COMPLEX NETWORK TOOLS TO ENABLE
IDENTIFICATION OF A CRIMINAL COMMUNITY

A thesis submitted
in fulfilment of the requirements for the degree of

Doctor of Philosophy

by

Pritheega Magalingam
BSc. (Computer Science), MSc. (Information Security)

School of Mathematical and Geospatial Sciences
College of Science, Engineering and Health
RMIT University
Melbourne
Australia
July, 2015
To my beloved father, Magalingam Vadivallet and mother, Komothi Chelliah.
DECLARATION

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis/project is the result of work which has been carried out since the official commencement date of the approved research program; any editorial work, paid or unpaid, carried out by a third party is acknowledged; and, ethics procedures and guidelines have been followed.

Pritheega Magalingam
Tuesday 14th July, 2015
Acknowledgements

My sincere gratitude to my primary supervisor, Associate Prof. Dr. Asha Rao for accepting me as her Ph.D. student, believing in me from the beginning and being a great support during each phase of my Ph.D. research program in every possible way. Her advice on both my studies as well as on my life has been priceless. She has been a wonderful mentor, inspired me to continue learning, motivated me to keep trying and encouraged me to develop new ideas with an open and positive mind. She helped me to overcome the fear of failure, fear of losing and the challenges. Throughout this Ph.D. journey, with her guidance and support, I have learned and developed a lot personally and professionally.

While my first supervisor is a great blessing from God, I would say, my second supervisor, Dr. Stephen Davis is a fortune given to me. I would like to thank him for his constant support and time-to-time valuable feedback. He has always shown his interest in my ideas. His comments and suggestions have contributed to the improvement in many technical aspects of my research. My sincere regards and appreciation to both my supervisors for providing the necessary directions and guidance to my research, on finalising this thesis and associated papers.

Words cannot express how thankful I am to my parents. This journey would have never been successful without their blessings. They always encouraged and made me emotionally strong when I felt low. It is their prayers that sustained me thus far. I would also like to thank my brother Saravana Magalingam who has sacrificed his valuable years taking care and staying with our parents instead of spreading his wings finding his favourable career path. I am indebted to my brother for taking the huge responsibility of caring for them during my absence. My love and thanks to my grandma Suppatha Perumal who has offered her valuable advice whenever I visited her during my holidays.

It has been an honour to work with all my fellow graduate students in School of Mathematical and Geospatial Sciences (SMGS). Thank you for the invaluable knowledge sharing and critics through seminars and group discussions. Many thanks to all of my friends who have supported and motivated me along this journey, especially Mia for her kindness and the time she spent for me during my stressful moments.

Lastly but not the least, I would like to extend my gratitude to SMGS staff who patiently responded to all my queries and have been very helpful throughout my Ph.D. candidature.
Contents

1 INTRODUCTION 3
   1.1 Overview ....................................................... 3
   1.2 Network Science .............................................. 3
   1.3 Money Laundering Network Detection and the Inherent Problems ........................................... 5
   1.4 Motivation and Research Scope .................................. 6
   1.5 Thesis Objectives and Contributions ............................... 9
      1.5.1 Objectives .................................................. 9
      1.5.2 Contributions .............................................. 9
   1.6 Structure of this thesis .......................................... 10

2 BACKGROUND 12
   2.1 Datasets ...................................................... 12
      2.1.1 Enron Data ............................................... 12
      2.1.2 Noordin Top Terrorist Dataset ............................ 14
   2.2 Social Network Analysis (SNA) .................................. 15
      2.2.1 Preliminaries .............................................. 17
      2.2.2 Implementation of SNA on Enron Email Communication Network ........................................ 19
      2.2.3 Implementation of Shortest Path Concept in Past Research ........................................ 21
   2.3 Scoring Important Nodes ......................................... 22
   2.4 Criminal Network Formations .................................... 23
   2.5 Software and Tools ............................................. 25

3 SPNSA Explorative Method on the Enron BCC Email Network 26
   3.1 Introduction .................................................... 26
   3.2 The Main Criteria for the Analysis ............................... 27
      3.2.1 Email Type Selection and Reasoning ....................... 28
      3.2.2 Statistical Analysis of Enron Emails with BCC Recipients ........................................ 29
   3.3 BCC Email Network Analysis ..................................... 31
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.1</td>
<td>The Algorithm</td>
</tr>
<tr>
<td>3.3.2</td>
<td>1-BCC criminal shortest paths network</td>
</tr>
<tr>
<td>3.3.3</td>
<td>2-BCC criminal shortest paths network</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Criminal shortest paths network analysis</td>
</tr>
<tr>
<td>3.4</td>
<td>Conclusion</td>
</tr>
<tr>
<td>4</td>
<td>SPNSA Validation and Comparison with Community Detection Algorithms</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>4.2</td>
<td>Comparison of SPNSA with existing community detection algorithms</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Enron email dataset analysis</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Criminals’ communication links in Netgraph and BCCNetgraph</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Identifying criminals’ community using community detection algorithms</td>
</tr>
<tr>
<td>4.3</td>
<td>Application of Shortest Path Network Search Algorithm</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Application of SPNSA on directed Netgraph and BCC Netgraph</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Application of SPNSA on the undirected Netgraph and BCC Netgraph</td>
</tr>
<tr>
<td>4.4</td>
<td>Crime investigation methods using SPNSA</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Extracting sub-networks using non-criminals</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Extracting subgraphs using random feed</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Extracting subgraphs using leave-one-out method</td>
</tr>
<tr>
<td>4.5</td>
<td>Conclusion</td>
</tr>
<tr>
<td>5</td>
<td>Shortest Paths Criminal Connection Analysis using Reliance Measure</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>5.2</td>
<td>Background</td>
</tr>
<tr>
<td>5.3</td>
<td>Preliminaries</td>
</tr>
<tr>
<td>5.4</td>
<td>The reliance measure</td>
</tr>
<tr>
<td>5.5</td>
<td>Comparison between different dependency techniques</td>
</tr>
<tr>
<td>5.6</td>
<td>Case Study 1: Enron Network</td>
</tr>
<tr>
<td>5.6.1</td>
<td>Node ranking for on the Enron ‘BCC’ network</td>
</tr>
<tr>
<td>5.6.2</td>
<td>Extracting a suspect’s reliance sub-network from the Enron Network using the reliance measure</td>
</tr>
<tr>
<td>5.7</td>
<td>Case Study 2: Noordin Top Terrorist Network</td>
</tr>
<tr>
<td>5.7.1</td>
<td>Node ranking for the Noordin Top terrorist network</td>
</tr>
<tr>
<td>5.8</td>
<td>Conclusion</td>
</tr>
<tr>
<td>6</td>
<td>Conclusion</td>
</tr>
</tbody>
</table>
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>(a) An undirected network. (b) A directed network</td>
<td>17</td>
</tr>
<tr>
<td>2.2</td>
<td>Most influential node’s position</td>
<td>19</td>
</tr>
<tr>
<td>2.3</td>
<td>Middle man node’s position</td>
<td>19</td>
</tr>
<tr>
<td>3.1</td>
<td>Number of BCC recipient percentage in emails</td>
<td>30</td>
</tr>
<tr>
<td>3.2</td>
<td>Number of BCC recipients in emails</td>
<td>30</td>
</tr>
<tr>
<td>3.3</td>
<td>BCC recipients in emails with more than 5 recipients</td>
<td>31</td>
</tr>
<tr>
<td>3.4</td>
<td>1-BCC criminals’ shortest paths network. All the criminals who occurred in this network, Andrew Fastow (686), Lea Fastow (11010), and Kevin Hannon (10068), are highlighted. The MM (16383 - Sara Shackleton) and the MI (19075 - Vince Kaminski) are indicated.</td>
<td>39</td>
</tr>
<tr>
<td>3.5</td>
<td>2-BCC criminals’ shortest paths network. The MM (16383 - Sara Shackleton) and the MI (19075 - Vince Kaminski) are indicated. All the criminals who occurred in this network, Andrew Fastow (686, 687), Lea Fastow (11010), Kevin Hannon (10068), Ben Glisan (1369), Kenneth Rice (9994) and Rex Shelby (15224), are highlighted.</td>
<td>42</td>
</tr>
<tr>
<td>3.6</td>
<td>Connection from A to D</td>
<td>42</td>
</tr>
<tr>
<td>4.1</td>
<td>The figures show that the degree distributions for both Netgraph (a) and BCC Netgraph (b) are heavy-tailed with many nodes being of low degree and some nodes being highly connected.</td>
<td>47</td>
</tr>
<tr>
<td>4.2</td>
<td>The figures above show the networks formed by using 1,2,3,4-neighbourhood (1,2,3,4-N) of Andrew Fastow (686) in the undirected BCC Netgraph respectively. In the 1-neighbourhood of Andrew Fastow only one other criminal was found, Ben Glisan (1369). The 2, 3, and 4- neighbourhood networks are clearly too dense to be able to identify other criminals easily.</td>
<td>53</td>
</tr>
</tbody>
</table>
4.3 Figures above shows the community numbered 5 (2195 nodes) using fast greedy algorithm and community numbered 36 (1254 nodes) using walktrap algorithm on undirected BCC Netgraph.

4.4 The shortest paths network formed using the directed Netgraph. All 7 criminals who existed in Netgraph were found. The network formed is sparse and criminals’ connections can be easily identified. This sub-network contains 30 nodes. The nodes highlighted in red represent the criminals with Net. ID from Table 4.1.

4.5 The shortest paths network formed using directed BCC Netgraph. 2 out of 10 criminals were lost. The network formed is sparse and criminals’ connections can be identified. The number of nodes in this sub-network is 55. The nodes highlighted in red represent the criminals with BCCNet. ID from Table 4.1.

4.6 The shortest paths network formed using undirected Netgraph. No criminal IDs were lost. The network formed is sparse and criminals’ connections can be identified. The size of this sub-network is 30 nodes. The nodes highlighted in red represent the criminals with Net. ID from Table 4.1.

4.7 The shortest paths network formed using undirected BCC Netgraph. No criminal IDs were lost. The network formed is sparse and criminals’ connections can be identified. This sub-network consists of a main component of 74 nodes and another small component of 3 nodes. The nodes highlighted in red represent the criminals with BCCNet. ID from Table 4.1.

4.8 Enron undirected BCC Netgraph shortest paths network using all top managers excluding criminals as algorithm feed. The nodes highlighted in red are all criminals.

4.9 Enron undirected BCC Netgraph shortest paths network using all finance managers as algorithm feed. The nodes (686, 687 and 1369) highlighted in red are finance managers who are also criminals. The other criminal who was found is Lea Fastow (11010).

4.10 Enron undirected BCC Netgraph shortest paths network when Michael Kopper (12708) is left out from the criminal feed list. The nodes highlighted in red are all criminals.

5.1 A small network. This network has 8 nodes and 8 edges. The floating-point numbers near to each node represent the trust value of source node 1 on each intermediate node \{2, 3, 5, 7\} in the path 1 → 2 → 3 → 5 → 7 → 8.
5.2 The result of applying the different dependency formulae and the reliance formula on the network shown in Figure 5.1. Here, for the purpose of comparison I choose the source as node 1 and take all shortest paths from this source node to all possible end nodes. The value of reliance of node 1 on other intermediate nodes is compared to the other dependency values.

5.3 A sample graph. This sample graph is used to compute the dependency of node 1 on other nodes.

5.4 The Enron finance manager BCC sub-network node ranking. The total reliance (Equation 5.8) of all finance managers (686, 687, 16929, 1369, 15077, 8071) on each $v$ in the Enron Finance Manager BCC sub-network is used for this comparison. The nodes are ranked based on the descending order of betweenness centrality (Brandes’ (Equation 5.3) and Geisberger et. al (Equation 5.4)) values. There is a clear difference in node ranking between the betweenness centrality measures of Brandes and Geisberger et al. (there is no difference between them) and the total reliance. For example, node 2473 is more heavily relied on by the finance managers than 686, where as both the betweenness centrality measures rank them the other way around.

5.5 The Enron finance manager BCC sub-network node ranking. The total reliance of all finance managers on each $v$ in the Enron Finance Manager BCC sub-network is used for this comparison. The nodes are ranked based on the descending order of PageRank values. There is a clear difference in node ranking using these three different methods. For example, nodes 3945, 3973, 10917 and 7974 have the same ranking value using Markov centrality and PageRank methods but these nodes are ranked differently by the reliance value.

5.6 The Enron manager BCC sub-network node ranking. Three different methods that is betweenness centrality by Brandes (Equation 5.3), by Geisberger et al. (Equation 5.4) and reliance value using reliance measure (Equation 5.8) are used to show the difference between node ranking. Note that only the nodes with significant reliance value are displayed here.
The Enron manager BCC sub-network node ranking. The total reliance of all managers on each \( v \) in the Enron Manager BCC sub-network is used for this comparison. The nodes are ranked based on the descending order of PageRank values. There is a clear difference in node ranking using the three different methods. For example PageRank, Markov centrality and the reliance measure ranks node 19075, 6673 and 17697 as having the highest value respectively.

The Enron Finance Manager ‘BCC’ sub-network formed using the SPNSA and only emails that have recipients in BCC field. The finance managers [6] that are used as the algorithm feed are given in Table 4.10. The finance managers who are captured in this shortest path network are highlighted in red. This is an undirected graph with 30 nodes and 53 edges.

Second phase Finance Manager BCC sub-network: This sub-network is formed using the nodes (1369, 2473, 15406, 686 and 19075). These nodes (highlighted in red) are the ones that the finance managers rely on the most. This is an undirected graph and has 19 nodes and 32 edges.

Enron finance managers’ reliance ‘BCC’ sub-network: The figure above shows a sub-network formed using our reliance measure algorithm repetitively. The finance managers are used as the first phase feed to the SPNSA, followed by the next set of nodes that are of the highest reliance to the finance managers. These nodes are called the finance managers’ persons of interest and they are, in turn, used as the third phase feed and so on. The process stops when the same nodes obtain the highest reliance value or when there is no out-component from the persons of interest, resulting finally in a hierarchal network as in this figure.

The Enron Manager BCC sub-network formed using the SPNSA and only emails that have recipients in BCC fields only. The feed for the algorithm are all the Enron managers [6] including finance managers; (see Table 4.9 and 4.10). Here, I assume that I do not know the criminal list and I use all managers even though some happen to appear in the criminal list [23]. The managers who are captured in this shortest path network are highlighted in red. This is an undirected graph and has 121 nodes and 314 edges.

Enron managers’ reliance ‘BCC’ sub-network formed using the nodes that all top managers and finance managers from the BCC network rely on the most.
5.13 Enron finance managers’ reliance ‘TO/CC’ sub-network: The figure shows a sub-network formed using the nodes that all finance managers in ‘TO/CC’ network rely on the most. The finance managers [6] that are used as the algorithm feed are given in Table 4.10. Note that some employees’ email addresses that exist in the ‘BCC’ network do not occur in the ‘TO/CC’ network, for example email address (andrew.fastow@ljminvestments.com) that belongs to Andrew Fastow occurs only in the ‘BCC’ network. Thus, the use of separate ID for the nodes in both the networks.

5.14 Enron managers’ reliance ‘TO/CC’ sub-network: The figure shows a sub-network formed using the nodes that all managers (including the finance managers) in ‘TO/CC’ network rely on the most.

5.15 The total highest reliance of all nodes in the terrorist-friendship and terrorist-classmate sub-network on each $v$ is used for this comparison. For figures a and b, the nodes are ranked based on the descending order of Brandes’ betweenness centrality values and the reliance and betweenness values are shown for different subsets of terrorists, that vary in size and membership. For each graph, terrorists with zero reliance and ranked by betweenness (Brandes) values as below the terrorist with the lowest reliance, are not included.

5.16 The total highest reliance of all nodes in the terrorist-friendship and terrorist-classmate sub-network on each $v$ is used for this comparison. For figures a and b, the nodes are ranked based on the descending order of PageRank scores and the reliance, Markov centrality and PageRank scores are shown for different subsets of terrorists, that vary in size and membership. For each graph, terrorists with zero reliance and ranked by PageRank scores as below the terrorist with the lowest reliance, are not included.
# List of Tables

2.1 Enron money laundering criminals .......................................... 14

3.1 Enron money laundering criminals .......................................... 28

3.2 Shortest paths in Andrew Fastow's ego sub-network in the 1-BCC network ................................................................. 37

3.3 Shortest paths in Lea Fastow's ego sub-network in the 1-BCC network ................................................................. 38

3.4 Shortest paths in Kevin Hannon's ego sub-network in the 1-BCC network ................................................................. 38

3.5 Shortest paths in Andrew Fastow's sub-network ................................ 40

3.6 Shortest paths in criminal’s 2-BCC sub-network \(N_{EC_i+1...EC_n}\) ................................................................. 41

4.1 Enron money laundering criminals .......................................... 48

4.2 Shortest path length from criminal to criminal in the directed Netgraph ................................................................. 49

4.3 Shortest path length from criminal to criminal in the directed BCC Netgraph ................................................................. 49

4.4 Shortest path length from criminal to criminal in the undirected Netgraph ................................................................. 50

4.5 Shortest path length from criminal to criminal in the undirected BCC Netgraph ................................................................. 51

4.6 Enron money laundering criminals’ degree ...................................... 52

4.7 Criminals Found in Different Communities ........................................ 55

4.8 Comparison between results found using R \textit{igraph} community detection algorithms and SPNSA ................................................................. 60

4.9 Enron managers ................................................................. 62

4.10 Enron finance managers ................................................................. 63

4.11 Leave-\textit{C}_i-out Method ................................................................. 66

5.1 Nodes that the finance managers rely on the most in the Enron Finance Manager BCC sub-network ................................................................. 83

A.1 Path connecting Andrew Fastow to other nodes ........................................ 94

A.2 Path connecting Kevin Hannon to other nodes ........................................ 95

A.3 Path connecting Lea Fastow to other nodes ........................................ 96

A.4 Frequency of node connecting Andrew Fastow to other nodes ................................................................. 97
Summary

Retrieving criminal ties and mining evidence from an organised crime incident, for example money laundering, has been a difficult task for crime investigators due to the involvement of different groups of people and their complex relationships. Extracting the criminal association from enormous amounts of raw data and representing them explicitly is tedious and time consuming. A study of the complex networks literature reveals that graph-based detection methods have not, as yet, been used for money laundering detection. In this research, I explore the use of complex network analysis to identify the money laundering criminals’ communication associations, that is, the important people who communicate between known criminals and the reliance of the known criminals on the other individuals in a communication path.

