)}}$

11.       end

12. end

13. Compute  $A(T+1) = UV(T+1)^T = U(\sum_{i=0}^d W^{(i)}(T+1)^i)^T$

## 2.4 Wavelet Coefficient Temporal Model (WCTM) for Evolving Network Analysis

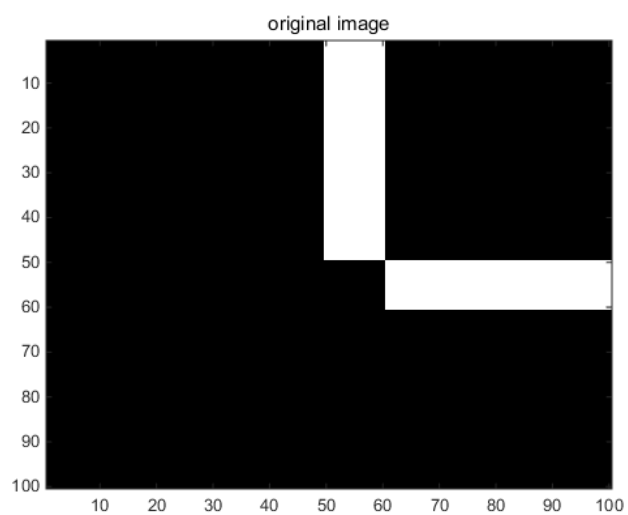
With the tools introduced in Sections 2.2 and 2.3 developed, the four steps mentioned in Section 2.1 can be followed to realize a general framework for evolving network analysis (ENA).

Once a network graph image is decomposed into the corresponding high-frequency and low-frequency components using 2DWT, a temporal model of the network can be developed by following the methodology outlined in Section 2.3. The temporal model is then used to project the next images of the components as the network evolves. The images of the individual components are then assembled together using the inverse wavelet transform to generate a new graph image of the network. From the scores of the grayscale image matrix, one can predict whether a node-pair would connect, or not, in the next moment.

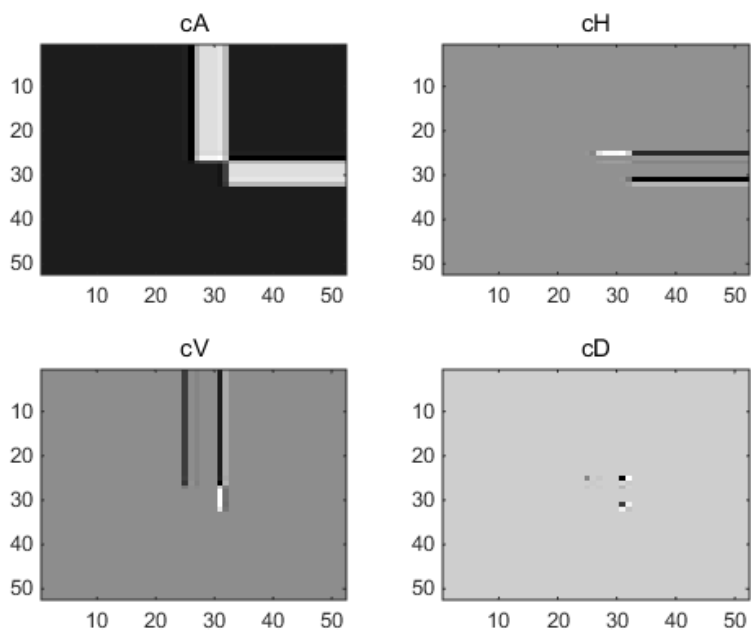
As mentioned in Section 2.2 that wavelet transform can help extract the feature that underlies a gradually changing network. Applying 2DWT to the 100-node network given in Figure 19 in which the critical node (tagged as 50) is connected to the rest of the 99 nodes. (A node can't connect to itself.) This is therefore an unweighted binary network where two connected nodes

receive a 1 and a 0 if there is no connection. Figure 20 plots the corresponding 2DWT of the image in Figure 19.

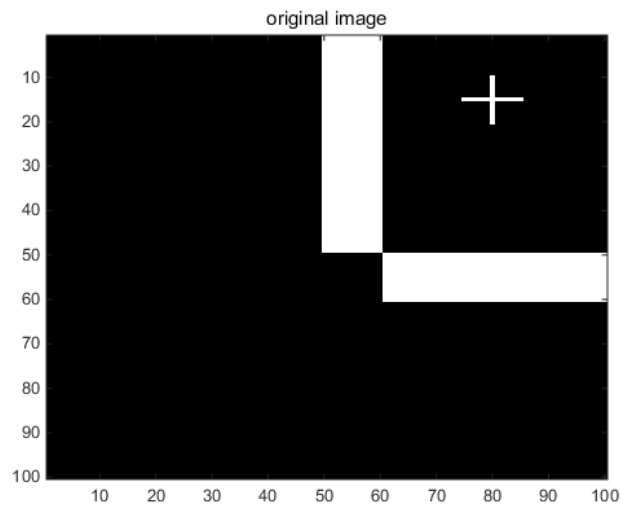
From Figure 19 and Figure 20, it is noted that the connections of the vital node "disappear" from the high-frequency image. New connections between the node group tagged 50-60 with the node group tagged 1-49 are clearly indicated in Figure 21 and the corresponding 2DWT in Figure 22. In the high-frequency images, the small group connection is visible and information on the vital node is not available. The connection of the vital node is indicated by the "coarse" (low frequency) information while the connections of the rest of the nodes are indicated by the "detail" (high frequency) information. Using wavelet transform, "coarse" and "detail" information of a network can be extracted without aliasing. In addition to providing low-frequency information and high-frequency information in the simultaneous time-frequency domain, 2DWT also effectively lowers the dimensionality of the network analysis problem. That no information is lost either through wavelet decomposition or via synthesis is essential to ensure that the underlying feature of a network can be extracted from the 'coarse' and 'detail' information without distortion.



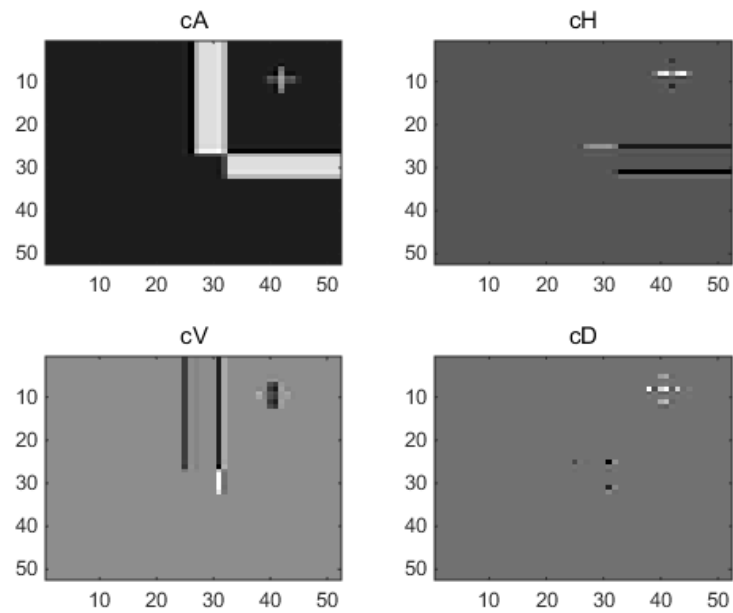
**Figure 19: Network Image**



**Figure 20: 2DWT of the Network Image in Figure 19**

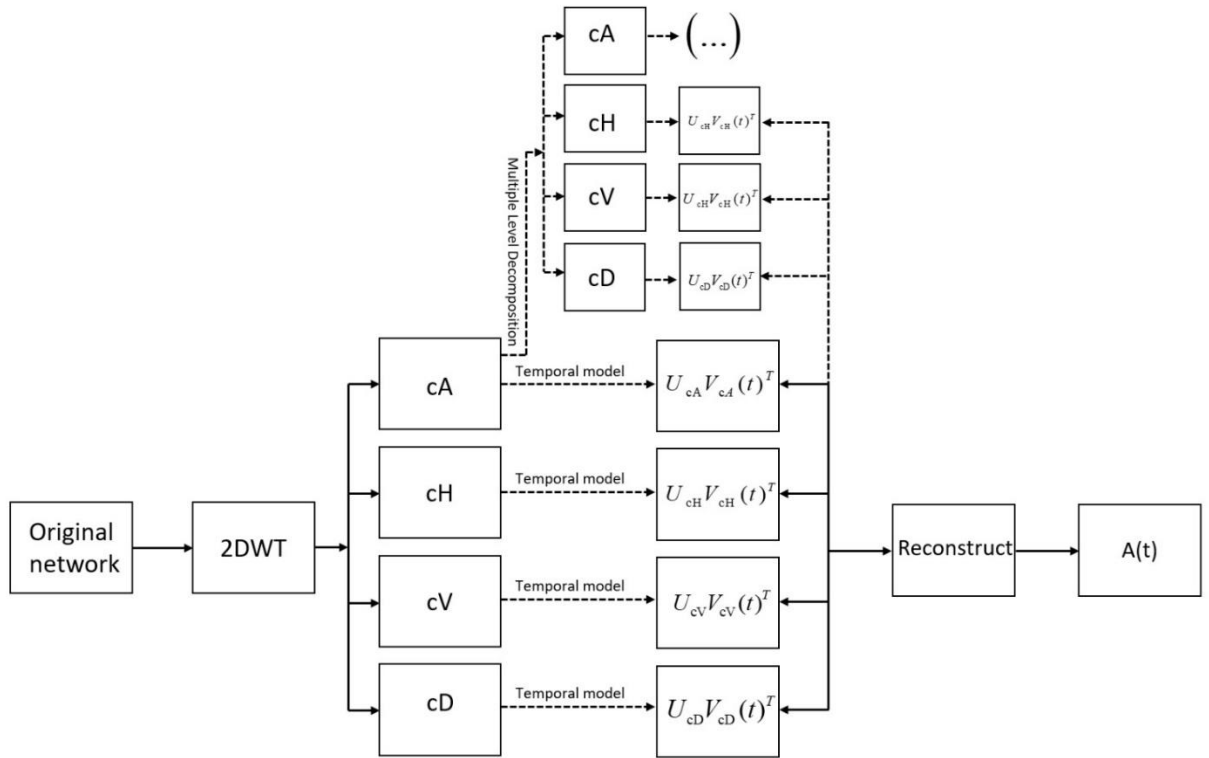


**Figure 21: Network Image with More Details**



**Figure 22: 2DWT of the Network Image in Figure 21**

Figure 23 is the implementation of the 4-step methodology feasible for evolving network analysis. It is also a general framework for modeling ENA problems. The first step is to decompose the network into its cA, cH, cV, and cD wavelet coefficients. 2DWT is to be applied again to the cA subset should the next level of decomposition is needed. A temporal model is then followed to learn the underlying pattern of the evolving frequency coefficients. In the last step, the network image is reconstructed in the time domain using the wavelet coefficients.



**Figure 23: General Framework for Evolving Network Analysis**

### 3 BENCHMARK DATASETS AND LINK PREDICTION ALGORITHMS

#### 3.1 Dataset Description

In this section, 5 benchmark network datasets are introduced. Networks in the real world are dynamic. They evolve in time. Depending on the method utilized for data recording and acquisition, there are static dataset and evolving dataset. Facebook (nips) network dataset [37] is a static dataset where all data come with no time stamp. In the directed network of user-user friendships, a node represents a user and an edge indicates that the user represented by the left node is a friend of the user represented by the right node. In this network, there is no time information to tell when two users became friends on Facebook. However, in the real-world friendship networks evolves in time.