For this purpose I use the publicly available Enron email database that happens to contain the communications of 10 criminals who were convicted of money laundering crime. I show that my new shortest paths network search algorithm (SPNSA) combining shortest paths and network centrality measures is better able to isolate and identify criminals’ connections when compared with existing community detection algorithms and k-neighbourhood detection. The SPNSA is validated using three different scenarios: (i) when the investigator knows all the criminals, (ii) when the investigator fails to detect one of the criminals and (iii) when the investigator is at the starting stage and doesn’t have any information about the criminals, but suspects a crime is occurring. In each of these scenarios, the criminal network graphs formed are small and sparse hence suitable for further investigation. The algorithm manages to extract a criminal network with a minimum of 4 criminals when none of the criminals’ information is known. Different algorithm feeds were used in the different investigation scenarios discussed above. I show that SPNSA using a relevant algorithm feed that is related to a crime incident, yields the best result. This is validated by applying random feed to the SPNSA, and the result shows the probability of retrieving criminals using a random feed is very low; only 1% in 1000 graphs. In another scenario, in 5 out of 9 cases SPNSA was successful in retrieving the left out criminal in the resulting sub-network.

My research starts with isolating emails with ‘BCC’ recipients with a minimum of two recipients bcc-ed. ‘BCC’ recipients are inherently secretive and the email connections imply a trust relationship between sender and ‘BCC’ recipients. There are no studies on the usage of only those emails that have ‘BCC’ recipients to form a trust network, which leads me to analyse the ‘BCC’ email group separately. SPNSA is able to identify the group of criminals and their active intermediaries in this ‘BCC’ trust network. Corroborating this information with published information about the crimes that led to the collapse of Enron yields the discovery of persons of interest that were hidden between criminals, and could have contributed to the money laundering activity. For
validation, larger email datasets that comprise of all ‘BCC’ and ‘TO/CC’ email transactions are used. On comparison with existing community detection algorithms, SPNSA is found to perform much better with regards to isolating the sub-networks that contain criminals.

Finally I have adapted the betweenness centrality measure to form a reliance measure. This measure calculates the reliance of a criminal on an intermediate node and ranks the importance level of each intermediate node based on this reliability value. SPNSA in combination with the reliance measure could be used as primary investigation tools to investigate connections between criminals in a complex network.
Chapter 1

INTRODUCTION

1.1 Overview

The aim of this chapter is to give the background and the purpose of this research work. Section 1.2 introduces network science and its applications. Money laundering system detection and its inherent problems are explained in section 1.3. The motivation and scope of this research is given in section 1.4. Section 1.5 outlines the objectives and the contributions of this research. Finally, section 1.6 gives the structure of the rest of this thesis.

1.2 Network Science

Network Science involves the study of complex networks in different fields such as telecommunications, science, computer science, sociology, etc. One of the main methods applied to the study of complex networks is graph theory. A first practical application, the Seven Bridges of Konigsberg problem was solved using graph theory by Leonhard Euler in 1736. Graph theory has been applied mainly to represent a complex network in the form of nodes and edges. For example, web pages of the world wide web can be designated as nodes and the hyperlinks that connect different web pages as edges [52]. Similarly with internet connections, the routers or computers are the nodes and the physical connections between them are the edges [34]. In an ecological network, the species form the nodes and the predator-prey relationships are shown as edges [10]. In another example, human relationships are represented as links between nodes; humans being the nodes and the edges between them created when they communicate or socialise with each other [140]. The terms used for the nodes and edges differ in different areas of study. While in computer science it is usual to refer to nodes and links, in sociology it is more common to refer to these as actors and ties respectively, and in mathematics, as vertices and edges [115].
The focus of this thesis is on social network analysis, the study of the relationships between people, and applies graph theory and a systematic study of the nodes’ connections. It involves the application of certain parameters and formulas to create algorithms. These algorithms serve different purposes for example, searching for clusters or cliques [22], finding community structures and topologies of the network [43], calculating the central values [121], etc. The study also extends to the detection of individual or group behaviours. There are two types of social network analyses; sociocentric and egocentric network analysis [115]. Egocentric network analysis examines a known individual’s network (ego network) connections. On the other hand, sociocentric network analysis focuses on large groups of people where all the information related to the members of the network are obtained beforehand. Sociocentric analysis quantifies the relationships and impact of the relationships’ patterns between groups. In my research, I have used both egocentric and sociocentric analysis.

The first stage of network analysis is the element level analysis. This often starts with looking at the number of edges to and from an individual, also called ego, as edges (links) can be used to define many different characteristics of an individual. All individuals that directly link to the ego, or node, of interest, are called its neighbours. This part of network analysis is called element level analysis. The degree of a node is the number of its neighbours, and a node has highest degree centrality if it has the highest number of neighbours. Thus, degree centrality gives a very simple ranking of individuals based on degree. There are other types of centrality measures for example, closeness centrality, betweenness centrality, eigenvector centrality, etc [64]. Nodes with high degrees are said to have strong connections and are often called hubs. Nodes that have fewer neighbours are said to have weak ties to the network. An individual who connects two parts of a network that may not otherwise be connected is called a broker [115]. There are many other properties of a network that depend on the edges connecting distinct nodes. The distance between a pair of nodes is given by the length of the shortest path that links the pair. The average path length between any two nodes in a network is called the diameter of the network. Element level analysis involves all of the properties above and is suitable to be applied during egocentric network analysis.

The second stage of analysis is the group-level analysis and comes under the umbrella of sociocentric analysis. Here, one can do many things, such as network clustering, the method by which individuals with similar characteristics are grouped together [115]. Network clustering is conducted using similarity measures that can be defined based on the function of each node, for example, the set of journal papers about cancer published by a group of academics could be said to belong to the cancer research collaboration network [116]. It could be also considered as a cluster
or a community within the network of all research papers. There are different types of clustering algorithms for example, k-means algorithm where the size of the partition is known; agglomerative algorithms such as hierarchical clustering that merge subnetworks based on a defined distance; divisive algorithms that divide a sub-network based on object dissimilarity, etc. There is also data clustering, for instance, random walks that group paths based on the data flow within a network [115].

The third stage of analysis is the network-level analysis that focuses on the topological properties of the network based on the position of the nodes [115]. For example, given a node that is connected to two other nodes, the probability that these two nodes are linked to each other to create a connected triangle is called the clustering coefficient. Network analysis can also be based on nodes’ degree distributions. When there are a small number of nodes with high degree and a large number of nodes with low degree we get a degree distribution that is highly skewed to the right, resembling a power law distribution. A network with this property is called a scale-free network. Exploring the static and dynamic properties of a network are more recent network analysis methods. While the static properties are based on the structure of a network, the dynamic properties explain the changes in the network structure over time. Scientists who study social networks use various network properties to develop tools and techniques for different practical applications [70]. In this thesis, I will use a combination of some of these analyses and properties to explore a money laundering network.

1.3 Money Laundering Network Detection and the Inherent Problems

Money laundering is a technique used by criminals to wash off the traces of the illegal source of the money. Money laundering is divided into three stages: placement, layering and integration [13]. Placement involves transporting and depositing illegally obtained money into various domestic banks, informal financial institutions or international financial institutions. Layering is a complex series of transactions that involves the transferring of money from one bank or account to another through a corporation or trust with the aim of moving the money as far away from its source as possible. At this point the illegal funds will have become integrated with legitimate funds and are considered ready for investments, savings or expenditure [13]. The anti-money laundering (AML) analyst investigating a suspected money laundering activity at this stage, would need to do a logical analysis to be able to differentiate between the legitimate and illegal transaction patterns, something that is difficult to automate [13].
It is vital to detect this illegal transport of money in order to prevent the crime evolving behind it. A common way of detecting money laundering is wire transfer screening or wire transfer monitoring that applies a knowledge-based system, which needs knowledge acquisition and that, in turn, needs human experts. A knowledge-based system uses a database containing original records and determines decisions based on codification of rules for example the IF-THEN rules. Even though human judgment is the best we have for differentiating between the legitimate and illegitimate wire transfers from large amounts of data, it is prone to false positive errors [108].

Another common way of detecting money laundering is by using data mining methods [166; 92; 144; 66]. This process requires some pre-indicators, for example, the persons’ names or account numbers to identify whether the criminal activity has occurred. Persons’ names or account numbers are clustered using behavioural variables. Accounts that deviate from normal behaviour or transaction patterns are categorized as illegal. In some cases, it is difficult to distinguish between normal and illicit activity without these pre-indicators. In 2003, Zhang et al. [166] introduced correlation and histogram analysis to recognize money laundering patterns. However, given large numbers of transactions involving various individuals, they found it difficult to detect suspicious behaviour when there were very few peaks or no “peak hills” [166] in the histogram. In 2005, Tang and Yin [144] attempted to detect unusual customer behaviour, but the heterogeneous nature of bank transaction records caused difficulty in selecting suitable parameters, leading to uncertainty in the experimental results. In 2009, Le-Khac et al. [92] sought to detect suspicious and non suspicious cases within investment transaction analysis. Their method unfortunately could not differentiate between suspicious activity and peak business hours activity and also could not be applied to large datasets [92]. Thus it has been very difficult to determine a money laundering community by just using statistical values or pre-defined anomaly parameters to guide human judgments.

1.4 Motivation and Research Scope

Money laundering criminals have their own modus operandi. As discussed above, criminals use financial institutions as instruments to transfer illegal funds and the transactions tend to mimic genuine ones [153; 13] making it difficult to detect them using transaction data mining and knowledge-based systems. My research project aims to build a new approach to detecting a money laundering community using social network analysis. I focus on the element-level analysis and group-level analysis of a network, representing a suspect’s community as a graph that consists of persons as nodes and relationships as edges. Since it was difficult to obtain actual financial transaction data from banks or other financial institutions due to a strict privacy policy I first
use the publicly available Enron email database [44] and focus on the static properties of these
email transactions. I use the Enron email dataset, for two reasons; it is publicly available and it
happens to contain 10 money laundering criminals [102, 23].

Many researchers from different backgrounds have worked on various aspects of the Enron data
to detect important nodes [160, 138], and business processes [138], identify manager-subordinate
relationships, [54] and explore organisational behaviour [55]. Besides this, email transaction short-
est paths between Enron employees have also been studied by various researchers [146, 162]. In an
email network, not all messages from one criminal to another may contain information regarding
an illegal event. If we take the scenario of money transactions, a person can transfer money to
another person for genuine reasons or for illegal purposes. In the context of money transfer, the
illegal money transfer cannot be detected by just using the amount transferred as in most cases
this amount will be below the threshold. Similarly, email messages might not have the same ID or
sequence of illegal contents to be grouped as one community. This was one of the problems that
I encountered while exploring the email dataset.

Based on the nature of money laundering, often the intermediary money carrier or money
porter may not know the initial senders and final receivers in the chain [13, 134]. Bearing this in
mind, I assume that the information passed on from one criminal to another may not contain a
direct message to the other and the interconnected entities may not know that they are actually
delivering suspicious messages related to money laundering or any other crime. Consequently, I
choose to ignore the content of the various emails being exchanged between the criminals and
propose a very different way to start the building of a criminal sub-network, by considering the
type of emails based on recipient fields. I explore the use of social network analysis because this
methodology has been used to show relationships among entities through graphs and networks
[91, 60, 161]. Techniques such as link analysis [60, 161] are essential to show relationships
between objects in a money laundering network and to integrate different sources of information;
people, bank accounts, business, wire transfers or cash deposits.

In general, grouping people to form a network to aid a criminal investigation is a difficult task.
Past research has found that the extraction of criminal associations from raw data and representing
them in a network graph is tedious and time consuming. Nadji et al. [111] take passive connections
from a network attack database and manually create edges between IP addresses, while Krebs [91]
collects data about 9-11 terrorist attacks from press articles and forms the relationships between
them manually according to whether the terrorists lived at the same place, entered the same school,
traveled or had meetings together. Thus, there is a need for automatically grouping and forming
a network to ease a criminal investigation process. The existing community detection algorithms
for example the k-neighbourhood, Fastgreedy, Walktrap and Leading Eigenvector algorithms only show the number of communities formed. Using these existing community detection algorithms, an investigator still needs to manually search for the community to which a criminal belongs. Further, retrieving the neighbourhood subgraphs of a criminal can result in dense networks. In this research I combine network centrality measures and shortest paths to create an algorithm called the shortest paths network search algorithm (SPNSA) [102] that can automatically identify Enron money laundering criminals’ community from a static email network. The network formed becomes a primary source of investigation.

Another aim of this research is to rank the important nodes using sociocentric analysis. Many different centrality measures have been used by researchers in the past, to identify nodes of importance [84; 103]. Suspicious behaviour in a communication network can be highly dependent on the number of times a person is contacted, the number of meetings as well as the location of this person within the network. Centrality measures alone are insufficient to identify suspicious nodes. Instead of pin-pointing the influential nodes by calculating the centrality values of all the nodes in a network [74; 109], we start with a group of suspects. In [102], I used a set of initial suspects and then identified the nodes important to these suspects as those having the highest frequency connecting a suspect to all other nodes in the network; I show this experiment and the results in Chapter 3. Following that, in Chapter 4, I compare my results with those obtained from community detection algorithms. Later, in Chapter 5, I explore pair-wise dependency and enhance our method in [102] to identify nodes that a criminal investigator should investigate further.

There is some past research that has found intermediate nodes as important nodes. White and Smyth [157] adapt several graph theory and social network tools including the k-short Node-Disjoint Paths to measure the relative importance of a node in a graph given a set of nodes of interest using the average number of hops needed to reach a target node. A Markov walk-based function has also been proposed to assign scores to each node and enhance the scoring function with relevant feedback information [1]. I have adapted Brandes’ algorithm [20] and the technique by Geisberger et al. [67] to form a reliance measure. This measure calculates the reliance of a criminal on an intermediate node and ranks the importance level of each intermediate node based on this reliability value. I focus on the nodes with the highest reliance that occur between the criminals (or suspects) and other nodes. These nodes become a set of new suspicious nodes and are used iteratively as a feed for the SPNSA to find new communities. The network search stops when there is no longer any out-component found from the node under suspicion to other nodes.

While I started this research using the Enron dataset, I later applied my analysis to the Noordin Top terrorist network [57] to illustrate that SPNSA could be used on any criminal network.
Very different from the Enron email network, the Noordin Top terrorist network consists of different types of connections between the same nodes. The first type of relationships consist of terrorist type affiliations such as terrorist organisations joined, educational institutions attended, business, religious institution, etc. The second group includes relationships between people such as classmates, kin, friends while the third group shows individuals that provided logistical support or participated in training events, terrorist operations, and meetings. My research involves the terrorist-people relationships. Butt et al. [31], predict some possible key players in this dataset by using particular centrality measures (degree, betweenness, closeness centrality and eigenvector) along with some classifying techniques such as k-nearest neighbours (kNN), etc. Liebig and Rao [95] use a bipartite clustering coefficient to find important nodes in this multi structured terrorist network. I use the shortest paths network search algorithm and the reliance formula to extract an investigative network first and then to identify source-intermediate node rankings that could aid further investigation.

1.5 Thesis Objectives and Contributions

The goal of this research is to use complex network analysis to build an algorithm to extract criminals’ or suspects’ connections from a large dataset, obtaining a sub-network that is feasible for criminal investigation. This is followed by using other concepts from network science together with the reliance model to rank new suspects into a hierarchical network graph.

1.5.1 Objectives

The objectives of this research work are:

- To use complex network analysis to identify a sub-network of criminals or suspects.
- To develop an algorithm to extract the relationships between money laundering criminals or terrorists.
- To evaluate the effectiveness of the proposed algorithm and compare with existing community detection algorithms.
- To identify important nodes based on a criminal or suspect’s reliance ranking and form a hierarchical criminal reliability sub-network for investigation.

1.5.2 Contributions

The contributions of this research study are as follows:
• A new shortest paths network search algorithm (SPNSA) that incorporates network central-ity measures and shortest paths to extract a criminal’s network of trust. Different investiga-tive methods are applied to the algorithm and the result shows that the algorithm produces sparse and small sub-networks that could feasibly identify a list of persons and relationships for further investigation.

• A method of identifying active intermediate nodes to show the criminals’ or suspects’ connection patterns.

• A comparison method to show the efficacy of SPNSA. The algorithm begins with an ‘algo-rithm feed’, a small subset of source nodes of particular interest, and builds an investigative sub-network. The algorithm feed may consist of known criminals or suspects, or persons of influence as this yields better results as proven through SPNSA random feed validation. SPNSA is then compared to existing community detection algorithms and k-Neighbourhood approach.

• A novel reliance formula to calculate the dependency of a source node on an intermediate node. The reliance formula is used as a tool to build a source-intermediate node reliance measure algorithm that can measure the suspicious sub-network that contains the criminal or suspects and their associates. This measure calculates the reliance of a criminal or suspect on an intermediate node and ranks the importance level of each intermediate node based on this reliability value.

• A comparison between the node ranking methods using existing pair-wise dependency mea-sures, Markov centrality and PageRank to the reliance measure. The intermediate nodes with highest reliance value are gathered and called the crime priority nodes. The ranking results show that a criminal investigation using the reliance measure, will lead to a different prioritisation in terms of possible people to investigate.

1.6 Structure of this thesis

The structure of the rest of the thesis is as follows:

Chapter 2 gives a background and a review of past literature. It includes the history of Enron, focusing on the money laundering crime. The Enron communication network and the Noordin Top terrorist network datasets are described in more detail. Preliminary information used to build the shortest paths network search algorithm (SPNSA) is included, followed by a literature survey on the methods used in the past to identify important nodes and the difficulties
encountered in criminal network formations. Literature more relevant to each chapter is provided at the beginning of that chapter.

**Chapter 3** introduces the shortest paths network search algorithm (SPNSA) and its implementation using a particular subset of the Enron data. Email transactions have different types of recipients. This chapter gives the reason for selecting the bcc-ed recipients’ email transactions. It then gives the results of the statistical analysis applied to the Enron emails with a certain number of BCC recipients. A list of people who were convicted as Enron money laundering criminals are used as the algorithm feed to extract the criminal shortest paths network.

**Chapter 4** validates and compares the performance of the SPNSA with the existing community detection algorithms. Here larger email datasets comprising of all ‘BCC’ in the first instance and ‘TO/CC’ email transactions in the second, are used. The SPNSA is validated using two different criminal investigation methods: extracting sub-networks using non-criminals or first suspects and extracting sub-networks using a leave-one-out method. Another validation is conducted by applying random feed in the algorithm to create 1000 random graphs. The criminals extracted in each case are checked. The number of criminals that could be extracted using random nodes as opposed to known suspects as the algorithm feed are compared.

**Chapter 5** introduces a novel source-intermediate node reliance measure. The reliance values are used to rank the importance of intermediate nodes that result in the identification of persons of interest. The reliance node ranking is compared to the ranking using the two other dependency values given by Brandes [20] and Geisberger et al. [67] as well as comparing the rankings that are obtained using PageRank and Markov centrality algorithms. A reliance sub-network that could be used to analyse the connections between criminal or suspects and their intermediate nodes is extracted.

**Chapter 6** summarises the methods and contributions of this research.
Chapter 2

BACKGROUND

A criminal investigation into patterns of correspondence or social contacts between possible suspects, represented most naturally as networks, faces considerable difficulties in detecting criminals due to the size and complexity of the networks. This chapter reviews published studies that have used complex network analysis to explore the publicly available Enron email network where email addresses correspond to nodes and sent e-mails correspond to edges. In conjunction with this literature survey, I describe the importance of the Enron data and the history of Enron. I give a brief outline of the money laundering crime and the employees convicted of this crime in section 2.1. In addition to the Enron data, the methods developed in this thesis are applied to the Noordin Top terrorist network, the origins of which I describe in this chapter. I introduce concepts from social network analysis, and in particular two centrality measures that are used to build the core algorithm of this thesis. Section 2.3 reviews different methods for scoring or ranking nodes using a pair-wise dependency approach.

2.1 Datasets

2.1.1 Enron Data

Researchers from different backgrounds have worked on many aspects of the publicly available Enron data for a variety of reasons including the detection of organisational structure and social standing [126], detection of important nodes [160] and business processes [138], identification of manager-subordinate relationships [54], exploration of organisational behaviour [55] and identification of hierarchy in the organisation with respect to the salary structure [40].

In July 1985, the Enron Corporation was created by Kenneth Lay through merging Houston Natural Gas, a utility company and Omaha-based InterNorth, a gas pipeline company. At that time, Kenneth Lay became the Chief Executive Officer (CEO) and Chairman of Enron [26]. In
1988, Enron management took the opportunity to open branches in other countries, expanding its business and becoming the middleman for energy trading between the United Kingdom, Europe, South America and India. Enron’s hidden aim was to earn commissions through regulating the fluctuating energy price in the market which was due to increasing competition among old and new suppliers. In 1989, Enron’s Gas Bank helped Enron Corp. emerge from a business of piping gas to secure a prominent place between suppliers and consumers in energy trading. In the 1990s, Enron became an exceptionally large player in the United States’ energy market. Later, in 1998, Andrew Fastow was employed as the Chief Financial Officer of Enron.

On 29th November 1999, the company introduced the Enron online transaction system. This system had a better view of the current prices of US power products and US and Canadian natural gas products, crude oil and refined products, coal, emission credits, weather derivatives and pulp and paper from traders. It also made marketing raw materials more efficient. Within a very short time, this internet-based global transaction system resulted in Enron becoming the sixth-largest energy company in the world.

In February 2001, Jeffery Skilling took over as CEO for 6 months until Kenneth Lay resumed office in September 2001. Unexpectedly, in October 2001, Enron reported a loss of US$618 million and at the same time the Chief Financial Officer, Andrew Fastow was sacked by Enron. As reported in the BBC news on 4th February, 2002, in August 2001, Kenneth Lay was alerted to the accounting irregularities that lead the Security and Exchange Commission of the United States to start an investigation on various internal business and financial activities including the partnerships set up by Andrew Fastow. This was followed in early January 2002 by other criminal investigations on many Enron executives.

**The Crime**

Andrew Fastow was convicted of securities fraud, mail and wire fraud, money laundering, conspiracy, obstruction of justice, falsifying books and records, insider trading and filing false personal tax returns. Here, I focus on three crimes that Andrew Fastow was convicted of. The first, that he planned and designed a complex web pattern of offshore partnerships and questionable accounting practices. The second, he illegally acquired funds for himself and his family; U.S authorities seized up to USD$23 million United States’ assets from Andrew Fastow and his family members. The report by Thomsen and Clark from Division of Enforcement, U.S. Securities and Exchange Commission shows that the cash came from illegal business transactions that Andrew Fastow arranged. The third crime was that Andrew Fastow was involved in covert side deals and earned money through sham transactions.
A sham transaction is a deceptive transaction used for evading tax liability [150]. It is done through illusory contracts when an employer causes an employee to become an independent contractor with the intention of using the contract for illegal purposes. The employee can only follow the instructions but can’t make any decisions [12]. Examples of the sham transactions that feature money laundering mechanisms include under-declaration of imported goods, overbilling, dummy business, false contracts or contracts with misleading information and underestimated prices [94; 130; 56]. In this research, the findings about Andrew Fastow and his involvement in sham transactions [149; 147] are a motivation for starting the money laundering crime investigation by first identifying his clique from the Enron email network.

Other Employees Charged with Money Laundering Crime

The article by Brickey [23] lists major corporate fraud prosecutions from March 2002 till August 2003. Through this article, I obtained the criminal charges against the Enron employees. Besides Andrew Fastow, there are 9 other criminals [23] who were reported to be involved in the money laundering case. Table 2.1 shows the money laundering criminals.

<table>
<thead>
<tr>
<th>Name</th>
<th>Designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrew Fastow</td>
<td>Enron Chief Financial Officer</td>
</tr>
<tr>
<td>Lea Fastow</td>
<td>Enron Director and Assistant Treasurer of Corporate Finance</td>
</tr>
<tr>
<td>Kevin Hannon</td>
<td>Enron Broadband Services COO</td>
</tr>
<tr>
<td>Kenneth Rice</td>
<td>Enron Broadband Services Chairman and CEO</td>
</tr>
<tr>
<td>Rex Shelby</td>
<td>Enron Broadband Senior Vice President of Engineering Operations</td>
</tr>
<tr>
<td>A. Khan</td>
<td>NewCom President, CEO, and Chairman of the Board</td>
</tr>
<tr>
<td>Michael Kopper</td>
<td>Enron Managing Director</td>
</tr>
<tr>
<td>Ben Glisan</td>
<td>Enron Treasurer and Former Andersen Accountant</td>
</tr>
<tr>
<td>Joe Hirko</td>
<td>Enron Broadband Services CEO</td>
</tr>
<tr>
<td>S. Yaeger</td>
<td>Enron Broadband Services Vice President of Strategic Development</td>
</tr>
</tbody>
</table>

Table 2.1 shows the list of criminals who were involved in Enron money laundering crime [23; 44] and their designation.

2.1.2 Noordin Top Terrorist Dataset

A small dataset, the Noordin Top terrorist network [57] is also used to test the shortest paths network search algorithm application and to analyse results from the reliance formula. The Noordin Top terrorist dataset [57] consists of different types of connections. Its structure is different
from the Enron dataset. The connections between terrorists in the Noordin Top terrorist dataset are shown in binary format whereas the Enron email dataset consists of a collection of senders’ and recipients’ email addresses. Nodes in the terrorist network represent people whereas nodes represent email addresses in the Enron email network. The size of the terrorist network is smaller than the Enron dataset and both datasets are publicly available.

The first group of Noordin Top terrorist connections gives the terrorists’ affiliations such as terrorist organisations, educational institutions, business and religious institution. The second group contains relationship information such as classmates, kin, friends and the third group comprises individuals that provided logistical support or participated in training events, terrorist operations, and meetings.

For my thesis, I take 7 different subsets from this dataset. These are terrorist-training events, terrorist-education, terrorist-attack operations, terrorist-friendship, terrorist-soulmates, terrorist-kinship and terrorist-classmates. I rank the terrorists using source-intermediate reliance value by applying the shortest paths network search algorithm and the reliance formula.

In the next section, I discuss the importance of social network analysis and the two important measures that are used to form the shortest paths network search algorithm (SPNSA).

### 2.2 Social Network Analysis (SNA)

A network consists of nodes that are connected through edges. Basically, entities in the real world are referred to as the nodes, for example, employees of an organisation, assets in a company, books in a library, houses in one residential area, universities or schools in one town, etc. The nodes can also represent entities’ common features. The edges between them depict the communication or multiple relationships among them such as email sender-receiver, phone call caller-receiver, product seller-buyer, etc. The combination of all entities and edges of a specific functional group is called a network. Social network analysis is regularly used to investigate and view communities as networks of individual relationships that people develop in their daily lives.

The multiple connections of criminals and the intermediaries in the money laundering scheme gave me the idea of exploring each node’s central position and the connections of the central nodes to other members of the network.

There are a number of centrality measures used in social network studies to emphasize the importance of a node. A node that has the highest number of neighbours or the most connections with other nodes is the central node in a network. This measure is called degree centrality. Eigenvector centrality is an extension of degree centrality where a node will have a high eigenvector value if the node has high degree and is connected to other nodes that also have high
Katz centrality gives free degree to the isolated nodes by adding an extra parameter value in the centrality formula. So, when the nodes with null centrality value are connected to other nodes, all nodes get the advantage of being measured, their level of importance increases; if the node is linked from other highly linked nodes or the node itself becomes highly linked. Katz centrality is different from eigenvector centrality as in the former even the unconnected nodes can obtain a centrality value depending on the extra parameter value given to these nodes [115]. Two other types of centrality measures, closeness and betweenness centrality are based on shortest paths. Closeness centrality is measured based on the mean distance of a node from all other nodes in a network. It is the inverse of the sum of the shortest distances between each node and every other node in the network [115]. Betweenness centrality measures the centrality of a node \( v \) by taking the fraction of all possible shortest paths between pairs of nodes that pass through node \( v \). A node that occurs the most number of times through the shortest paths between all sources and destinations is called the node with the highest betweenness centrality.

Centrality measures have been used to detect influential nodes in a network. For example, Maiya and Berger-Wolf [103] have used a crawling algorithm to identify critical individuals in a co-authorship network. The crawling algorithm picks one random node and induces a subgraph of its immediate neighbours. A frequently visited node from the subgraph is also a node with high eigenvector centrality. Two other centrality measures, betweenness and closeness centrality were also considered by Maiya and Berger-Wolf [103]. A node that has the most diverse neighbours has the highest betweenness centrality and the one that is just one hop away from the other nodes represents highest closeness [103]. Hansler [74] in his project on analysing an email network has used the page rank algorithm to calculate eigenvector centrality values with an important node being picked based on the highest eigenvalue. This node is also called a hub. Kayes et al. [84] use multiple centrality measures based on a blogger’s position in the network to rank the influential bloggers. Nankani and Simoff [112] use betweenness, eigenvector and closeness centrality to improve influence prediction. Influence of an actor is based on the actor’s role and his or her central position. For each predictive model, there were ranks assigned to the node’s importance. The ranking values of the important nodes are used to improve the predictive accuracy of a given dataset [112]. Kiss et al. [87] have mentioned that the influence of a customer depends on the ability of the customer to pass a message to the entire network. This is calculated using degree distribution of customers for both in-degree and out-degree. A node with high out-degree is noted to be the direct source of information. Memon et al. [107] aim to find the hierarchy in a terrorist network by first identifying the parent node. The parent node has maximum neighbours and the ordering of the remaining nodes is based on traversing all the nodes according to the number of
neighbours. A node that has higher degree is always placed higher than another with lower degree and the link moves from the higher degree node to the node below. This method can help to give an early alert on the terrorist connections but still needs expert intervention. In this thesis, I adapt the two centrality measures; betweenness centrality and eigenvector centrality to identify the two main identities. I build an algorithm that combines shortest paths and these two centrality measures to identify connections of and between criminals if they are positioned in this group.

2.2.1 Preliminaries

This section includes the graph-theoretic terminology [72] for defining the shortest paths and the two centrality measures that are most important to the work in this thesis.

2.2.1.1 Terminology of Shortest Path

It is often important to identify the sequence of links and nodes that a message passes through from source to destination. This requires me to define a path and then, to identify the shortest of such paths. Below are the definitions of network path and shortest path.

Definition 1. : Paths

A path consists of a sequence of nodes connected by edges [113]. For example \( \{v_1, e, v_2\} \) represents a path between two nodes \( v_1 \) and \( v_2 \) with \( e_1 \) being the edge between \( v_1 \) and \( v_2 \) [113].

A node is also called a vertex and it represents an entity that has connections with other entities. For example, in a friendship network, people are the nodes and the edges represent the friendship between them. Connection of multiple paths forms a network. An edge in a network can be categorised as directed or undirected. A directed edge is a link that comprises of a start node which points to a specific end node. An undirected edge consist of a link connecting two nodes with no direction pointing either way [113]. Figure 2.1 shows the vertex, edge and difference between a directed network and undirected network.

Figure 2.1: (a) An undirected network. (b) A directed network
The distance of one node to another is the number of hops from the start node to the end node. A node, \( i \), is located one hop away from node \( j \), if node \( j \) is adjacent to node \( i \), that is there is a link from \( i \) to \( j \). Thus, the path length from node \( i \) to \( j \) is 1 \[115\]. In another example, the path from \( i \) to \( j \) has a length of three means the number of links or hops from \( i \) to \( j \) is three, that is, there are two other nodes, \( p \) and \( k \) located in between \( i \) and \( j \) \[115\].

\[ i \rightarrow p \rightarrow k \rightarrow j \]

A network of \( n \) nodes and \( m \) links may contain multiple paths connecting one node to another. If paths exist from \( i \) to \( j \), a shortest path is a geodesic path between these two nodes such that no shorter path exists \[115\]. A shortest path has minimum length between two nodes; “the length of a geodesic path, often called the geodesic distance or shortest distance, is thus the shortest network distance between the vertices” \[115\]. The terms discussed above are used throughout this thesis.

2.2.1.2 Centrality Measures

My analysis involves unique steps to obtain the criminal links within the Enron email network and Noordin Top Terrorist network. The eigenvector centrality measure allows me to identify a node that not only has a large number of neighbours but is also connected to other nodes which themselves have large degree \[115\] \[19\].

**Definition 2. Eigenvector Centrality**

*Given a graph \( G = (V, E) \), with nodes \( V \) and edges \( E \), let \( A \) be the adjacency matrix for this graph where \( a_{ij} = 1 \) if nodes \( i \) and \( j \) are connected by an edge and \( a_{ij} = 0 \) if they are not. The eigenvector centrality of a node is proportional to the sum of the centralities of the nodes to which it is connected. \( \lambda \) is the largest eigenvalue of \( A \) and \( n \) is the number of nodes \[19\]:*

\[
Ax = \lambda x, \quad \lambda x_i = \sum_{j=1}^{n} a_{ij} x_j, \quad i = 1, ..., n.
\]  

(2.1)

A node has highest eigenvector centrality when it is connected to many nodes that have high degree. The node in yellow in Figure 2.2 is the node with highest eigenvector centrality and the measure is an indicator of popularity that tends to identify centers of large cliques \[19\] \[132\]. Thus, the node with highest eigenvector centrality is also called the “most influential”.

The other important centrality measure for this work is betweenness centrality. This measure can identify the nodes that on elimination could lead to the graph becoming disjoint \[115\] \[19\].

**Definition 3. Betweenness Centrality**
Given a path \((s, t)\), \(s\) the source node and \(t\) the end node, in between these two nodes lies the alternating sequence of nodes and edges, for instance, \(s, e_1(s, v_1), v_1, e_2(v_1, v_2), v_2, ..., e_i(v_i, t), t\) \cite{21}. Let the number of shortest paths from \(s\) to \(t\) be given by \(\sigma_{st}\), and let \(\sigma_{st}(v)\) be the number of shortest paths from \(s\) to \(t\) that pass through \(v\). The betweenness centrality \cite{64} of \(v\) is given by:

\[
BC(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}.
\]

Betweenness centrality of a node is equal to the number of times a node appears in the shortest paths that bridge one node to another or one component to another. A node with high betweenness plays the role of a ‘middleman’. This is shown by the yellow node in Figure 2.3.

2.2.2 Implementation of SNA on Enron Email Communication Network

Many researchers have explored and applied Social Network Analysis on the Enron Corporation’s email dataset which is publicly available \cite{44}. Shetty and Adibi \cite{138} have identified most relevant individuals using a graph entropy model \cite{138} which can predict the number of occurrences of an email sequence of certain length. For example, if an email sequence \(X\) has length \(h\) (\(h\) hops away), the entropy is defined as the number of occurrences of email sequence \(X\) divided by the total number of possible sequences with length \(h\) in the graph. An important node is spotted when the removal of the node and its connecting edges brings a large change in the graph entropy value. In this method, Shetty and Adibi \cite{138} set three properties of dependency of one email on another. In the sequence of email transactions, the second depends on the first with some condition, that is, if both appear within the same time frame, if a major part is copied from one email to another, if that email is forwarded and if the link is based on certain event. In this method, the results
show that choosing different lengths of email sequence $X$ gives different sets of influential nodes, all of which are identified to be in the group of higher authorities in the Enron company.

Qian et al. \[126\] have used sender and receiver information from the Enron email dataset \[44\] to connect and create edges between them and then used link mining to find communities. They categorize a community as strong when the in-degree of each node within a network is more than its out-degree and a community as weak when the sum of all in-degrees within a subgraph is larger than the sum of all out-degrees \[126\]. A divisive algorithm is then used to find the link clustering coefficient. The algorithm takes the fraction of possible loops of three nodes or four nodes that pass through a certain link. The link with minimum value is removed to detect different communities \[126\]. The main aim of Murshed et al. \[109\] is to find the communication network pattern during the dissolution period of a company. They \[109\] use three types of measurement, that is, cliques, centralization and connectedness to analyze the changes in a network structure with their result \[109\] showing that cliques grow when a company dissolves and connectedness among top management increases when the crisis is at its peak but at the same time, employees become decentralized during disintegration. However, I can’t apply this result to conclude the evolution of the network pattern in general because this analysis \[109\] is limited to the connection between the Enron employees only and was designed without considering outer vertices.