Figure 24 shows the first 20 rows in the Facebook dataset. The first column is the users on the left and the second column is the users on the right.

Timestamp information is provided to data in evolving datasets. Figure 25 is the first 20 rows of the infectious network dataset. The third column is the timestamp at which the edge appears in the network. Network datasets used in this thesis are evolving datasets. Five evolving datasets are listed in the followings. More details are found in Table 1[38] [39] [40][41]:

1	% sym unweighted
2	% 2981 2888 2888
3	1 2
4	1 3
5	1 4
6	1 5
7	1 6
8	1 7
9	1 8
10	1 9
11	1 10
12	1 11
13	1 12
14	1 13
15	1 14
16	1 15
17	1 16
18	1 17
19	1 18
20	1 19

**Figure 24: First 20 Rows of a Static Dataset**

1	1,2,0.0
2	1,2,0.0055555555555555556
3	1,2,0.0111111111111111112
4	1,2,0.0333333333333333333
5	1,2,0.0388888888888888889
6	5,6,0.0388888888888888889
7	3,4,0.0388888888888888889
8	5,6,0.0444444444444444446
9	5,7,0.0444444444444444446
10	6,7,0.0444444444444444446
11	3,4,0.0444444444444444446
12	1,2,0.0444444444444444446
13	5,6,0.05
14	3,4,0.05
15	1,2,0.0555555555555555555
16	5,6,0.0666666666666666667
17	1,2,0.0666666666666666667
18	6,7,0.0666666666666666667
19	3,4,0.0666666666666666667
20	5,6,0.0722222222222222222

**Figure 25: First 20 Rows of an Evolving Dataset**

- Infectious

This network describes the face-to-face behavior of people during the exhibition INFECTIOUS: STAY AWAY in 2009 at the Science Gallery in Dublin. Nodes represent exhibition visitors while edges represent face-to-face contacts that were active for at least 20 seconds. Multiple edges between two nodes are possible. The network contains the data collected from the day with the most interactions.

- Hagggle

This undirected network represents contacts between people measured by carried wireless devices. A node represents a person and an edge between two persons indicates that there was a contact between them.

- arXiv hep-th

This is the collaboration graph of authors of scientific papers from the arXiv's High Energy Physics – Theory (hep-th) section. An edge between two authors represents a common publication. Timestamps denote the date of a publication.

- DNC emails

This is the directed network of emails in the 2016 Democratic National Committee email leak. The Democratic National Committee (DNC) is the formal governing body for the United States Democratic Party. A dump of emails of the DNC was leaked in 2016. Nodes in the network correspond to persons in the dataset. A directed edge in the dataset denotes that a person has

sent an email to another person. Since an email can have any number of recipients, a single email is mapped to multiple edges in this dataset, resulting in the number of edges in this network being about twice the numbers of emails in the dump.

- UCI Message

This directed network contains sent messages between the users of an online community of students at the University of California, Irvine. A node represents a user. A directed edge represents a sent message. Multiple edges denote multiple messages.

<i>Dataset</i>	<i>infectious</i>	<i>Haggle</i>	<i>arXiv</i>	<i>dnc</i>	<i>UCI</i>
<i>Direct/not</i>	not	not	not	dir	dir
<i>Weight/not</i>	N	N	N	N	N
<i>Time range</i>	7.89 h	4d	9y	32.7mo	6.5mo
<i>Size</i>	401	274	22908	2029	1899
<i>Volume</i>	17298	28244	2673133	39264	59835
<i>Aver. degree</i>	84.4	206.16	233.4	38.7	63.017
<i>Max. degree</i>	294	2092	11967	5813	1546
<i>Cluster coeff.</i>	43.6%	56.6%	26.9%	8.9%	5.68%

**Table 1: Details of Experimental Datasets**

### 3.2 SNA Methods

6 mainstream link prediction algorithms are introduced in the section. They are all similarity based [26]. The first three are local similarity-based algorithms and the rest are global similarity-based algorithms. In these algorithms, a score  $s_{xy}$  is assigned to each node pair (x, y) in the network. The score is defined as the similarity of the node pair. All non-observed nodes are given a score. All the nodes are then ranked using their scores. The higher the score, the more likely they will appear in the network in the next time instance. Similarity-based algorithms can be simple or complex and they can be effective for some networks and ineffective for other networks. As mentioned in the introduction section, the attribute of the data is not being considered in this thesis. So similarity-based algorithms introduced in the followings are all structural similarity algorithms.

- Common Neighbors (CN) Index

For node x in the network,  $\Gamma(x)$  is the set of neighbors of node x. It is common sense that x and y are more likely to connect with each other if they share more common neighbors. So the similarity of node x and y can be defined as:

$$s_{xy}^{CN} = |\Gamma(x) \cap \Gamma(y)|$$

- Jaccard index

Similar to CN, Jaccard index is also a local similarity index which was proposed by Jaccard over a hundred years ago.

$$s_{xy}^{Jaccard} = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$$

- Resource Allocation Index (RA)

RA was motivated by resource allocation dynamics on the complex network [42].  $x$  and  $y$  are nodes in the network and they are not connected with each other directly. Assume that  $x$  can transmit resource to  $y$  through common neighbors. If a node has one unit resource which is distributed equally to all its neighbors, the similarity of node  $x$  and  $y$  can be defined as the amount of resources  $y$  received from  $x$ .

$$s_{xy}^{RA} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{k_z}$$

- Average Commute Time (ACT) Index

$m(x, y)$  denotes the average number of steps required by a random walker starting from node  $x$  to reach  $y$ . Then the average of time of commute between  $x$  and  $y$  is

$$n(x, y) = m(x, y) + m(y, x)$$

alternatively,  $n(x, y)$  can be represented as [43][44]

$$n(x, y) = M(l_{xx}^+ + l_{yy}^+ - 2l_{xy}^+)$$

where  $l_{xy}^+$  denotes the corresponding entry in  $L^+$ . The similarity of two nodes here can be defined as if they have similar average time of commute. And it can be represented as

$$s_{xy}^{ACT} = \frac{1}{l_{xx}^+ + l_{yy}^+ - 2l_{xy}^+}$$

- Matrix Forest Index (MFI)

In MFI, the similarity  $x$  and  $y$  can be understood as the ratio of the number of spanning rooted forests such that nodes  $x$  and  $y$  belong to the same tree rooted at  $x$  to all spanning rooted of the network [45].

$$S = (I + \alpha L)^{-1}, \alpha > 0$$

- Preferential Attachment Index

At each time step, an old link is removed and a new link is generated. The probability that this new link will connect  $x$  and  $y$  is proportional to  $k_x \times k_y$ . Motivated by this mechanism, the corresponding similarity index can be defined as:

$$s_{xy}^{PA} = k_x \times k_y$$

PA is widely used to quantify the functional significance of links subject to various network-based dynamics such as percolation, synchronization, and transportation.

### 3.3 Evaluation Methods

A basic metric for evaluating link prediction algorithms, AUC is used to determine the accuracy of the predicted result by calculating the area under the ROC (receiver operating characteristic curve) [46]. AUC is calculated using the following equation:

$$AUC = \frac{n' + 0.5n''}{n}$$

where  $n$  is the independent comparison times while  $n'$  is the number of times of missing links having a higher score and  $n''$  is the number of times they have the same score. When doing link

prediction, every link in the set  $U - E$  (nonexistent set, more details can be found in Section 2.1) is scored and missing links and nonexistent links are randomly picked to compare their scores. In order words, the closer of AUC to 1, the better the algorithm predicts.

The followings are an example on AUC. Figure 26 presents a network with 5 nodes at time T and Figure 27 presents the network at time T+1 (where the dot-line is the new links of the test dataset). There are 5 nodes and 10 possible edges in this network. Figure 26 shows that there are 7 observed edges in the network. Prediction algorithm scores 3 unobserved links and returns rates:  $\{s_{12}=0.4, s_{13}=0.5, s_{45}=0.6\}$ .  $\{s_{13}=0.5, s_{45}=0.6\}$  is the rate set of the test data and  $\{s_{12}=0.4\}$  is rate set of the nonexistent links. The rate in the test data is compared with the rates of the nonexistent links.

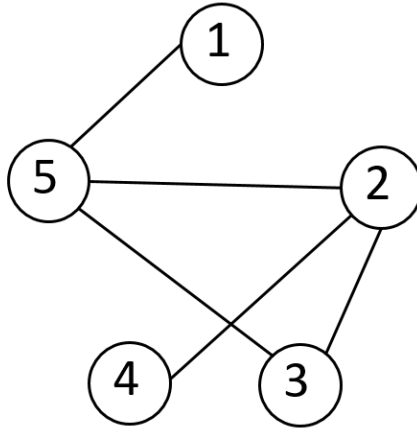
For  $\{s_{13}=0.5\}$

$$s_{13} = 0.5 > s_{12} = 0.4 \text{ (AUC +1)}$$

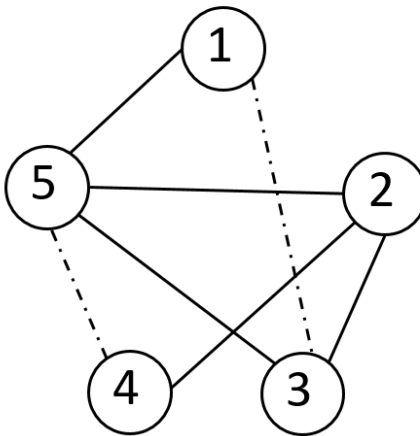
For  $\{s_{45}=0.6\}$

$$s_{45} = 0.6 > s_{12} = 0.4 \text{ (AUC +1)}$$

In this case, there are 3 pairs of test links that are larger than the nonexistent links and 2 pairs are equal to the nonexistent links. Then  $AUC = (1+1) / 2 = 1$



**Figure 26: Links for Training**



**Figure 27: Observed Links with Test Links**

## 4 EXPERIMENT AND DISCUSSION

In this chapter, the WCTM general framework model developed in Chapter 3 is evaluated using AUC as the evaluation metric. Static network analysis methods introduced in Chapter 3 are also applied to benchmark datasets for comparison with WCTM.