Rowe et al. \[131\] have used average response time and maximal complete subgraphs (cliques) to show the hierarchical rank of employees in Enron. Multiple graph theoretical metrics were used to calculate a social score for each email account and they \[131\] assumed that an account that belongs to a large number of cliques will be assigned the highest rank. The above research is more into investigation of social hierarchy and individual roles in an organization. Chapanond et al. \[40\] set a threshold where employees must have exchanged at least 30 emails with each member of a pair having sent 6 emails to the other. This threshold is used to filter noise between email connections where an edge representing a small number of emails is considered as noise and is removed. The main goal of this research \[40\] is to find community structure using graph metrics when the threshold increases or decreases with the end result showing that the clustering process used to form community structure was affected by the filtering process.

The task of manager-subordinate relationship identification by \[54\] involves a ranking method that analyses traffic and content of the emails. A scoring function is applied to rank each relationship with the purpose of identifying relevant communications that confirm certain social relationships such as friendship, trust, advice or management. Lim \[96\] uses a hierarchical fuzzy inference system to infer the degree of change and identify abnormalities. The case study in \[96\] takes an individual involved in the illegal activities within Enron as input and the degree of each
suspected node is used to describe the rating for largest and smallest change in the communication behaviour. The limitation of this system is its design complexity and the fuzzy sets need to be constructed manually based on the empirical knowledge of the email traffic. Diesner and Carley have used Organization Risk Analyzer (ORA) to identify the structural features of the Enron network with their software being able to identify the most important people using network centrality measures: closeness centrality, betweenness centrality and eigenvector centrality. The analysis results in a comparison between the period where Enron faced high crisis and when there was a smooth flow in the business. It was found that during the crisis, the connectivity among the top executives was very strong compared to that within the other employees. Since focus is given for only these two time periods, it is difficult to standardize the claims to the overall Enron network or to other organisations.

2.2.3 Implementation of Shortest Path Concept in Past Research

The shortest path concept has been used in many ways to identify strongly connected nodes. Xu and Chen modify the breadth-first-search and priority-first-search algorithms to identify the shortest path between two nodes. They make the assumption that a group of nodes have strong links if the product of link weights of each pair of nodes in the group is high. The weights for each link is given by a probability measure of how likely it is for the two nodes to be associated. On the other hand, Yunkai et al. combine the priority-first-search and two-tree Dijkstra algorithms to find the set of nodes that connect two different sets of source nodes at minimum distance. The authors have suggested using Xu and Chen’s link weight approach to detect a money laundering crime, however they have not shown the use or the results of the combined algorithms or the link weights on a money laundering dataset.

Sparrow in his paper states that removal of a node that appears the most number of times connecting a pair of nodes will disrupt the communications between those two nodes. This is supported by Memon et al. who developed dependence centrality formulae to obtain the dependency value of a node on an intermediate node to reach or communicate with a third node. If the removal of could isolate , then node is most dependent on to keep communicating or being able to link with other nodes in the network. The authors, Xu and Chen and Memon et al. have used the shortest paths from a node to all other nodes in the network for their experiments.

Tang et al. have used shortest lengths, that is, minimum number of paths to other nodes to represent closeness and to study the spread of a message. The authors also apply the concept of betweenness centrality that counts the number of shortest paths that pass through a
node to show the importance of that node to retaining a communication path showing that the method improves when taking into consideration the time that a source node takes to deliver a message.

The shortest path concept has been also used to explore the Enron dataset, for example, by Kostakos et al. [89] to quantify information flow between Enron employees within a given time frame. Meanwhile, total weight of shortest paths in Enron email hypergraphs [65] has been used to identify important persons. The experiment by Gao [65] shows that the higher the rank of an employee in Enron, the shorter his distance to the CEO.

In my research, I argue that criminals are positioned within a highly connected group if the criminals are found at less than 6 hops away from the most influential as well as the middle man in the combined shortest paths network; a new sub-network. The criminals who do not appear in this new sub-network stay peripheral. I will show that combining all shortest paths resulting from the shortest path network search Algorithm 1 in Chapter 3 can be used to identify new suspects for further investigation.

2.3 Scoring Important Nodes

There is some past research that has used different scoring methods to identify important nodes. White and Smyth [157] have modified the Markov chains and PageRank-style ranking algorithms by adapting, for example, the k-short Node-Disjoint Paths for measuring the relative importance of a node in a graph given a set of nodes of interest. The Markov walk-based function was also proposed to assign scores to each node and enhance the scoring function with relevant feedback information [1]. Similar to White and Smyth, Haveliwala et al. [73] use a query based page importance scoring method to rank the web pages.

A hierarchical method was introduced by Weiming and Qingxian [156] to estimate node importance through inheritance. Katz et al. [83] take the connection of individual $i$ to individual $j$ to determine the weight to construct an individual’s importance level. The level of influence of individual $j$ depends on the total number of paths between individuals $i$ and $j$. Node importance is also given by the centrality measure introduced by Stephenson and Zelen [142]. The centrality measure is built based on information flow between two pairs of nodes which takes into consideration all the possible paths for optimal information. The weight of each path between the two pairs of nodes is proportional to the amount of information exchanged and inversely proportional to the distance [142].

I explore how betweenness centrality can measure the importance of nodes. Betweenness centrality has been widely used in social network analysis [21], complex route planning [154],
highway-node routing [68 67], and computer network analysis [50] to determine important nodes. Betweenness centrality was first proposed by Anthonisse in 1971 [7] and later by Freeman [64]. I explore in detail two other techniques, by Brandes [20] and Geisberger et al. [67], that modify betweenness centrality to better quantify betweenness of nodes. I adapt these two latter techniques to calculate the reliance of a source node or the known suspect or criminal on an intermediate node and rank the importance level of each intermediate node based on this reliance value. The betweenness centrality measure and these techniques are discussed in detail in Chapter 5.

2.4 Criminal Network Formations

Past research shows that extracting criminal associations from raw data and representing them in a network graph not only needs preliminary information about criminal relationships but is also time consuming [15 53 42 120]. The task of the criminal investigator is further hampered by the mass of data needing to be searched with an important part of the start of an investigation being the identification of a smaller group that embodies the relationships within the criminal participants. Several prior works propose solutions to extract criminal networks in the form of associations between people or texts. In this section, we particularly focus on such extractions to produce evidence for criminal investigations.

There has been much research into the formation of criminal network graphs. Krebs [91] manually gathers data about 9-11 terrorist attacks from press articles forming the relationships between terrorists based on information such as same locality, same school, similar travel, etc. By mining the information available in the press articles, Krebs [91] has tried to identify the September 11, 2001 attack’s terrorist network organisation. The information collected was grouped into three categories; hijackers who were living together or entered the same school were supposed to have strong ties, hijackers who were travelling together or had meetings together were assigned moderate ties and those who have had only a single transaction or occasional meeting given the weakest links in the network. The drawbacks found in this technique are that links from one node to another in the network were formed manually and the ties that have been built are not necessarily relevant to the case. For example, linking those who went to the same school doesn’t strongly justify that they would have strong enough ties for plotting a criminal activity. Christin et al. [42] collect website domain names, bank accounts and phone numbers that belong to the same type of fraud to form fraud subgraphs. That a preliminary grouping process is needed is also shown by Didimo et al. [53]. In their paper, Didimo et al. [53] set bank names, customers, bank accounts as nodes and the transaction types as the edges. The network grouping is then done based on the different transaction types [53]. Nadji et al. [111] start with a known attack database to form
the community of malicious IP addresses using the Louvain method of community detection. In their work, the links between IP addresses can only be created using a set of recognised attack signatures for example URLs embedded in spam emails, network traces from malware dynamic analysis, etc. In a similar manner, Oatley and Crick group UK crime gangs by taking associations that are related not only to crime incidents but also include sibling relationships, partner, cohabitant, etc. Communities are seen within these gangs by exploring the network structure that is obtained when highly interconnected gangs are connected to one another through loosely linked intermediaries.

Mining relevant terms from a large volume of police department’s incident summaries and assigning the terms’ co-occurrence frequency to its weight is a method used by to design a criminal network. Yang and Ng use web crawlers to gather identities who joined in some crime related topics in web blog pages and represent them in a network. Louis and Engelbrecht conduct a text mining exercise on passages of a mystery novel to show the association between texts in the form of a graph that leads to the identification of murders. In, the posts that promote hate and violence in dark web forums are grouped in different cliques using a novel algorithm that adopts several similarity measures based on content, time, author and title. Similar to, in order to identify criminal cliques, Iqbal et al. have done some chat topic analysis and several entities that belong to the same chat session are formed into a clique.

Mining topics and keywords that could reveal criminal activity involvement is tedious, time consuming and could influence the decisions made by prosecutors. For example, using keywords from web-based data is a major problem when electronic documents, chat messages, web blogs or emails contain incomplete information or could mislead detection algorithms. Another way of forming groups or sub-networks is to analyse the community structure. Community detection algorithms have been built based on network modularity. In recent work, a log analysis tool that adapts several existing community detection algorithms such as Girvan and Newman’s algorithm and a variant based on modularity optimization, called Newman’s fast algorithm is developed to discover communities from phone call networks.

Communities are also formed using pre-defined rules. For example in, a set of people belong to the same community if their names appear together in one document. In other research, people who have an overlapping interest across various chat sessions are grouped into one community. I propose an algorithm in Chapter 3 where unlike and, I don’t restrict the community to be made up of the people who appear in the same email content or have the same keywords in their emails. Al-Zaidy et al. eliminate duplicate names that refer to the same person. I retain the duplicate email addresses of a person as these addresses may be important to obtain
secret trust connections. I presented the algorithm, the shortest path network search algorithm (SPNSA) that combines shortest paths and network centrality measures to extract a criminal’s trust network [102] in Chapter 3.

2.5 Software and Tools

In this thesis, I use the R language for the analysis of networks. R is “a language and environment for statistical computing and graphics” [128]. This programming language is written in RStudio that is “a free and open source integrated development environment for R” [128, 129]. Many software packages are available in RStudio such as ‘sna’, ‘network’, ‘shiny’, etc. to create and manipulate networks. The main software package that I use for this thesis is ‘igraph’ [48] to visualize the connections between entities and perform social network analysis. I also use ‘sna’ [32] and ‘network’ [33] packages to calculate the betweenness centrality value that is used to compare with the reliance measure. To present my data and output of the analysis, I use Microsoft’s Excel (2010) [79].
Chapter 3

SPNSA Explorative Method on the Enron BCC Email Network

*Part of this chapter is published in [102].

3.1 Introduction

Extracting a subset of an actor and then tracing the links from and to the actor is an easy way to form a small network for an investigation [161]. Two people are connected through a shortest path when the path between them travels through the least number of edges [45]. Shortest paths are usually used to retrieve an individual’s nearest connections and have been applied to find the strongest connections between criminals [158; 164]. I extend this technique by combining social network measures with the shortest path algorithm to achieve my research aim.

The Enron email dataset contains 1,887,305 email transactions [44] that were sent using the fields ‘TO’, ‘CC’ or ‘BCC’. Out of these emails, 16,116 are senders of the emails and 68,203 are receivers of the emails. The Enron email dataset contains a mix of internal and external email transactions. Within the 16,116 email senders, 5,831 email transactions are from email addresses that are Enron company email accounts having the name ‘enron’ in their email address and the rest of the addresses are external, for example andrew.fastow@ljminvestments.com, anitatatr@earthlink.net, etc. The emails that are sent from A to B or from B to A implies a directed network. In this chapter I use this directed email network.

Email transaction shortest paths between Enron employees have been studied by many researchers. Shetty and Adibi [138] choose different path lengths to obtain different sets of influential nodes within the group of higher authorities. Creamer et al. [17] use a clique detection algorithm to recursively produce Enron email subgraphs. They then calculate the mean length of
the shortest paths from a specific vertex to all other vertices in the subgraphs to obtain a geodesic
distance between vertices. Tang et al. [145] analyse the shortest path according to the time taken
by a sender to deliver an email to a recipient using the spread of a message to show the importance
of a person to retain a communication. Similarly, Yelupula and Ramaswamy [162] use email flow
and the number of recipients to predict the hierarchy of Enron management.

As discussed above, shortest paths have been mostly used to identify important actors [65; 145],
or to analyse the distance between actors [47] in the Enron email dataset. In this research, I focus
on the source based shortest paths problem to identify criminal connections. Distinct from other
research, for example [40; 54; 55; 109; 126; 131] and [138], I start by using emails that have BCC
recipients. The main reason for choosing BCC recipients is because the ‘to’ and ‘cc’ recipients are
visible to every recipient, while ‘bcc’ recipients are inherently secretive [105; 18], thus implying a
trust connection between sender and recipient.

Unlike the single-source shortest paths problem that requires identification of shortest paths
from a source vertex \( v \), to all other vertices in a graph, the proposed shortest paths network
search algorithm (SPNSA) in this chapter identifies the shortest paths from and to a selected set
of nodes called the algorithm feed. I combine this shortest path search with two social network
centrality measures. The node with highest betweenness centrality is defined as the middle man
while the node with highest eigenvector centrality is considered to be the most influential. These
two key nodes along with ego networks and shortest paths extraction are incorporated to build
the shortest path network search algorithm. I create the sub-network of an ego, retrieve shortest
paths between selected nodes and merge the links to uncover new connections. I start by applying
the algorithm to the ego network of the Chief Financial Officer of Enron, Andrew Fastow and
then follow similar steps for the other money laundering criminals. This approach results in the
identification of a new sub-network and new suspects.

In the next section, I describe the main criteria for our analysis, discuss the email selection,
give reasons for choosing BCC type recipients as well as giving a statistical analysis of the BCC
emails. In Section 3.3 I describe the shortest paths network search algorithm, show the discovery
of 1-BCC and 2-BCC shortest paths criminal network, the analysis and consequent results. In the
analysis part active intermediate nodes between criminals are identified.

3.2 The Main Criteria for the Analysis

According to a number of news articles [26; 39; 28; 25; 38; 113] as well as Brickey’s paper [23]
10 people were convicted of money laundering after the collapse of Enron. Table 3.1 shows the
criminals’ email addresses. Since the Enron dataset was stored into the SQL Server Management
Studio and unique IDs generated using structured query language code (SQL) code for distinct email addresses, several of the criminals have more than one node ID.

Table 3.1: Enron money laundering criminals

<table>
<thead>
<tr>
<th>Name</th>
<th>ID</th>
<th>Email Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrew Fastow</td>
<td>686</td>
<td><a href="mailto:andrew.fastow@enron.com">andrew.fastow@enron.com</a></td>
</tr>
<tr>
<td>Andrew Fastow</td>
<td>687</td>
<td><a href="mailto:andrew.fastow@ljminvestments.com">andrew.fastow@ljminvestments.com</a></td>
</tr>
<tr>
<td>Lea Fastow</td>
<td>11010</td>
<td><a href="mailto:lfastow@pop.pdq.net">lfastow@pop.pdq.net</a></td>
</tr>
<tr>
<td>Lea Fastow</td>
<td>11009</td>
<td><a href="mailto:lfastow@pdq.net">lfastow@pdq.net</a></td>
</tr>
<tr>
<td>Kevin Hannon</td>
<td>10068</td>
<td><a href="mailto:kevin.hannon@enron.com">kevin.hannon@enron.com</a></td>
</tr>
<tr>
<td>Kenneth Rice</td>
<td>9994</td>
<td><a href="mailto:kenneth.rice@enron.com">kenneth.rice@enron.com</a></td>
</tr>
<tr>
<td>Rex Shelby</td>
<td>15224</td>
<td><a href="mailto:rex.shelby@enron.com">rex.shelby@enron.com</a></td>
</tr>
<tr>
<td>Rex Shelby</td>
<td>15225</td>
<td><a href="mailto:rex_shelby@enron.net">rex_shelby@enron.net</a></td>
</tr>
<tr>
<td>A. Khan</td>
<td>205</td>
<td><a href="mailto:adnankkhk@hotmail.com">adnankkhk@hotmail.com</a></td>
</tr>
<tr>
<td>Michael Kopper</td>
<td>12708</td>
<td><a href="mailto:michael.kopper@enron.com">michael.kopper@enron.com</a></td>
</tr>
<tr>
<td>Ben Glisan</td>
<td>1369</td>
<td><a href="mailto:ben.glisan@enron.com">ben.glisan@enron.com</a></td>
</tr>
<tr>
<td>Joe Hirko</td>
<td>8716</td>
<td><a href="mailto:joe.hirko@enron.com">joe.hirko@enron.com</a></td>
</tr>
<tr>
<td>S. Yaeger</td>
<td>861</td>
<td><a href="mailto:anne.yaeger@enron.com">anne.yaeger@enron.com</a></td>
</tr>
</tbody>
</table>

Table 3.1 shows the list of criminals who were involved in Enron money laundering crime [23; 44]. Unique IDs generated using structured query language (SQL) code are assigned to distinct email addresses. Thus a criminal is referred to by one or more node IDs depending on the number of email addresses that the criminal holds.

3.2.1 Email Type Selection and Reasoning

Hershkop [75] has used sender and recipient email addresses, domain names, zones, the message size and the number of attachments to analyse email transactions, using these features to build a behaviour model by combining the statistical values; recipients’ frequency or the histogram of sender’s past activity to detect spam. Hershkop [75] suggests that if it was possible to find a rule to represent the appearance of TO, CC and BCC addresses then it would be easy to distinguish the pattern which violates the norms. Stolfo et al. [143] compile a list of recipients of a particular sender and reduce it by removing all the duplicates. This list is then divided into different subsets and used to model user cliques with any new clique that violates the normal pattern being detected as abnormal. Frequency of receiving the same email from a sender are used for detecting anomalies [143].

Yelupula and Ramaswamy [162] have focused on the features for ‘sent email’; number of emails sent, number of recipients in ‘TO’, ‘CC’ (Carbon Copy) and ‘BCC’ (Blind Carbon Copy) field, etc. for predicting organisational structure. Similarly, Zhang et al. [165] have used email sent and
received types of a group of employees to analyse organisational structure. Meanwhile, FROM, TO and CC fields are used by Hanseler [74] to map connections between the sender and receiver, with the advantage of modeling the subsets of TO and CC recipients being the ability to identify the frequency of email exchange from a particular sender to recipients. However, Hanseler [74] admits that the method could result in false allegations on those who make genuine communications, for example those who communicate with new customers, business partners or traders.