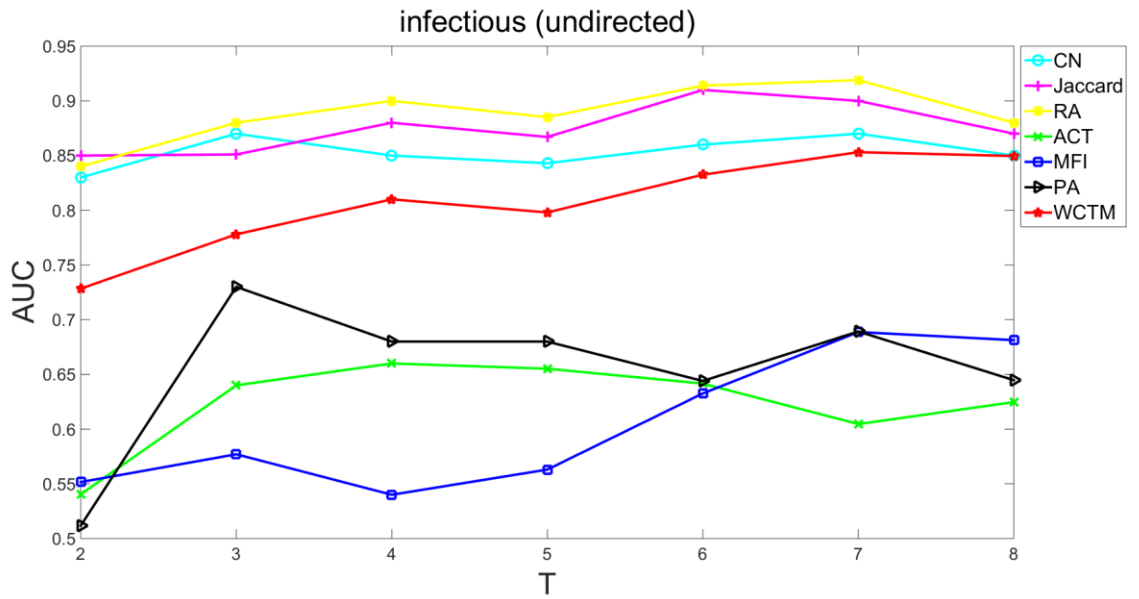
In this section, the WCTM experiment results and benchmarks are first presented. The T-1 time is chosen as the training set and the  $T^{\text{th}}$  time is the test set. The accuracy of the link prediction algorithm is evaluated using AUC.

As discussed in Chapter 2, the WCTM is dictated by the parameters in Table 2. In the experiment, through tuning these parameters, the values of these parameters as in the third column of Table 2 for all five network analysis experiments are defined.

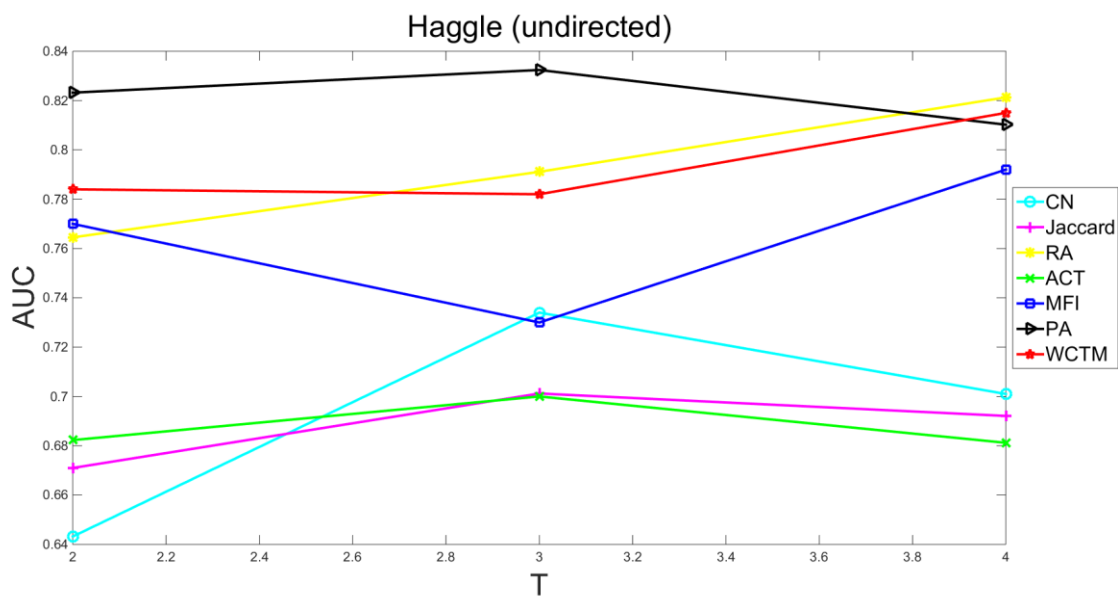
The experimental result is shown in Figure 28 to Figure 32 where the six benchmarks are compared with the WCTM result using the same parameters.

<i>Parameter</i>	<i>Definition</i>	<i>Experimental Setting</i>
$k$	Rank of the decomposed matrix $V(t)$	9
$d$	Dimension of the fitting polynomial $V(t)$	1
$\lambda$	Learning rate	0.001
$\alpha$	Overfitting regulator	0.01
$\beta$	Overfitting regulator	0.01
$\theta$	Decay rate	0.1

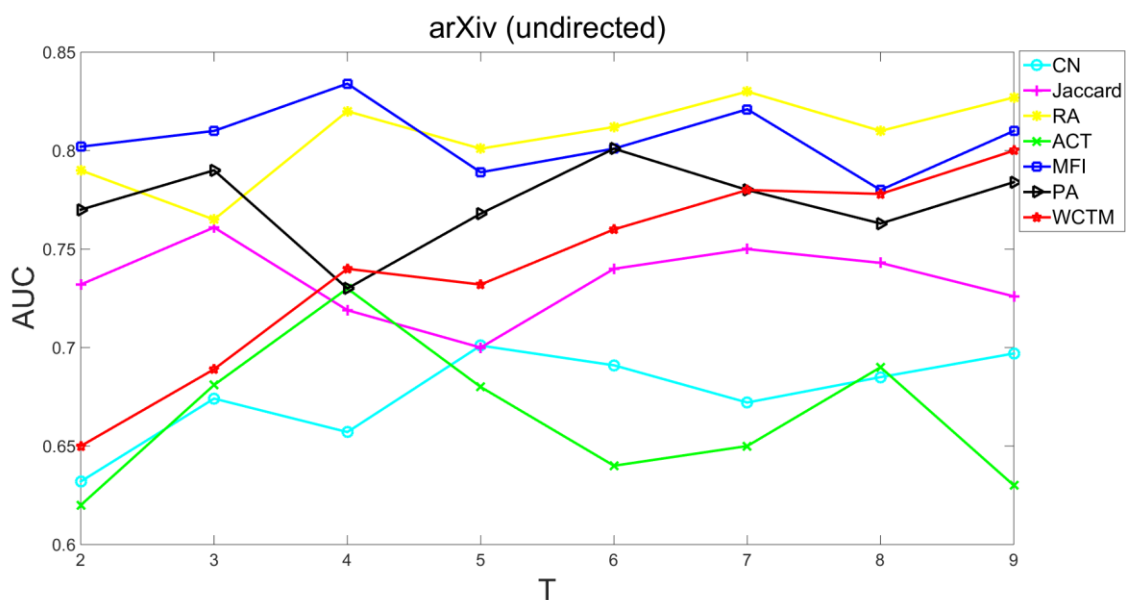
**Table 2: Model Parameters and Experimental Setting**