The focus of this chapter is on a particular type of email recipient field. There are three types of email recipient fields; TO, CC (Carbon Copy) and BCC (Blind Carbon Copy). An email’s TO field contains the actual recipient who needs to act upon and possibly reply to the message received. A copy of that email can be sent to a second recipient or a group of recipients who may not be directly involved in the message (the carbon copies) but who may need to know of the actions being taken. While the ‘to’ and ‘cc’ recipients are visible to every recipient, the BCC recipients are inherently secretive because the email addresses of recipients in the BCC field are concealed from the other recipients [105]. This implies trust between the sender and the BCC recipients [61]. In general, emails are bcc-ed when the sender intends to prevent the main receiver from knowing that the messages are being revealed to a third party [105]. Thus the communication between the sender and the bcc recipients implies a trust that the bcc recipients will not reveal themselves to the main receiver.

There are no studies in the literature related to usage of only those emails that have BCC recipients to investigate criminal activities. This motivates us to test social network analysis on this particular set of data that is available in the Enron dataset. I start with a statistical analysis of this dataset and identify a reduced dataset on which to conduct further analysis.

### 3.2.2 Statistical Analysis of Enron Emails with BCC Recipients

There are a total of 60,649 emails with BCC recipients in the Enron email database. This is named the BCC group. As Yasin et al. [161] point out, taking only a part of a large amount of data from a network helps to reduce the complexity of identifying criminals. Hence, I start with a statistical analysis to further reduce our search space, calculating the proportion of BCC recipients in each email and the average number of BCC recipients in an email. Figure 3.1 shows the BCC recipient percentage and the number of emails with distinct numbers of BCC recipients showing that within the bcc-ed emails, a large group, 17,784 emails had 33.73% BCC recipients, that is, approximately 1 out of 3 recipients were bcc-ed.
Figure 3.1: Number of BCC recipient percentage in emails

The second test on the average number of BCC recipients in an email is depicted in Figure 3.2 where the graph shows that the majority of emails had at most 5 BCC recipients.

Figure 3.2: Number of BCC recipients in emails

To get a better understanding of the emails with large BCC recipient lists, the ratios of BCCs in the email group with more than 5 recipients were calculated. There were 34,195 emails with more than 5 recipients. Figure 3.3 shows the ratio of BCC recipients to total recipients in each email in this group with a small number of emails having large bcc recipients lists. In addition, there are a few unusual scenarios: in one case, 1 recipient out of 948 recipients was bcc-ed; in other cases 1 recipient was bcc-ed out of 947; 457 recipients bcc-ed out of 915 and 448 recipients bcc-ed out of 897. Labelling such emails as illegal or suspicious may lead to false positives. Due to the doubts as to why only one recipient out of 948 or 947 was bcc-ed, as well as the other cases where the numbers of bcc-ed recipients were large and varying from the usual, I identified the
sender, recipients and also the contents.

Figure 3.3: BCC recipients in emails with more than 5 recipients

The sender and recipients of these emails shared emails that refer to “associate or analyst program”, “a call for round table discussion” and “upcoming conferences, seminars and workshops” respectively. These emails’ don’t seem to imply any kind of trust. If the BCC recipient list is large in an email, the sender would have likely shared general discussions among recipients. We assume then that emails that have large lists of BCC recipients do not imply a trust relationship, rather they are a quirk of the email system or just an information security practice.

For a more stringent analysis, I divided the emails with bcc recipients into emails with more than 5 recipients and emails that had at most 5 recipients. I analysed further those emails that were sent to at most 5 recipients. There were 26,454 emails of this type. On average, out of 5 recipients between 1 and 2 were bcc-ed. Accordingly, I restricted our search to emails where only one or two recipients were bcc-ed. The next section gives the BCC email network analysis of these particular groups and the results obtained.

3.3 BCC Email Network Analysis

The BCC email analysis in this section starts with a description of the proposed shortest paths network search algorithm (SPNSA). This is followed by the steps to discover 1-BCC and 2-BCC criminal shortest paths network by selecting the money laundering criminals in Table 3.1 as the algorithm feed.
3.3.1 The Algorithm

The R igraph package was used to create a network graph of all emails with either one recipient or two recipients bcc-ed (called the 1-BCC and 2-BCC groups). In the 1-BCC group, a large graph consisting of 5290 vertices and 17839 links was discovered, while the 2-BCC group gave a slightly smaller graph with 3766 vertices and 13486 links. Since I wish to focus mainly on the study of the network of one person, also called ego network analysis, at this stage, I begin with a selection of a particular node (ego) and then identify the other nodes (alters) to which this node (ego) is connected. This method is called ego-centered approach with alter connections [71]. My algorithm incorporates this method with the shortest paths and two centrality measures. The process is described in Algorithm 1.

3.3.1.1 Terms and Representation

The following abbreviations of terms are used in the algorithm:

\[
\begin{align*}
A_C & := \text{[Array of criminals (algorithm feed)]} \\
C_i & := \text{[i}^{\text{th}} \text{Criminal]} \\
EC_i & := \text{[Ego / i}^{\text{th}} \text{criminal in the list } A_C] \\
N_{EC_i} & := \text{[Ego / i}^{\text{th}} \text{criminal’s sub-network]} \\
MI_{N_{EC_i}} & := \text{[Most Influential Node in } N_{EC_i}] \\
MM_{N_{EC_i}} & := \text{[Middle Man Node in } N_{EC_i}] \\
OC & := \text{[Other Criminals in the ego sub-network } N_{EC_i}, \text{not including the ego } EC_i] \\
R & := \text{[Result]}
\end{align*}
\]

Details of terms used: \(C_i\) is a criminal and each criminal is stored in an array, \(A_C\). For each iteration, I take a criminal \(C_i\) from \(A_C\) as an ego and refer to it as \(EC_i\). An ego can be any suspicious entity. In our BCC email network analysis, \(C_i\) refers to the criminals who were involved in the Enron money laundering crime as discussed in section 2.1.1. \(N_{EC_i}\) is the \(i^{\text{th}}\) criminal’s sub-network, also called the ego sub-network. A sub-network is a network which comprises of all the vertices reachable from an \(i^{\text{th}}\) ego, \(EC_i\), all the vertices from which \(i^{\text{th}}\) ego, \(EC_i\), is reachable and all the links connecting these vertices. The set of all the vertices reachable from the \(i^{\text{th}}\) ego, \(EC_i\) is called the out-component and the set of all the vertices from which \(i^{\text{th}}\) ego, \(EC_i\), is reachable is called the in-component. \(MI_{N_{EC_i}}\) and \(MM_{N_{EC_i}}\) are the most influential and the middle man in
the ego sub-network $N_{EC_i}$ respectively. $OC$ refers to other criminals in the ego sub-network not including the ego $EC_i$. $R$ denotes the result that is obtained in every step of Algorithm 1.

Algorithm 1 comprises of 8 main functions. After selecting an ego, $EC_i$ from $A_C$, the subcomponent function from igraph is used to extract the in-component and out-component of the $EC_i$. Then, the subgraph function is used to create an ego sub-network, $N_{EC_i}$ containing the union of the in-component and out-component of the $EC_i$ and all the edges among them. Next, the Most Influential (MI) and the Middle Man (MM) in $N_{EC_i}$ are found and the direct links from the $EC_i$ to MI and MM are extracted. If a direct link doesn’t exist, the shortest paths from the $EC_i$ to MI and MM are extracted. I use the shortest paths from the $EC_i$ to the MI and the MM to reveal the position and importance of the $EC_i$ in the network. The closer the $EC_i$ to the MI, the less isolated the $EC_i$, indicating that it is positioned in between highly linked nodes. Meanwhile, the nearer the $EC_i$ to the MM, the higher the possibility the $EC_i$ could be using the MM to actively convey messages to the other nodes or criminals.

The next step includes extracting the shortest paths from other criminals in the array, $A_C$, to the MI and the MM followed by extracting the shortest paths from the $EC_i$ to the other criminals in the same ego sub-network. The steps above are repeated for the 9 other criminals by selecting each one as an ego. In the last step of the algorithm, the union of all the shortest paths (result from R1 to R7) give a shortest path network, showing each criminal’s position, relationships with
other nodes and the network association pattern. The steps are described in Algorithm 1.

Algorithm 1 Shortest Paths Network Search Algorithm (SPNSA)

A. Store the criminals, $C_1$ to $C_n$ in an array $A_C$.

B. Form sub-network of each ego and follow the steps below until all ego sub-networks have been tested.

```plaintext
for $i = 1$ to $n$ do
  1. Select a criminal, $C_i$ from $A_C$ as ego, $EC_i$
  2. Retrieve the ego sub-network $N_{EC_i}$.

(a) connection from ego to MI in ego sub-network $N_{EC_i}$
   (i) Find $MI_{N_{EC_i}}$.
   (ii) Find the direct path from $EC_i$ to $MI_{N_{EC_i}}$.
        if exists then
          retrieve from graph and output it, then go to (b)...R1
        else
          go to (a)(iii)
        end if
   (iii) Find the shortest path from $EC_i$ to $MI_{N_{EC_i}}$.
        if exists then
          retrieve from graph and output it, then go to (b)...R2
        else
          go to (b)
        end if

(b) connection from ego to MM in ego sub-network $N_{EC_i}$
   (i) Find $MM_{N_{EC_i}}$.
   (ii) Find the direct path from $EC_i$ to $MM_{N_{EC_i}}$.
        if exists then
          retrieve from graph and output it, then go to (c)...R3
        else
          go to (b)(iii)
        end if
   (iii) Find the shortest path from $EC_i$ to $MM_{N_{EC_i}}$.
        if exists then
          retrieve from graph and output it, then go to (c)...R4
        else
          go to (c)
        end if
```

Algorithm 1 Shortest Paths Network Search Algorithm (SPNSA) (continued)

(c) connection from OC to MI and MM in ego sub-network \( N_{ECi} \)
for \( i < n \) do
(i) Set OC = \( \{ EC_j | j \neq i \} \).
(ii) Find the shortest path from OC to \( MI_{N_{ECi}} \) in \( N_{ECi} \).
if exists then
retrieve from graph and output it, then go to c(iii)...R5
else
go to c(iii)
end if
(iii) Find the shortest path from OC to \( MM_{N_{ECi}} \) in \( N_{ECi} \).
if exists then
retrieve from graph and output it, then go to (d)...R6
else
go to (d)
end if
end for
(d) connection from ego to OC in ego sub-network \( N_{ECi} \)
for \( i < n \) do
(i) Set OC = \( \{ EC_j | j \neq i \} \).
(ii) Find the shortest path from \( EC_i \) to OC in \( N_{ECi} \).
if exists then
retrieve from graph and output it...R7
end if
end for
end for
C. Merge R1-R7 into a network

Then using the new combined sub-network, I calculated the number of times a node appeared connecting a known criminal to other nodes to discover new suspects.

3.3.2 1-BCC criminal shortest paths network

The subcomponent and subgraph functions from igraph were used to retrieve the ego sub-network from the 1-BCC email group. Note that Andrew Fastow, Lea Fastow and R. Shelby were found to have two email addresses each; (Andrew Fastow (686, 687), Lea Fastow (11009, 11010), Rex Shelby (15224, 15225)) (refer also to Table 3.1). I also added these three criminals’ second email address node IDs into our shortest paths network analysis. I start by investigating Andrew Fastow’s sub-network in the 1-BCC network followed by the other criminals’ sub-networks.

Step 1 : (B.1 Alg[1]) Retrieving ego network of \( C_i \), Andrew Fastow

Since Andrew Fastow was convicted of money laundering [38] and was the Chief Financial Officer of Enron, I started the investigation with him. The subcomponent function allowed me to obtain all the vertices reachable from Andrew Fastow (686) and all vertices from which Andrew Fastow (686) is reachable via directed paths. Then a subgraph was induced using the subcomponent value and a sub-network of Andrew Fastow was formed (for subcomponent and subgraph...
functions’ definition: refer to section 3.3.1. In this directed network reciprocity is low and it is actually relatively rare for a path to exist between two nodes; if a path from A to B exists this does not mean that a path from B to A exists. In the I-BCC network, Andrew Fastow’s (andrew.fastow@enron.com (686)) ego network exists but the ego network of Andrew Fastow from email address (andrew.fastow@ljminvestments.com (687)) didn’t exist.

Step 2 : (B.1 (a)(i), B.1 (b)(i) Alg 1) Find the MM and the MI

The two important centrality values in Andrew Fastow’s sub-network were then calculated. Node ‘16383’ (sara.shackleton@enron.com) had the highest betweenness centrality value and was denoted as the middle man (MM). According to [71][115], the MM represents the intermediate node that is positioned on the most number of shortest paths between other node pairs. Meanwhile, node ‘19075’ (vince.kaminski@enron.com) had the highest eigenvector centrality and was identified as the most influential (MI) node that connects to many other high degree nodes [71][115].

Step 3 : (B.1 (a)(ii)(iii), B.1 (b)(ii)(iii) Alg 1) Find the direct connection or the shortest path from Andrew Fastow to MM and MI

These ego network analysis steps were used to find a direct connection from Andrew Fastow to the most influential or middle man. There was no direct connection from Andrew Fastow (686) to the MM (16383) and the MI (19075), hence I obtained the shortest path connecting Andrew Fastow to these two nodes.

Step 4 : (B.1 (c)(ii)(iii) Alg 1) Find the shortest path from other criminals to MM and MI

Step 3 was repeated for the Lea Fastow (11010, 11009), Kenneth Rice (9994), Kevin Hannon (10068), Rex Shelby (15224, 15225) and A. Khan (205), in Andrew Fastow’s sub-network. The remaining criminals listed in the Table 3.1 did not belong to Andrew Fastow’s ego network. The shortest paths found were from Lea Fastow (11010) and Kevin Hannon (10068) to the two most important nodes, ‘16383’ (MM) and ‘19075’ (MI). The results of steps 3 and 4 are shown in Table 3.2.

Step 5 : (B.1 (d)(ii) Alg 1) Find the shortest paths from EC, (Andrew Fastow) to other criminals

Finally the algorithm finds the shortest paths from Andrew Fastow to the 5 other criminals in his ego sub-network. The only shortest path found connected Andrew Fastow to Kevin Hannon
All the shortest paths are also shown in Table 3.2. All the connections found thus far were kept for further analysis.

Table 3.2: Shortest paths in Andrew Fastow’s ego sub-network in the 1-BCC network

<table>
<thead>
<tr>
<th>Start Node</th>
<th>End Node</th>
<th>Members in the path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrew Fastow</td>
<td>Sara.Shackleton (MM)</td>
<td>Louise Kitchen, Mark Taylor</td>
</tr>
<tr>
<td></td>
<td>Vince Kaminski (MI)</td>
<td>Louise Kitchen</td>
</tr>
<tr>
<td>Lea Fastow (OC)</td>
<td>Sara Shackleton (MM)</td>
<td>Mike Mcconnell, David Forster, Mark Taylor</td>
</tr>
<tr>
<td></td>
<td>Vince Kaminski (MI)</td>
<td>Mike Mcconnell, David Forster, Louise Kitchen</td>
</tr>
<tr>
<td>Kevin Hannon (OC)</td>
<td>Sara Shackleton (MM)</td>
<td>Jeff Skilling, Sherri Sera, James Derrick, Kate Cole</td>
</tr>
<tr>
<td></td>
<td>Vince Kaminski (MI)</td>
<td>Jeff Skilling, Sherri Sera, Steven Kean</td>
</tr>
<tr>
<td>Andrew Fastow</td>
<td>Kevin Hannon (OC)</td>
<td>Steven Kean</td>
</tr>
</tbody>
</table>

The Table 3.2 shows shortest paths obtained by applying Algorithm 1 to Andrew Fastow’s (686) sub-network ($N_{EC}$). The shortest path connection is recorded, if it exists, from Andrew Fastow to MM, Andrew Fastow to MI, Andrew Fastow to other criminals (OC) and from other criminals (OC) to the MM (Sara.Shackleton) and the MI (Vince Kaminski). The other criminals (OC) are from the array of criminals ($A_C$) not including Andrew Fastow.

Step 6 : Application of algorithm to the other 5 criminals in their respective ego sub-networks

An igraph code was used to check which criminals existed in the 1-BCC network. The criminals, Michael Kopper, Ben Glisan, Joe Hirko and S. Yaeger did not exist in the 1-BCC email group. Next, sub-networks were formed for each of the other 5 criminals in the 1-BCC email group; Lea Fastow, Kevin Hannon, Kenneth Rice, Rex Shelby and A. Khan. Apart from Andrew Fastow, 2 other criminals’ sub-networks were found; Lea Fastow (11010) and Kevin Hannon (10068). On retrieval of the sub-networks of Kenneth Rice (9994) and A. Khan (205), there were found to be only 2 and 3 nodes respectively. In such small groups, paths don’t exist due to two scenarios; either betweenness centrality value does not exist or two nodes obtained the same eigenvalue. Since this, by definition, does not form an ego sub-network, the two sub-networks, of Kenneth Rice and A. Khan, were dropped. Similarly, Lea Fastow (11009) and Rex Shelby (15224, 15225) were also dropped.

Following this, the most influential and the middle man were picked from the sub-networks of the other ID of Lea Fastow (11010) and Kevin Hannon (10068) separately. In both cases, I obtained Vince Kaminski (19075) as the MI and Sara Shackleton (16383) as the MM, just as was
the case with Andrew Fastow’s ego network. Then, step 1 to step 5 above were repeated for these sub-networks. The results are presented in Tables 3.3 and 3.4.

Table 3.3: Shortest paths in Lea Fastow’s ego sub-network in the 1-BCC network

<table>
<thead>
<tr>
<th>Start Node</th>
<th>End Node</th>
<th>Members in the path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lea Fastow</td>
<td>Sara Shackleton(MM)</td>
<td>Mike Mcconnell, David Forster, Mark Taylor</td>
</tr>
<tr>
<td></td>
<td>Vince Kaminski(MI)</td>
<td>Mike Mcconnell, David Forster, Louise Kitchen</td>
</tr>
<tr>
<td>Andrew Fastow (OC)</td>
<td>Sara Shackleton(MM)</td>
<td>Louise Kitchen, Mark Taylor</td>
</tr>
<tr>
<td></td>
<td>Vince Kaminski(MI)</td>
<td>Louise Kitchen</td>
</tr>
<tr>
<td>Kevin Hannon (OC)</td>
<td>Sara Shackleton(MM)</td>
<td>Jeff Skilling, Sherri Sera, James Derrick, Kate Cole</td>
</tr>
<tr>
<td></td>
<td>Vince Kaminski(MI)</td>
<td>Jeff Skilling, Sherri Sera, Steven Kean</td>
</tr>
<tr>
<td>Lea Fastow</td>
<td>Andrew Fastow (OC)</td>
<td>Mike Mcconnell, Mark Mcconnell, Jan Moore, Rod Hayslett, Tim Despain</td>
</tr>
<tr>
<td></td>
<td>Kevin Hannon (OC)</td>
<td>Mike Mcconnell, Alhamd Alkhayat, Steven Kean</td>
</tr>
</tbody>
</table>

Table 3.3 shows result of the algorithm applied to Lea Fastow (11010)’s ego sub-network. The table rows consist of the start node, end node and members in a path. Lea Fastow acts as the ego. For each step of Algorithm [1] the shortest path connection is recorded, if it exists, from ego to MM, ego to MI, ego to other 5 criminals (OC) and from other 5 criminals (OC) to MM and MI.

Table 3.4: Shortest paths in Kevin Hannon’s ego sub-network in the 1-BCC network

<table>
<thead>
<tr>
<th>Start Node</th>
<th>End Node</th>
<th>Members in the path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kevin Hannon</td>
<td>Sara Shackleton(MM)</td>
<td>Jeff Skilling, Sherri Sera, James Derrick, Kate Cole</td>
</tr>
<tr>
<td></td>
<td>Vince Kaminski(MI)</td>
<td>Jeff Skilling, Sherri Sera, Steven Kean</td>
</tr>
<tr>
<td>Andrew Fastow (OC)</td>
<td>Sara Shackleton(MM)</td>
<td>Louise Kitchen, Mark Taylor</td>
</tr>
<tr>
<td></td>
<td>Vince Kaminski(MI)</td>
<td>Louise Kitchen</td>
</tr>
<tr>
<td>Lea Fastow (OC)</td>
<td>Sara Shackleton(MM)</td>
<td>Mike Mcconnell, David Forster, Mark Taylor</td>
</tr>
<tr>
<td></td>
<td>Vince Kaminski(MI)</td>
<td>Mike Mcconnell, David Forster, Louise Kitchen</td>
</tr>
<tr>
<td>Kevin Hannon</td>
<td>Andrew Fastow (OC)</td>
<td>Jeff Skilling, Sherri Sera, James Derrick, Kate Cole, t..hodge (anonymous), Rod Hayslett, Tim Despain</td>
</tr>
</tbody>
</table>

Table 3.4 shows result of the algorithm applied to Kevin Hannon (10068)’s ego sub-network. The table rows consist of the start node, end node and members in a path. Kevin Hannon acts as the ego. For each step of Algorithm [1] the shortest path connection is recorded if it exist, from ego to MM, ego to MI, ego to other 5 criminals (OC) and from other 5 criminals (OC) to MM and MI.

Step 7: (C Alg[1]) Merge all the shortest paths and form a shortest path 1-BCC
Step 7 is the last step in the algorithm where the results 1 (R1) to 7 (R7) (refer to Algorithm 1) are combined and a new shortest paths sub-network of the 1-BCC network is found (See Figure 3.4). In this 1-BCC shortest paths network, three criminals Lea Fastow (11010), Andrew Fastow (686) and Kevin Hannon (10068) occurred (see Figure 3.4).

![Figure 3.4: 1-BCC criminals' shortest paths network. All the criminals who occurred in this network, Andrew Fastow (686), Lea Fastow (11010), and Kevin Hannon (10068), are highlighted. The MM (16383 - Sara Shackleton) and the MI (19075 - Vince Kaminski) are indicated.](image)

### 3.3.3 2-BCC criminal shortest paths network

In the 2-BCC group of emails two recipients are bcc-ed in each email. The same steps as used for the 1-BCC group were used to analyse the 2-BCC group. This section discusses the result of running the algorithm on the ego networks of Andrew Fastow and the other criminals in the 2-BCC email group. Both Andrew Fastow (andrew.fastow@enron.com (686)) and (andrew.fastow@ljminvestments.com (687)) sub-networks were extracted. The two important central nodes in both of the Andrew Fastow’s (686 and 687) sub-networks were again (sara.shackleton@enron.com (16383)), the MM and (vince.kaminski@enron.com (19075)), the (MI).

There were no path from Andrew Fastow to sara.shackleton@enron.com (MM) or to vince.kaminski@enron.com (MI) in either of Andrew Fastow’s sub-networks (686 and 687). The shortest paths from all the other criminals, Lea Fastow, Kevin Hannon, Ben Glisan, Kenneth Rice and Rex Shelby were extracted. Table 3.5 shows the result for shortest paths from all criminals to sara.shackleton@enron.com (MM) and vince.kaminski@enron.com (MI) in each of Andrew Fastow’s sub-networks repectively. There were no shortest paths from Andrew Fastow to other criminals in either of Andrew Fastow’s sub-networks.
Table 3.5: Shortest paths in Andrew Fastow’s sub-network

<table>
<thead>
<tr>
<th>Sub-network</th>
<th>Start Node</th>
<th>End Node</th>
<th>Members in the path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrew Fastow (686)</td>
<td>Kevin Hannon (OC)</td>
<td>Sara Shackleton (MM)</td>
<td>Sherri Sera, Greg Piper, Mark Taylor</td>
</tr>
<tr>
<td></td>
<td>Kevin Hannon (OC)</td>
<td>Vince Kaminski (MI)</td>
<td>Sherri Sera</td>
</tr>
<tr>
<td>Andrew Fastow (687)</td>
<td>Kevin Hannon (OC)</td>
<td>Sara Shackleton (MM)</td>
<td>Sherri Sera, Greg Piper, Mark Taylor</td>
</tr>
<tr>
<td></td>
<td>Kevin Hannon (OC)</td>
<td>Vince Kaminski (MI)</td>
<td>Sherri Sera</td>
</tr>
</tbody>
</table>

The Table 3.5 shows shortest paths [refer to section 2.2.1.1] that were retrieved from the Andrew Fastow’s sub-networks. These shortest paths are the result of Steps 1 to 5. The table rows consist of sub-network name, start node, end node and members in a path. The shortest path connection was recorded if it existed from Andrew Fastow to MM, Andrew Fastow to MI, Andrew Fastow to other criminals (OC) and from other criminals (OC) to MM and MI in either sub-networks.

In both of Andrew Fastow’s sub-networks, only Kevin Hannon (10068) was connected to the most influential and middleman. Next, I focus on other criminals’ ego networks. I again checked which criminals exist in the 2-BCC network. The criminals, Michael Kopper, Joe Hirko, S. Yaeger and A. Khan did not exist in the 2-BCC email group. Sub-networks were formed using the algorithm for each of the other 5 criminals in the 2-BCC email group; Lea Fastow (11009 and 11010), Kevin Hannon (10068), Ben Glisan (1369), Kenneth Rice (9994) and Rex Shelby (15224 and 15225). The results are shown in Table 3.6.
Table 3.6: Shortest paths in criminal’s 2-BCC sub-network ($N_{EC_{i+1}...EC_n}$)

<table>
<thead>
<tr>
<th>Sub-network</th>
<th>Start Node</th>
<th>End Node</th>
<th>Members in the path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lea Fastow (11010)</td>
<td>Kevin Hannon (OC)</td>
<td>Sara Shackleton (MM)</td>
<td>Sherri Sera, Greg Piper, Mark Taylor.</td>
</tr>
<tr>
<td></td>
<td>Lea Fastow (11010)</td>
<td>Vince Kaminski (MI)</td>
<td>Sherri Sera.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Andrew Fastow (687) (OC)</td>
<td>-</td>
</tr>
<tr>
<td>Lea Fastow (11009)</td>
<td>Kevin Hannon (OC)</td>
<td>Sara Shackleton (MM)</td>
<td>Sherri Sera, Greg Piper, Mark Taylor.</td>
</tr>
<tr>
<td></td>
<td>Kevin Hannon (OC)</td>
<td>Vince Kaminski (MI)</td>
<td>Sherri Sera.</td>
</tr>
<tr>
<td>Kevin Hannon</td>
<td>Kevin Hannon</td>
<td>Sara Shackleton (MM)</td>
<td>Sherri Sera, Greg Piper, Mark Taylor.</td>
</tr>
<tr>
<td></td>
<td>Kevin Hannon</td>
<td>Vince Kaminski (MI)</td>
<td>Sherri Sera.</td>
</tr>
<tr>
<td></td>
<td>Kevin Hannon</td>
<td>Andrew Fastow (686) (OC)</td>
<td>Sherri Sera, Rosalee Fleming.</td>
</tr>
<tr>
<td></td>
<td>Kevin Hannon</td>
<td>Ben Glisan (OC)</td>
<td>Sherri Sera, Greg Piper, Louise Kitchen, Rick Buy.</td>
</tr>
<tr>
<td></td>
<td>Kevin Hannon</td>
<td>Kenneth Rice (OC)</td>
<td>Sherri Sera, Greg Piper, Louise Kitchen, Michelle Cash, Twanda Sweet.</td>
</tr>
<tr>
<td></td>
<td>Kevin Hannon</td>
<td>Rex Shelby (OC)</td>
<td>Sherri Sera, Greg Piper, Travis McCullough.</td>
</tr>
<tr>
<td>Ben Glisan</td>
<td>Kevin Hannon (OC)</td>
<td>Vince Kaminski (MI)</td>
<td>Sherri Sera.</td>
</tr>
<tr>
<td>Kenneth Rice</td>
<td>Kevin Hannon (OC)</td>
<td>Sara Shackleton (MM)</td>
<td>Sherri Sera, Greg Piper, Mark Taylor.</td>
</tr>
<tr>
<td></td>
<td>Kevin Hannon (OC)</td>
<td>Vince Kaminski (MI)</td>
<td>Sherri Sera.</td>
</tr>
<tr>
<td>Rex Shelby</td>
<td>Kevin Hannon (OC)</td>
<td>Sara Shackleton (MM)</td>
<td>Sherri Sera, Greg Piper, Mark Taylor.</td>
</tr>
<tr>
<td></td>
<td>Kevin Hannon (OC)</td>
<td>Vince Kaminski (MI)</td>
<td>Sherri Sera.</td>
</tr>
</tbody>
</table>

Table 3.6: Result of the algorithm applied to sub-networks that belong to Lea Fastow (11010, 11009), Kevin Hannon (10068), Ben Glisan (1369), Kenneth Rice (9994) and Rex Shelby (15224). The table rows consist of the sub-network name, start node, end node and members in a path. The shortest path connection was recorded if it exists, from ego to MM, ego to MI, ego to other 5 criminals (OC) and from other 5 criminals (OC) to MM and MI. The other criminals (OC) are the list of criminals in the array $A_C$, not including the ego.

Table 3.5 and 3.6 show that in all sub-networks, Kevin Hannon connected to the middle man (MM) and most influential (MM) the most number of times. Again the algorithm picked the criminals located in between closely connected nodes. There is a direct connection from Lea Fastow to Andrew Fastow’s second email address (andrew.fastow@ljminvestments.com (687)). I consider the emergence of Andrew Fastow’s second email address in the 2-BCC network email group as suspicious. In the last step of the algorithm, results 1 (R1) to 7 (R7) are combined (refer to Algorithm 1) and a new shortest paths network is formed (See Figure 3.5).
Figure 3.5: 2-BCC criminals’ shortest paths network. The MM (16383 - Sara Shackleton) and the MI (19075 - Vince Kaminski) are indicated. All the criminals who occurred in this network, Andrew Fastow (686, 687), Lea Fastow (11010), Kevin Hannon (10068), Ben Glisan (1369), Kenneth Rice (9994) and Rex Shelby (15224), are highlighted.

Discussion

Figures 3.4 and 3.5 show the new 1-BCC and 2-BCC shortest paths network connections respectively. Almost every node in these two networks is connected to every other either through an edge or multiple edges. I extracted three criminals; Lea Fastow (11010), Andrew Fastow (686) and Kevin Hannon (10068) in 1-BCC shortest paths network and six criminals; Lea Fastow (11010), Kevin Hannon (10068), Ben Glisan (1369), Kenneth Rice (9994), Rex Shelby (15224), Andrew Fastow (686, 687) in 2-BCC shortest paths network. These nodes are located less than 5 edges away from MM and MI, and thus would be said to be within the vicinity of highly linked nodes.

In the next section, the paths and intermediate nodes that connect every criminal to other nodes in the shortest paths network are identified. For this, I assume node A is connected to node D through the intermediate nodes B and C no matter the content of the email messages being exchanged (see Figure 3.6).
3.3.4 Criminal shortest paths network analysis

Tables A.1 - A.3 in Appendix A depict the paths from Andrew Fastow (686), Lea Fastow (11010) and K. Hannon (10068) to other nodes and the intermediate nodes along each path. From the paths formed, there were some common nodes that appeared the most number of times connecting these criminals to other nodes in the network. Tables A.4 - A.6 in Appendix A show the frequency with which these nodes appeared between Andrew Fastow (686), Kevin Hannon (10068) and Lea Fastow (11010) and other nodes. Louise Kitchen (louise.kitchen@enron.com (11370)) appears the most number of times connecting Andrew Fastow to other nodes. Meanwhile the common nodes that connect Kevin Hannon and Lea Fastow to other nodes are, respectively, Jeff Skilling (jeff.skilling@enron.com (8024)) and Mike Mcconnell (mike.mcconnell@enron.com (12935)).

Similar to the analysis above, paths connecting criminals to other nodes and the intermediate nodes in the 2-BCC shortest paths network are identified. The paths are shown in Table A.7 in Appendix A. The nodes found to be active in between Kevin Hannon (10068) and other nodes in the 2-BCC shortest paths network are Greg Piper (greg.piper@enron.com (6667)) and Sheri Sera (sherri.sera@enron.com (16926)). Table A.8 in Appendix A shows the frequency of a node connecting Kevin Hannon (10068) to other nodes.

New people to investigate

With the results that were obtained, I investigated the active intermediate nodes: Louise Kitchen, Mike Mcconnell, Jeff Skilling, Greg Piper and Sheri Sera to see if they had any relationship to important events that occurred during the period leading up to the collapse of the Enron.

Jeff Skilling was the President and Chief Operating Officer (COO) of Enron Corporation in December 1996 [6]. To support Enron’s fast growth in the 1990s, Skilling hired the best intellectuals for the company. This accounts for the appointment of Michael (Mike) McConnell [119, 133] as the Executive Vice President, Technology, Enron Corp., in July 1999. At the end of 1999, Enron Online came into being and Louise Kitchen, a trader at Enron [119, 133] was the main person involved in its start-up. McConnell later helped to promote Enron Online [119]. The development of the Enron Online was hidden from the COO, Jeff Skilling by Louise Kitchen [139] with the deployment of Enron online being revealed to him only two weeks before it was launched [139].

The next person of interest is Greg Piper (6667), the Managing Director of Enron NetWorks, who supported the growth of the web based trading introduced by Louise Kitchen [81]. He was responsible for all of Enron’s e-commerce systems development, such as ‘EnronOnline’ and ‘ClickPaper.com’ [81]. Thus, Louise Kitchen and Greg Piper were both connected with Enron.
Online. Although I can’t prove that these identities are suspicious, these interesting and intriguing events indicate that they should be investigated further. In fact, Louise Kitchen has been identified as an important node in prior research using node neighbourhood search, page rank [160] and rule-based search on the Enron employee’s job field [167]. On the other hand, even though Sheri Sera occurred most frequently between Kevin Hannon and other nodes, history does not indicate that she should be considered suspicious [80].

3.4 Conclusion

The experimental results show that SPNSA can extract a small and hence manageable network of e-mail addresses and their relationships from a subset of the Enron email dataset with bcc-ed recipients. Criminals who were linked in the shortest paths to MM, MI or other criminals in a network were identified. The criminals that appear in the merged shortest paths network within the vicinity of MM and MI (see section 2.2.1.1) are categorised as located in a closely connected network.

Identifying the criminal shortest paths network and connections with other active intermediate nodes can be the first step of an investigation. It is important to identify the people who emerge between these criminals and other nodes to see with whom the criminals connect the most. The active intermediate nodes between money laundering criminals in this analysis were found to be Louise Kitchen, Mike Mcconnell, Jeff Skilling, Greg Piper and Sheri Sera. The active intermediate nodes that are found through this analysis are merely an output of this analysis, and I do not claim the involvement of these people in the Enron money laundering activity. The work in this chapter shows that SPNSA is able to provide means of identifying possible people to investigate, when crime is suspected. In the next chapter, I validate the algorithm and show that it performs well in a variety of scenarios and on large networks.
Chapter 4

SPNSA Validation and Comparison with Community Detection Algorithms

*Part of this chapter is presented in [101].

4.1 Introduction

In the previous chapter, to start the shortest paths network search or analysis, a couple of small subsets; 1-BCC and 2-BCC networks were used. In this chapter, the entire Enron email dataset is divided into two different subsets, ‘BCC’ and ‘TO/CC’ email transactions. SPNSA is applied on these larger datasets to test the efficacy of the algorithm. The test comprises of the comparison between the sub-networks extracted using SPNSA and applying existing community detection and k-neighbourhood methods.

The experiments using the community detection algorithms are shown in section 4.2. Section 4.3 shows the application of the shortest path network search algorithm on the directed and undirected ‘BCC’ and ‘TO/CC’ email networks. Next, some crime investigation methods are proposed to measure the efficacy of the algorithm. The first scenario is the first suspect test; when an investigator doesn’t have any clue about the identity of the criminals. Different algorithm feed that represents different suspects are used to show the perfomance of the SPNSA. Then in section 4.4.2, random nodes are used as the algorithm feed to compare the result with the tests in the previous section. Another new scenario is also added to this investigation, that is when an investigator fails to identify one of the criminals. This corresponds to a form of validation known as leave-one-out. This is explained in section 4.4.3.
4.2 Comparison of SPNSA with existing community detection algorithms

Community structure exists when a graph consists of sets of nodes that are joined together in tightly knit groups forming separate individual communities [69; 115]. The link structure and node attributes are the common components used in community detection algorithms. Many authors have used algorithms to analyse community structure and to consequently identify groups or sub-networks. Girvan and Newman [69; 118] use link structure to find edge betweenness, removing the edge with highest betweenness score and then repeating these steps until the network becomes disconnected and a set of communities are identified.