**Figure 28: AUC Score for All Algorithms in Infectious Network**



**Figure 29: AUC Score for All Algorithms in Haggle Network**



**Figure 30: AUC Score for All Algorithms in arXiv Network**

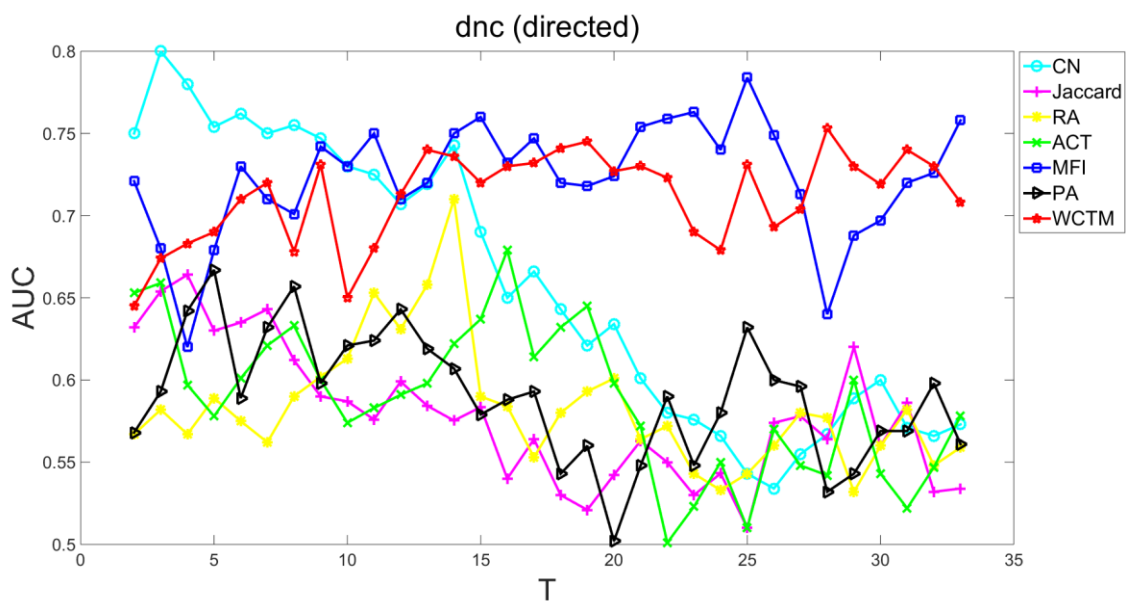


Figure 31: AUC Score for All Algorithms in DNC Network

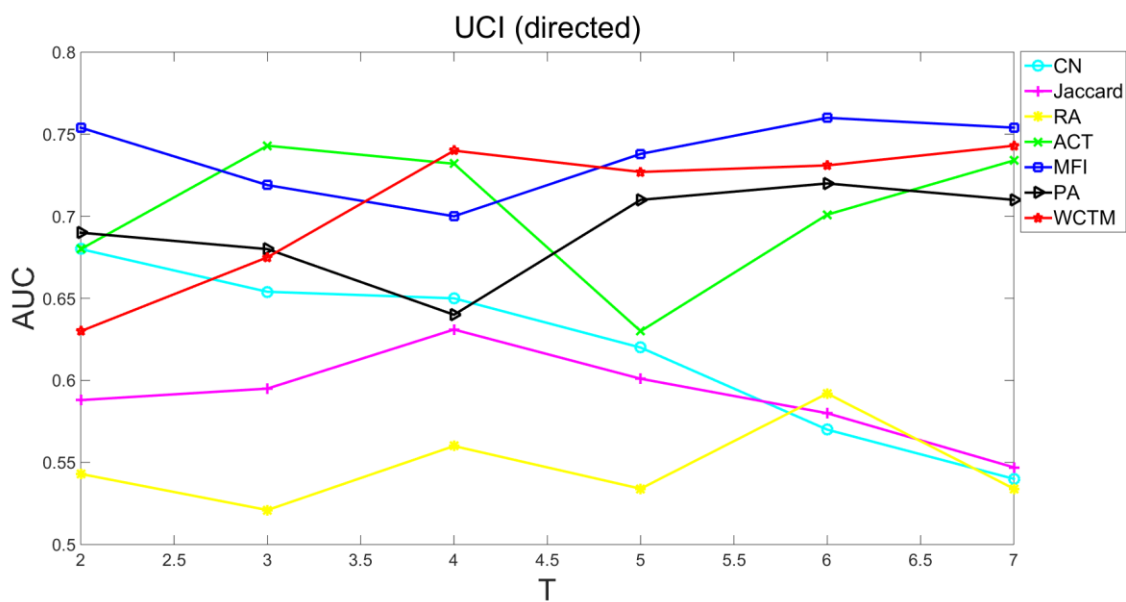


Figure 32: AUC Score for All Algorithms in UCI Message Network

The six algorithms are evaluated using the 5 network datasets in which three are undirected and two are directed. The accuracy of each algorithm is determined using the AUC metric.