Fast greedy community detection [43] uses a modularity optimization algorithm by first computing the fraction of within-community edges in a network, then subtracting from it the expected fraction of edges in a randomized version of the same network with the same degree distribution [43]. A nonzero value above 0.3 shows a good measurement for the density of links inside communities [43], [17]. The leading eigenvector community detection algorithm also implements the modularity optimization algorithm. It computes the modularity matrix and the eigenvector of the matrix. It then divides the community based on the positive or negative sign of the elements in the eigenvector. If the large elements have the same sign, then the network has no community structure [114]. Walktrap community detection merges similar nodes that are obtained using short random walks into a group [125].

A different way of detecting communities is by using the node based approach called agglomerative algorithms [17]. Pons and Latapy [125] introduce a random walk concept which picks nodes from a network based on a fixed distance between two nodes. The nodes are then grouped as a tree (dendogram). EAGLE is a software algorithm created by Shen et al. [137] to identify communities. First, it adapts the maximal clique calculation introduced by Bron and Kerbosch [29], then removes the subordinate maximal clique. The algorithm then calculates similarity between each pair of nodes in the clique, merges them into a new community, finds the similarity of the new community by comparing with an already existing community with the steps being repeated until only one community remains.

The results produced by the different methods that I use to identify criminals’ community are discussed in this section. The community detection methods used here are $k$-Neighbourhood, Fast-greedy, Walktrap and Leading Eigenvector algorithms. I present the efficacy of the shortest paths network search algorithm (SPNSA) compared to the existing community detection algorithms by the size of the sub-graphs and the number of criminals identified.
4.2.1 Enron email dataset analysis

For the purpose of the research in this section, I have used email transactions from the last 8 years of Enron, from 1995 to 2002 [133]. Irrelevant email transactions with email address such as ‘5673@aol.com’, that end with airline company name, for example ‘@aircanada.com’, that end with ‘xpedia.com’, ‘amazon.com’ and other auto response emails are removed. The dataset is divided into two parts. One part of it is made of email transactions that only have recipients in the ‘TO’ and ‘CC’ fields. The other part consist of emails with one or more bcc-ed recipients.

The Enron ‘TO and CC’ email network has 26,027 nodes and 1,048,572 edges. I name this network as Netgraph and each node ID in the Netgraph as Net. ID. The log-log plot of the degree distribution is displayed in Figure 4.1 (a).

![Degree distribution of 'TO & CC' email group](image)

(a) Degree distribution of ‘TO & CC’ email group

![Degree distribution of 'BCC' email group](image)

(b) Degree distribution of ‘BCC’ email group

Figure 4.1: The figures show that the degree distributions for both Netgraph (a) and BCC Netgraph (b) are heavy-tailed with many nodes being of low degree and some nodes being highly connected.

The network formed using emails with one or more bcc-ed recipients is called the BCC Netgraph with each node given a BCCNet. ID. The BCC Netgraph has 19,716 nodes and 238,761 edges. Each node in the BCC Netgraph has an average degree of 24.22 and the log-log plot of the degree distribution is shown in Figure 4.1 (b). Both the networks are directed graphs. As evident from Figure 4.1 (a) and (b), both the networks have many nodes with low degree and a few nodes with very high degree indicating that the degree distributions are heavy tailed.

Next, I present the email Net. ID and BCCNet. ID of the criminals that exist in either the Netgraph and/or BCC Netgraph in Table 4.1. Some email addresses of the criminals that occur in the BCC Netgraph did not occur in the Netgraph, thus the representation with different IDs.

---

[133]: Reference number
Table 4.1: Enron money laundering criminals

<table>
<thead>
<tr>
<th>Name</th>
<th>Net. ID</th>
<th>BCCNet. ID</th>
<th>Email Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrew Fastow</td>
<td>1472</td>
<td>686</td>
<td><a href="mailto:andrew.fastow@enron.com">andrew.fastow@enron.com</a></td>
</tr>
<tr>
<td>Andrew Fastow</td>
<td>-</td>
<td>687</td>
<td><a href="mailto:andrew.fastow@ljminvestments.com">andrew.fastow@ljminvestments.com</a></td>
</tr>
<tr>
<td>Lea Fastow</td>
<td>17589</td>
<td>11010</td>
<td><a href="mailto:lfastow@pop.pdq.net">lfastow@pop.pdq.net</a></td>
</tr>
<tr>
<td>Lea Fastow</td>
<td>17588</td>
<td>11009</td>
<td><a href="mailto:lfastow@pdq.net">lfastow@pdq.net</a></td>
</tr>
<tr>
<td>Kevin Hannon</td>
<td>16202</td>
<td>10068</td>
<td><a href="mailto:kevin.hannon@enron.com">kevin.hannon@enron.com</a></td>
</tr>
<tr>
<td>Kenneth Rice</td>
<td>16115</td>
<td>9994</td>
<td><a href="mailto:kenneth.rice@enron.com">kenneth.rice@enron.com</a></td>
</tr>
<tr>
<td>Rex Shelby</td>
<td>23983</td>
<td>15224</td>
<td><a href="mailto:rex.shelby@enron.com">rex.shelby@enron.com</a></td>
</tr>
<tr>
<td>Rex Shelby</td>
<td>23985</td>
<td>15225</td>
<td><a href="mailto:rex_shelby@enron.net">rex_shelby@enron.net</a></td>
</tr>
<tr>
<td>A.Khan</td>
<td>-</td>
<td>205</td>
<td><a href="mailto:adnankkhan@hotmail.com">adnankkhan@hotmail.com</a></td>
</tr>
<tr>
<td>Michael Kopper</td>
<td>20217</td>
<td>12708</td>
<td><a href="mailto:michael.kopper@enron.com">michael.kopper@enron.com</a></td>
</tr>
<tr>
<td>Ben Glisan</td>
<td>-</td>
<td>1369</td>
<td><a href="mailto:ben.glisan@enron.com">ben.glisan@enron.com</a></td>
</tr>
<tr>
<td>Joe Hirko</td>
<td>14052</td>
<td>8716</td>
<td><a href="mailto:joe.hirko@enron.com">joe.hirko@enron.com</a></td>
</tr>
<tr>
<td>S.Yaeger</td>
<td>-</td>
<td>861</td>
<td><a href="mailto:anne.yaeger@enron.com">anne.yaeger@enron.com</a></td>
</tr>
</tbody>
</table>

Table 4.1 gives the list of e-mail accounts associated with criminals involved in the Enron money laundering crime [23][44]. The IDs in the table are computer generated numbers assigned to distinct email addresses based on the type of network. The Net. ID refers to the email addresses of criminals in the Netgraph while the BCCNet. ID refers to the email addresses of criminals in the BCC email network.

4.2.2 Criminals’ communication links in Netgraph and BCCNetgraph

Criminal communication links within the Netgraph and BCCNetgraph are obtained by first calculating the length of the shortest paths from one criminal to another. Tables 4.2 and 4.3 show the length of shortest paths from criminal to criminal in the directed Netgraph and the BCC Netgraph respectively. The directed Netgraph has an average path length 3.203258 while the directed BCC...
Netgraph’s average path length is 4.445533.

Table 4.2: Shortest path length from criminal to criminal in the directed Netgraph

<table>
<thead>
<tr>
<th>Directed Netgraph</th>
<th>1472</th>
<th>17589</th>
<th>17588</th>
<th>16202</th>
<th>16115</th>
<th>23983</th>
<th>23985</th>
<th>20217</th>
<th>14052</th>
</tr>
</thead>
<tbody>
<tr>
<td>1472</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>17589</td>
<td>Inf</td>
<td>0</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
</tr>
<tr>
<td>17588</td>
<td>Inf</td>
<td>Inf</td>
<td>0</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
</tr>
<tr>
<td>16202</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>0</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
</tr>
<tr>
<td>16115</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>0</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
</tr>
<tr>
<td>23983</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>23985</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>0</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
</tr>
<tr>
<td>20217</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>0</td>
</tr>
<tr>
<td>14052</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>0</td>
<td>Inf</td>
</tr>
</tbody>
</table>

Table 4.2 shows the lengths of shortest paths from criminal to criminal in the directed Netgraph.

Table 4.3: Shortest path length from criminal to criminal in the directed BCC Netgraph

<table>
<thead>
<tr>
<th>Directed BCC Netgraph</th>
<th>686</th>
<th>687</th>
<th>11010</th>
<th>11009</th>
<th>10068</th>
<th>12708</th>
<th>1369</th>
<th>15224</th>
<th>15225</th>
<th>861</th>
<th>205</th>
</tr>
</thead>
<tbody>
<tr>
<td>686</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>687</td>
<td>Inf</td>
<td>0</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
</tr>
<tr>
<td>11010</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>11009</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>0</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
</tr>
<tr>
<td>10068</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>12708</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>0</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
</tr>
<tr>
<td>1369</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>9994</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
</tr>
<tr>
<td>8716</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td></td>
</tr>
<tr>
<td>15224</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>15225</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td></td>
</tr>
<tr>
<td>861</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td></td>
</tr>
<tr>
<td>205</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td>Inf</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.3 shows the lengths of shortest paths from criminal to criminal in the directed BCC Netgraph.

From Tables 4.2 and 4.3 it is clear that, in the directed graphs under consideration, only a
few criminals have directed paths connecting them to other criminals. If we make the broad assumption that an email sent from A to B implies an *undirected* relationship between A and B then the graphs become undirected. In this case, in the BCC Netgraph, 12 of the 13 accounts associated with criminals belong to the same connected component and a path can be found from one criminal’s account to another’s (see Table 4.5). The exception is the account associated with A. Khan (adnankkhan@hotmail.com) which belongs to a separate component. The assumption regarding reciprocal relationship seems most appropriate for the trust network (BCC Netgraph) where if A includes B as a BCC recipient there is a personal trust relationship implied between A and B that I assume is reciprocated to some degree.

The shortest path lengths between the criminals in the undirected graphs are shown in Tables 4.4 and 4.5. The average path length values of the undirected Netgraph and the undirected BCC Netgraph are 3.264676 and 5.033507 respectively. The average path length of the criminals in the undirected Netgraph and the undirected BCC Netgraph are 2.93 and 3.65 respectively; lower than the average path length of the entire graphs.

Table 4.4: Shortest path length from criminal to criminal in the undirected Netgraph

<table>
<thead>
<tr>
<th></th>
<th>1472</th>
<th>17589</th>
<th>17588</th>
<th>16202</th>
<th>16115</th>
<th>23983</th>
<th>23985</th>
<th>20217</th>
<th>14052</th>
</tr>
</thead>
<tbody>
<tr>
<td>1472</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>17589</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>17588</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>16202</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>16115</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>23983</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>23985</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>20217</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>14052</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.4 above shows the lengths of shortest paths from criminal to criminal in undirected Netgraph.
Table 4.5 above shows the lengths of shortest paths from criminal to criminal in undirected BCC Netgraph. The distance between any two criminals in the undirected Netgraph ranges from 2-4 (see Table 4.4) while in the undirected BCC Netgraph it ranges from 1-7 (see Table 4.5). Using the shortest paths’ lengths count, there are variations in the connections formed between criminals in the undirected BCC Netgraph compared to the undirected Netgraph. In the undirected BCC Netgraph, there are some direct links between certain criminals, for example Andrew Fastow (BCCNet. ID 686) to Lea Fastow (11010), Andrew Fastow (686) to Ben Glisan (1369) and from Michael Kopper (12708) to Ben Glisan (1369). The direct links between criminals shows close relationship between these criminals in the BCC Netgraph.

4.2.3 Identifying criminals’ community using community detection algorithms

The community detection methods used in this section to identify criminals’ communities are $k$-Neighbourhood, Fastgreedy, Walktrap and Leading Eigenvector algorithms. Due to the connections between criminals being more visible in the undirected graph compared to the directed graph (see Tables 4.2, 4.3, 4.4 and 4.5) and since the majority of the community algorithms can
only be applied to undirected graphs, I use the undirected Netgraph and BCC Netgraph for this exercise.

### 4.2.3.1 k-Neighbourhood detection

Each criminal’s sub-network is extracted using their neighbouring nodes. First the total degree of each criminal in both the undirected Netgraph and BCC Netgraph is computed, see Table 4.6. Using Table 4.6, I took Andrew Fastow (686), (the first criminal in our list) as an example and found all of his neighbours in the undirected BCC Netgraph at a distance of 1 to 4 to form the \( k \)-Neighbourhood networks.

<table>
<thead>
<tr>
<th>Name</th>
<th>Net. ID</th>
<th>Degree</th>
<th>BCCNet. ID</th>
<th>Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrew Fastow</td>
<td>1472</td>
<td>261</td>
<td>686</td>
<td>25</td>
</tr>
<tr>
<td>Andrew Fastow</td>
<td>-</td>
<td>-</td>
<td>687</td>
<td>1</td>
</tr>
<tr>
<td>Lea Fastow</td>
<td>17589</td>
<td>3</td>
<td>11010</td>
<td>4</td>
</tr>
<tr>
<td>Lea Fastow</td>
<td>17588</td>
<td>4</td>
<td>11009</td>
<td>4</td>
</tr>
<tr>
<td>Kevin Hannon</td>
<td>16202</td>
<td>1</td>
<td>10068</td>
<td>36</td>
</tr>
<tr>
<td>Kenneth Rice</td>
<td>16115</td>
<td>11</td>
<td>9994</td>
<td>4</td>
</tr>
<tr>
<td>Rex Shelby</td>
<td>23983</td>
<td>97</td>
<td>15224</td>
<td>21</td>
</tr>
<tr>
<td>Rex Shelby</td>
<td>23985</td>
<td>2</td>
<td>15225</td>
<td>2</td>
</tr>
<tr>
<td>A. Khan</td>
<td>-</td>
<td>-</td>
<td>205</td>
<td>2</td>
</tr>
<tr>
<td>Michael Kopper</td>
<td>20217</td>
<td>40</td>
<td>12708</td>
<td>6</td>
</tr>
<tr>
<td>Ben Glisan</td>
<td>-</td>
<td>-</td>
<td>1369</td>
<td>105</td>
</tr>
<tr>
<td>Joe Hirko</td>
<td>14052</td>
<td>11</td>
<td>8716</td>
<td>2</td>
</tr>
<tr>
<td>S. Yaeger</td>
<td>-</td>
<td>-</td>
<td>861</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4.6 shows the total degree of each criminal in Netgraph and BCC Netgraph.

Figure 4.2 shows the networks found using 1,2,3,4-neighbourhoods of Andrew Fastow (686) in the undirected BCC Netgraph respectively.
Figure 4.2: The figures above show the networks formed by using 1,2,3,4-neighbourhood (1,2,3,4-N) of Andrew Fastow (686) in the undirected BCC Netgraph respectively. In the 1-neighbourhood of Andrew Fastow only one other criminal was found, Ben Glisan (1369). The 2, 3, and 4- neighbourhood networks are clearly too dense to be able to identify other criminals easily.

In the 1-neighbourhood network of Andrew Fastow only one criminal was found, Ben Glisan (1369). According to Tables 4.2, 4.3, 4.4, 4.5 using either of Andrew Fastow’s email accounts (BCCNet. ID 686 or 687) and either of the undirected or directed BCC Netgraphs, the one or two neighbourhoods would only rarely contain the other known criminals. The size of the network becomes bigger as the neighbourhood increases. The same method when applied to the undirected Netgraph, also produces large graphs that are difficult to explore. Clearly, using these dense network graphs, it is difficult to analyse a criminal’s occurrence and connections with other
nodes.

4.2.3.2 Community detection algorithms in R igraph

In the next two sub-sections, the community detection algorithms Fast-greedy [43], Walktrap [125] and Leading Eigenvector [114] are used to compare the number of criminals, communities and connections of criminals to other nodes in the communities formed by these algorithms.

4.2.3.2.1 Results of undirected Netgraph  The Fastgreedy algorithm detects 15 communities. These communities range in size from 5,803 nodes to just 2 nodes. 6 out of the 10 criminals were found in the second largest community that had 5,163 nodes and 22,936 edges. One criminal, Kevin Hannon (16202) appeared in a much smaller group consisting of 230 nodes and 781 edges. Next, the Walktrap community detection algorithm was applied to the undirected Netgraph with the length of the random walk being 10 steps. Here, 530 communities were found by this algorithm, with the first community detected consisting of 3,126 nodes and 12,462 edges, and containing 6 of the 10 criminals (see Table 4.7). It was the second largest community formed by Walktrap. The largest community had 7,935 nodes while the smallest one had just 1 node. The third community detection algorithm used was the Leading Eigenvector. This detection algorithm detected 2 communities with the larger one containing 26,025 nodes and the smaller 2 nodes. All 7 criminals appeared in the larger community but the criminals were found to be isolated (see Table 4.7).

4.2.3.2.2 Results of undirected BCC Netgraph  The community detection algorithms were next applied to the BCC Netgraph that contains 65,532 edges and 19,716 nodes. The Fast-greedy algorithm found 832 communities, finding a number of small communities with less than 6 nodes each. 5 criminals were found to be in the largest community that had 2,195 nodes and 7,903 edges; Andrew Fastow (BCCNet. ID 686), Lea Fastow (BCCNet. ID 11010, BCCNet. ID 11009), Kevin Hannon (BCCNet. ID 10068), Ben Glisan (BCCNet. ID 1369) and Kenneth Rice (BCCNet. ID 9994). Ben Glisan (BCCNet. ID 1369) had the highest degree in this community. Meanwhile, Rex Shelby (15224, 15225) and S. Yaeger (861) belonged to the second largest community with 2,142 nodes and 8,222 edges. The criminal who was in one of the smaller communities was Joe Hirko (8716).

The Walktrap community detection algorithm produced 1,773 communities from the undirected BCC Netgraph. The largest community had 1,493 nodes and the smallest one had just 1 node. Seven out of 10 criminals happened to exist in the same community that had 1,254 nodes.
The Leading Eigenvector community detection algorithm found 719 communities in the BCC Netgraph. It ranged from the largest community with 15,792 nodes and the smallest with 1 node. Most of the criminals appeared in the largest community. In Table 4.7, I list the communities that the criminals belong to and the community size in both BCC Netgraph and Netgraph, that I identified manually.