From the experimental result it is seen that

- WCTM performs well on the 3 undirected networks. Though not the best one, WCTM are within 7%, 0.8%, and 3% on the corresponding AUC values with the most accurate algorithm for all the 3 networks.
- It is evident that WCTM is more competitive than all the benchmarks on the directed networks. Only MFI performs slightly better than WCTM on the two directed networks. The differences in prediction accuracy are 6% and 1.3%.
- WCTM leads all the link prediction algorithms in consistently giving a high level of prediction accuracy for all the networks considered.
- Table 3 gives the average AUC values of each of the prediction algorithm. The accuracy of WCTM is among the top 3 in all cases.
- The networks analyzed in the experiment include social networks and communication networks which represent a wide range of real-world networks. WCTM is applicable to all of them.

<b>ALGORITHMS</b>	<b>INFECTIOUS</b>	<b>HAGGLE</b>	<b>ARXIV</b>	<b>DNC</b>	<b>UCI</b>
<b>CN</b>	0.8533	0.6927	0.6762	0.6577	0.6190
<b>JACCARD</b>	0.8754	0.6881	0.7339	0.5783	0.5903
<b>RA</b>	0.8883	0.7923	0.8069	0.5829	0.5473
<b>ACT</b>	0.6238	0.6878	0.6651	0.5882	0.7033
<b>MFI</b>	0.6049	0.7640	0.8059	0.7230	0.7375
<b>PA</b>	0.6542	0.8219	0.7732	0.5903	0.6917
<b>WCTM</b>	0.8071	0.7931	0.7411	0.7117	0.7077
<b>AVERAGE</b>	0.7581	0.7487	0.7532	0.6332	0.6567

**Table 3: Average AUC Value for All the Algorithms**

## 5 CONCLUSION AND FUTURE WORK

### 5.1 Conclusion

A general link prediction model – WCTM is developed in the thesis for evolving network analysis. Different from all the contemporary evolving network analysis ideas, WCTM decomposes an evolving network into the wavelet time-frequency domain to extract the underlying evolving pattern of the wavelet coefficients. Compared with static analysis methods, WCTM offers the following benefits:

- Reduced network dimensionality

WCTM performs link predictions not by calculating the adjacency matrix directly. Consider an  $N$ -node network as an example. The size of the adjacency matrix of the network is  $N \times N$ . Rather than calculating the matrix directly, WCTM decomposes the  $N \times N$  matrix into  $4 \times \frac{N+R}{2} \times \frac{N+R}{2}$  matrices using  $R$  numbers of wavelet coefficients. The implications for reduced matrix dimensionality are many including improved computational efficiency and larger capacity for handling networks with large numbers of nodes.

- Feature identification in the simultaneous time-frequency domain

Network graphs are decomposed using 2DWT into “coarse” (low frequency) networks and “details” (high frequency) networks without losing any information. While it is difficult to find the inherent pattern in the time-domain, features indicative of link addition or removal are readily

resolved in the frequency domain. As problems of evolving networks can range from being linear to stochastic [47], being able to identify salient features in the simultaneous time-frequency is of significant implications for addressing both evolving and static networks.

- **Dynamic link prediction**

Link prediction algorithms that are single snapshot based can only describe a network as it evolves into the next moment in time. WCTM can project the future responses of a network in a continuous manner using a temporal model.

WCTM provides a general framework for link prediction. It is shown to be applicable to a wide range of characteristically different networks, evolving and static alike. Differ from snapshot-based methodologies, the temporal model of WCTM's explores the development of the network in time rather than analyzing the network structure using the heuristic approach. In this regard, WCTM is conceptually not unlike the system identification feature commonly found in adaptive control.

## **5.2 Future Work**

When considering the development of a network in time ( $A(t) = UV(t)$ ), WCTM transforms the problem into an LSO problem. As such one of the future works is to investigate other optimization methods for better prediction performance. In addition, WCTM can be applied to

- More network problems such as rating networks, hyperlink networks, traffic networks to generate knowledge essential for understanding and controlling the networks.

- Explore ENA problems other than link prediction including but not limited to anomaly detection and community detection.

## REFERENCES

- [1] Barnes, J.A., 1954. Class and committees in a Norwegian island parish. *Human Relations*, 7(1), pp.39-58.
- [2] Infectious network dataset -- KONECT, April 2017 [<http://konect.uni-koblenz.de/networks/sociopatterns-infectious>]
- [3] Scott, J., 2017. *Social Network Analysis*. Sage.
- [4] Zhang, M. and Chen, Y., 2017, August. Weisfeiler-Lehman neural machine for link prediction. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 575-583). ACM.
- [5] Pech, R., Hao, D., Pan, L., Cheng, H. and Zhou, T., 2017. Link prediction via matrix completion. *EPL (Europhysics Letters)*, 117(3), p.38002.
- [6] Ghasemian, A., Galstyan, A. and Clauset, A., 2017. Highly Accurate Link Prediction in Networks Using Stacked Generalization.
- [7] Lin, D., 1998, July. An information-theoretic definition of similarity. In *ICML* (Vol. 98, No. 1998, pp. 296-304).
- [8] Leicht, E.A., Holme, P. and Newman, M.E., 2006. Vertex similarity in networks. *Physical Review E*, 73(2), p.026120.
- [9] Sun, D., Zhou, T., Liu, J.G., Liu, R.R., Jia, C.X. and Wang, B.H., 2009. Information filtering based on transferring similarity. *Physical Review E*, 80(1), p.017101.

- [10] Sales-Pardo, M., Guimera, R., Moreira, A.A. and Amaral, L.A.N., 2007. Extracting the hierarchical organization of complex systems. *Proceedings of the National Academy of Sciences*, 104(39), pp.15224-15229.
- [11] White, H.C., Boorman, S.A. and Breiger, R.L., 1976. Social structure from multiple networks. I. Blockmodels of roles and positions. *American journal of sociology*, 81(4), pp.730-780.
- [12] Zhou, C., 2006. C. Zhou, L. Zemanová, G. Zamora, CC Hilgetag, and J. Kurths, *Phys. Rev. Lett.* 97, 238103 (2006). *Physical Review Letters*, 97, p.238103.
- [13] Neville, J., 2006. Statistical models and analysis techniques for learning in relational data.
- [14] Heckerman, D., Meek, C. and Koller, D., 2004. Probabilistic models for relational data. Technical Report MSR-TR-2004-30, Microsoft Research.
- [15] Yu, K., Chu, W., Yu, S., Tresp, V. and Xu, Z., 2007. Stochastic relational models for discriminative link prediction. In *Advances in Neural Information Processing Systems* (pp. 1553-1560).
- [16] Fortunato, S., 2010. Community detection in graphs. *Physics Reports*, 486(3-5), pp.75-174.
- [17] Fortunato, S. and Castellano, C., 2012. Community structure in graphs. In *Computational Complexity* (pp. 490-512). Springer, New York, NY.
- [18] Girvan, M. and Newman, M.E., 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12), pp.7821-7826.
- [19] Wu, F.Y., 1982. The potts model. *Reviews of Modern Physics*, 54(1), p.235.

- [20]Chandola, V., Banerjee, A. and Kumar, V., 2009. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3), p.15.
- [21]Taylor, O. and Addison, D., 2000. Novelty detection using neural network technology. In *COMADEM 2000: 13th International Congress on Condition Monitoring and Diagnostic Engineering Management* (pp. 731-743).
- [22]Boriah, S., Chandola, V. and Kumar, V., 2008, April. Similarity measures for categorical data: A comparative evaluation. In *Proceedings of the 2008 SIAM International Conference on Data Mining* (pp. 243-254). Society for Industrial and Applied Mathematics.
- [23]Guha, S., Rastogi, R. and Shim, K., 2000. ROCK: A robust clustering algorithm for categorical attributes. *Information Systems*, 25(5), pp.345-366.
- [24]Anscombe, F.J., 1960. Rejection of outliers. *Technometrics*, 2(2), pp. 123-146.
- [25]Aggarwal, C. and Subbian, K., 2014. Evolutionary network analysis: A survey. *ACM Computing Surveys (CSUR)*, 47(1), p.10.
- [26]Lü, L. and Zhou, T., 2011. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and Its Applications*, 390(6), pp.1150-1170.
- [27]Pastor-Satorras, R. and Vespignani, A., 2001. Epidemic spreading in scale-free networks. *Physical Review Letters*, 86(14), p.3200.
- [28]Newman, M.E., Strogatz, S.H. and Watts, D.J., 2001. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64(2), p.026118.
- [29]ERDdS, P. and R&WI, A., 1959. On random graphs I. *Publ. Math. Debrecen*, 6, pp.290-297.

- [30] Watt, D.J. and Strogatz, S.H., 1998. . Collective dynamics of ‘small-world’ networks. networks. Nature, 393(6684), p.440.
- [31] Barabási, A.L. and Albert, R., 1999. Emergence of scaling in random networks. Science, 286(5439), pp.509-512.
- [32] Antonini, M., Barlaud, M., Mathieu, P. and Daubechies, I., 1992. Image coding using wavelet transform. IEEE Transactions on Image Processing, 1(2), pp.205-220.
- [33] Daubechies, I., 1990. The wavelet transform, time-frequency localization and signal analysis. IEEE transactions on Information Theory, 36(5), pp.961-1005.
- [34] Daubechies, I., 1992. Ten Lectures on Wavelets (Vol. 61). SIAM.
- [35] Dunlavy, D.M., Kolda, T.G. and Acar, E., 2011. Temporal link prediction using matrix and tensor factorizations. ACM Transactions on Knowledge Discovery from Data (TKDD), 5(2), p.10.
- [36] Yu, W., Aggarwal, C.C. and Wang, W., 2017, February. Temporally factorized network modeling for evolutionary network analysis. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (pp. 455-464). ACM.
- [37] Facebook (nips) network dataset -- KONECT, April 2017. [<http://konect.uni-koblenz.de/networks/ego-facebook>]
- [38] Huggle network dataset -- KONECT, April 2017. [<http://konect.uni-koblenz.de/networks/contact>]

- [39]arxiv hep-th network dataset -- KONECT, April 2017. [<http://konect.uni-koblenz.de/networks/ca-cit-HepTh>]
- [40]Dnc emails network dataset -- KONECT, April 2017. [<http://konect.uni-koblenz.de/networks/dnc-temporalGraph>]
- [41]Uc irvine messages network dataset -- KONECT, April 2017. [<http://konect.uni-koblenz.de/networks/opsahl-ucsocial>]
- [42]Ou, Q., Jin, Y.D., Zhou, T., Wang, B.H. and Yin, B.Q., 2007. Power-law strength-degree correlation from resource-allocation dynamics on weighted networks. *Physical Review E*, 75(2), p.021102.
- [43]Fouss, F., Pirotte, A., Renders, J.M. and Saerens, M., 2007. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 19(3), pp.355-369.
- [44]Klein, D.J. and Randić, M., 1993. Resistance distance. *Journal of Mathematical Chemistry*, 12(1), pp.81-95.
- [45]Chebotarev, P. and Shamis, E., 2006. The matrix-forest theorem and measuring relations in small social groups. arXiv preprint [math/0602070](https://arxiv.org/abs/math/0602070).
- [46]Hanley, J.A. and McNeil, B.J., 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1), pp.29-36.

- [47] Yasami, Y. and Safaei, F., 2018. Detecting chaotic behaviors in dynamic complex social networks using a feature diffusion-aware model. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(6), p.061106.