Table 4.7: Criminals Found in Different Communities

<table>
<thead>
<tr>
<th>Net. ID</th>
<th>FG Com. ID</th>
<th>WT Com. ID</th>
<th>LEC Com. ID</th>
<th>BCCNet. ID</th>
<th>FG Com. ID</th>
<th>WT Com. ID</th>
<th>LEC Com. ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1472</td>
<td>{7, 5163}</td>
<td>(1, 3126)</td>
<td>(1, 26025)</td>
<td>686</td>
<td>(5, 2195)</td>
<td>(36, 1254)</td>
<td>(1, 2001)</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>687</td>
<td>(5, 2195)</td>
<td>(594, 2)</td>
<td>(719, 15792)</td>
</tr>
<tr>
<td>17589</td>
<td>{7, 5163}</td>
<td>(1, 3126)</td>
<td>(1, 26025)</td>
<td>11010</td>
<td>(5, 2195)</td>
<td>(594, 2)</td>
<td>(719, 15792)</td>
</tr>
<tr>
<td>17588</td>
<td>{7, 5163}</td>
<td>(1, 3126)</td>
<td>(1, 26025)</td>
<td>11009</td>
<td>(5, 2195)</td>
<td>(36, 1254)</td>
<td>(719, 15792)</td>
</tr>
<tr>
<td>16202</td>
<td>(10, 230)</td>
<td>(29, 228)</td>
<td>(1, 26025)</td>
<td>10068</td>
<td>(5, 2195)</td>
<td>(36, 1254)</td>
<td>(719, 15792)</td>
</tr>
<tr>
<td>16115</td>
<td>{7, 5163}</td>
<td>(1, 3126)</td>
<td>(1, 26025)</td>
<td>9994</td>
<td>(5, 2195)</td>
<td>(36, 1254)</td>
<td>(719, 15792)</td>
</tr>
<tr>
<td>23983</td>
<td>{7, 5163}</td>
<td>(1, 3126)</td>
<td>(1, 26025)</td>
<td>15224</td>
<td>(3, 2142)</td>
<td>(36, 1254)</td>
<td>(719, 15792)</td>
</tr>
<tr>
<td>23985</td>
<td>{7, 5163}</td>
<td>(4, 7935)</td>
<td>(1, 26025)</td>
<td>15225</td>
<td>(3, 2142)</td>
<td>(1365, 1)</td>
<td>(719, 15792)</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>205</td>
<td>(224, 3)</td>
<td>(1030, 3)</td>
<td>(719, 15792)</td>
</tr>
<tr>
<td>20217</td>
<td>{7, 5163}</td>
<td>(1, 3126)</td>
<td>(1, 26025)</td>
<td>12708</td>
<td>(2, 2113)</td>
<td>(36, 1254)</td>
<td>(1, 2001)</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1369</td>
<td>(5, 2195)</td>
<td>(45, 1493)</td>
<td>(719, 15792)</td>
</tr>
<tr>
<td>14052</td>
<td>{7, 5163}</td>
<td>(1, 3126)</td>
<td>(1, 26025)</td>
<td>8716</td>
<td>(17, 561)</td>
<td>(36, 1254)</td>
<td>(719, 15792)</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>861</td>
<td>(3, 2142)</td>
<td>(54, 1101)</td>
<td>(719, 15792)</td>
</tr>
</tbody>
</table>

Total number of communities: 15
530 2 832 1773 719

Table 4.7 shows community IDs to which each criminal belongs. The community and the size of each community is represented in curly brackets as \{i_{th} \text{ community, size}\}. The title of the columns are Net. ID (Netgraph ID), BCCNet. ID (BCC Netgraph ID), FG Com. ID (Fastgreedy Community ID), WT Com. ID (Walktrap Community ID) and LEC Com. ID (Leading Eigenvector Community ID). The total number of communities formed is shown at the bottom of the table. The communities with small numbers of nodes, 1-3 nodes, are highlighted in red.

4.2.3.3 Discussion of results obtained by R igraph community detection algorithms

The network partitioning using leading eigenvector to form communities seems not to be very effective for either the undirected Netgraph or BCC Netgraph, as the biggest community contains almost all the nodes. Some abnormal network structures were found in the communities of certain criminals (see Table 4.7). From the results of the Walktrap algorithm on the BCC Netgraph
Andrew Fastow (687) and Lea Fastow (11010) belong to a small group of just two nodes, forming a community on their own. A. Khan (205) also belongs to a community of just three nodes. A. Khan is linked to two other nodes with email addresses; toriarules@aol.com and mmorales@arnel.com. These email addresses are found to be external emails that do not belong to the Enron company email group. The Fastgreedy algorithm results in Rex Shelby (15225) being isolated in a community of his own, numbered 1365. All of these abnormalities occurred in the undirected BCC Netgraph. Through this analysis I found more communities appearing in the undirected BCC Netgraph compared to the undirected Netgraph. The total number of communities formed using each detection algorithm is also shown in the last row of Table 4.7.

The community that contains the most criminals; community numbered 5 using Fastgreedy algorithm and community numbered 36 using Walktrap algorithm on the undirected BCC Netgraph were extracted. The detection using Walktrap algorithm on the BCC Netgraph yields the best result, with some criminals appearing in a small community on their own while in another community 7 of 1254 members were criminals. However the result of visualising these communities shows enormous number of nodes and links; networks (as shown in Figure 4.3) an investigator would need to analyse in order to find any connections between criminals and other nodes.

![Two graphs showing community detection results](image)

(a) Community numbered 5 using Fast Greedy algorithm  (b) Community numbered 36 using WalkTrap algorithm

Figure 4.3: Figures above shows the community numbered 5 (2195 nodes) using fast greedy algorithm and community numbered 36 (1254 nodes) using walktrap algorithm on undirected BCC Netgraph.

Clique percolation community detection developed by Palla et al. [122] was also used to identify the Enron criminal community. I found that the clustering coefficient values for both the undirected Netgraph and BCC Netgraph were so low that the nodes did not converge to form communities. A high average clustering coefficient to an appropriate random network is needed to form sub-networks or clusters [122][76]. In the next section, I apply our shortest paths network search algorithm to all four networks, the directed and undirected, Netgraph and BCC Netgraph.
4.3 Application of Shortest Path Network Search Algorithm

In Chapter 3, the shortest paths network search algorithm (SPNSA) was used to identify a sparse network from a directed network of emails that have 1 and 2 bcc-ed recipients respectively to start an investigation. Here, I apply the SPNSA to the directed and undirected Netgraph and the BCC Netgraph. Unlike Chapter 3, the directed and undirected BCC Netgraph used here contains all emails with bcc-ed recipients. In this section, all the criminals in Table 4.1 are used as the feed for SPNSA. There are 7 criminals in the Netgraph and 10 criminals in BCC Netgraph (see Table 4.1).

4.3.1 Application of SPNSA on directed Netgraph and BCC Netgraph

Applying the SPNSA to the directed Netgraph results in all 7 criminals who exist in the network, occurring in the extracted shortest paths network except for one email ID of Lea Fastow (Net. ID 17589) (see Figure 4.4). Lea Fastow’s Net. ID 17589 that represents her second email address didn’t appear in the sub-network because the node doesn’t have an out-component that builds paths to other criminals or the central nodes.

Figure 4.4: The shortest paths network formed using the directed Netgraph. All 7 criminals who existed in Netgraph were found. The network formed is sparse and criminals’ connections can be easily identified. This sub-network contains 30 nodes. The nodes highlighted in red represent the criminals with Net. ID from Table 4.1.

The sub-network of the directed BCC Netgraph captured using SPNSA is shown in Figure 4.5. This network contains 8 out of the 10 criminals who exist in the BCC Netgraph. The two other criminals did not have any connection to other criminals or to the MM or MI, thus did not occur in this sub-network.
4.3.2 Application of SPNSA on the undirected Netgraph and BCC Netgraph

The SPNSA was then applied to the undirected Netgraph and the result is depicted in Figure 4.6. Again, all seven criminals occurred in this graph. This time all double email Net. IDs were captured in this sub-network.

The SPNSA was finally applied to the undirected BCC Netgraph and the result is shown
The result shows that, in this case, SPNSA is able to identify two connected components, with all 10 criminals. When compared to the result obtained from the undirected Netgraph, application of SPNSA to the undirected BCC Netgraph gives better results as it is able to show all the criminals’ connections and has more connected components (see Figure 4.7).

Figure 4.7: The shortest paths network formed using undirected BCC Netgraph. No criminal IDs were lost. The network formed is sparse and criminals’ connections can be identified. This sub-network consists of a main component of 74 nodes and another small component of 3 nodes. The nodes highlighted in red represent the criminals with BCCNet. ID from Table 4.1.

The criminals and their links can be clearly seen (see Figures 4.4, 4.5, 4.6 and 4.7). The undirected BCC Netgraph yields the most number of criminals in the shortest paths network. A direct comparison of the results of using R igraph community detection algorithms with the result of shortest paths network search algorithm (SPNSA) is documented in Table 4.8.
Table 4.8: Comparison between results found using R igraph community detection algorithms and SPNSA

<table>
<thead>
<tr>
<th>Community Detection Algorithm</th>
<th>Shortest Paths Network Search Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undirected Netgraph</td>
<td>Undirected BCC Netgraph</td>
</tr>
<tr>
<td>Community distribution</td>
<td>Community Size: 250 &lt; nodes &lt; 26,100.</td>
</tr>
<tr>
<td>Nature: large and difficult to explore. Community Size: 1,000 &lt; nodes &lt; 16,000.</td>
<td></td>
</tr>
<tr>
<td>Abnormalities</td>
<td>Walktrap community detection: found Andrew Fastow’s external email add. (687) and Lea Fastow (11010) appeared in the same small community of size 2 nodes.</td>
</tr>
<tr>
<td>Andrew Fastow’s external email add. (687) doesn’t exist.</td>
<td></td>
</tr>
<tr>
<td>A. Khan (205) doesn’t exist</td>
<td>Fastgreedy and Walktrap community detection: found A. Khan (205) belongs to a small community of size 3.</td>
</tr>
<tr>
<td>8/10 criminals.</td>
<td>All 10 criminals.</td>
</tr>
</tbody>
</table>

Table 4.8 shows the comparison between results found using R igraph community detection algorithms and SPNSA. The community formed by SPNSA is small and suitable for investigation. The 3-clique connection is formed as a separate network component and the nodes that are connected to the criminal can be easily identified. In the row giving total criminals, $\{i_{th} \text{ community, size}\}$ refers to $\{i_{th} \text{ community, size}\}$ and the same follows for others.

4.4 Crime investigation methods using SPNSA

Anwar and Abulaish [8] have analysed their algorithm’s performance by implementing three different scenarios based on the availability of information. Similar to [8], here I specify certain ways by which an investigator could extract possible criminal subgraphs using the shortest paths network search algorithm for a preliminary investigation. In this section, the undirected BCC Netgraph is chosen instead of the undirected Netgraph due to three reasons found through our experiments in section 4.2.2 as well as the comparison between results as given in Table 4.8: more connections were detected between the criminals in the undirected BCC Netgraph (see Tables 4.2, 4.3, 4.4 and 4.5), cliques of two or three criminals were found in the undirected BCC Netgraph (see Table 4.8) and the most number of criminals were found in the undirected BCC Netgraph (see Table 4.8).
4.4.1 Extracting sub-networks using non-criminals

In this section, I run the shortest paths network search algorithm (SPNSA) with an assumption that the crime investigator doesn’t know the existence of any money laundering criminals in advance. This is called first suspect test. This first suspect test involves removing all criminals, $C_i$ from the criminal list $A_C$ of the shortest paths algorithm (see Chapter 3, section 3.3.1) and replace it with the Enron finance managers or other top managers. In this test, the Enron managers (finance managers or the other top managers) act as egos.

First I simulate such a scenario by feeding in to the algorithm all the top managers (apart from financial managers) (see Table 4.9) in Enron to obtain their communication network from the undirected BCC Netgraph. I excluded top managers who are also money laundering criminals and also those with financial manager post. In this test, if a criminal happens to appear in the top manager shortest paths network, then the first suspect test is successful in identifying people to investigate without any criminal information. The result is shown in Figure 4.8.
Table 4.9: Enron managers

<table>
<thead>
<tr>
<th>Name</th>
<th>ID</th>
<th>Designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jeff Skilling</td>
<td>8024</td>
<td>President, Chief Operating Officer, and CEO</td>
</tr>
<tr>
<td>Kenneth Lay</td>
<td>9986, 9988</td>
<td>Chief Executive Officer and Chairman of The Board</td>
</tr>
<tr>
<td>Richard Causey</td>
<td>15330</td>
<td>Chief Risk Officer</td>
</tr>
<tr>
<td>Richard Buy</td>
<td>2164</td>
<td>Senior Risk Officer</td>
</tr>
<tr>
<td>Greg Whalley</td>
<td>6673</td>
<td>President and Chief Operation Officer</td>
</tr>
<tr>
<td>Mark Frevert</td>
<td>11865</td>
<td>Vice President</td>
</tr>
<tr>
<td>Rebecca Mark-Jusbasche</td>
<td>15130</td>
<td>CEO of Enron International</td>
</tr>
<tr>
<td>Lou Pai</td>
<td>11352, 11354</td>
<td>CEO of Enron Energy Services</td>
</tr>
<tr>
<td>Forrest Hoglund</td>
<td>7108</td>
<td>CEO of Enron Oil and Gas</td>
</tr>
<tr>
<td>Clifford Baxter</td>
<td>3155</td>
<td>CEO of Enron North America</td>
</tr>
<tr>
<td>Jim Derrick</td>
<td>4411</td>
<td>Enron General Counsel</td>
</tr>
<tr>
<td>Mark Koenig</td>
<td>11882</td>
<td>Head of Enron Investor Relations</td>
</tr>
<tr>
<td>Andrew Fastow</td>
<td>686, 687</td>
<td>Enron Chief Financial Officer</td>
</tr>
<tr>
<td>Ben Glisan</td>
<td>1369</td>
<td>Enron Corp Treasurer</td>
</tr>
<tr>
<td>Kenneth Rice</td>
<td>9994</td>
<td>CEO of Enron Wholesale and Enron Broadband Services</td>
</tr>
</tbody>
</table>

Table 4.9 shows the list of the Enron top managers. The top managers known to be money laundering criminals (see Table 4.1) and finance managers (see Table 4.10) are not included in this first suspect test; thus the top managers highlighted in red are removed from the SPNSA feed. Unique IDs are assigned to distinct top manager’s email addresses. A top manager is referred to by one or more node IDs depending on the number of email addresses that the manager holds.

Figure 4.8: Enron undirected BCC Netgraph shortest paths network using all top managers excluding criminals as algorithm feed. The nodes highlighted in red are all criminals.
When compared to the network graph obtained in Figure 4.7, 7 out of 10 criminals are found by the algorithm this time around. Apart from the top managers known to be criminals (see Tables 4.1 and 4.9), the SPNSA also extracts 4 other criminals; Lea Fastow (11010), Kevin Hannon (10068), Rex Shelby (15224) and Joe Hirko (8716) with this top manager feed test.

Next a shortest paths network is formed using only finance managers. However in this case, I assume that investigator doesn’t have any clue that some finance managers are also money laundering criminals [23; 44] and use all the the finance managers listed in Table 4.10. The network formed is shown in Figure 4.9.

Table 4.10: Enron finance managers

<table>
<thead>
<tr>
<th>Name</th>
<th>ID</th>
<th>Designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sherron Watkins</td>
<td>16929</td>
<td>Head of Enron Global Finance</td>
</tr>
<tr>
<td>Andrew Fastow</td>
<td>686</td>
<td>Enron Chief Financial Officer</td>
</tr>
<tr>
<td></td>
<td>687</td>
<td></td>
</tr>
<tr>
<td>Ben Glisan</td>
<td>1369</td>
<td>Enron Corp Treasurer</td>
</tr>
<tr>
<td>Rick Causey</td>
<td>15077</td>
<td>Chief accounting officer</td>
</tr>
<tr>
<td>Jeff McMahon</td>
<td>8071</td>
<td>Chief Financial Officer of Enron after Andrew Fastow.</td>
</tr>
</tbody>
</table>

Table 4.10 shows the list of Enron finance managers. Unique IDs are assigned to distinct finance manager’s email addresses. A finance manager is referred to by one or more node IDs depending on the number of email addresses that the finance manager holds.

Figure 4.9: Enron undirected BCC Netgraph shortest paths network using all finance managers as algorithm feed. The nodes (686, 687 and 1369) highlighted in red are finance managers who are also criminals. The other criminal who was found is Lea Fastow (11010).

When comparing the criminals found in the sub-network formed using the Finance managers
(see Figure 4.9), with those found in Figure 4.7, we see that Lea Fastow (11010) is the only other criminal found here, other than the finance managers who were also criminals. In the next section, random algorithm feed were used to test the efficacy of the SPNSA.

4.4.2 Extracting subgraphs using random feed

In the previous sections, I explored the results obtained by the shortest path network search algorithm when using a selected algorithm feed: sets of actual and possible suspects to a crime incident. Here, we contrast the results with the performance of the shortest paths algorithm using a random feed, for example a set of random nodes of a given size and that can be a combination of any nodes. As before, the test will be run on the undirected BCC Netgraph. The size of the set of random nodes is chosen based on the number of email addresses of the finance managers or the top managers as the algorithm feed since the aim is to test the performance of the SPNSA using these random feeds, and to compare the results obtained. There are 6 email addresses of the finance managers and 18 of the top managers and hence the tests in the next section are carried out using firstly a random feed of size 6 and then a random feed of size 18.

Random feed analysis and results

In the first instance, the SPNSA was run 1000 times and at each round 6 random nodes were generated using a random node generator to use as the algorithm feed, thus constructing 1000 random graphs. At each round; in each graph, the money laundering criminals [23, 44] are extracted and recorded. The total frequency of each money laundering criminal for 1000 random graphs are noted. Using random feeds, only 2 criminals occurred in 1000 random graphs; node 1369 (Ben Glisan) occurred 10 times and node 15224 (Rex Shelby) appeared only once. The rest of the criminals have frequency zero.

I also analysed the density of criminals that occur in 1000 random graphs. 11 of the 1000 graphs had a maximum of 1 criminal. There were no criminals found in the other 989 random graphs. Running SPNSA with the finance managers as the algorithm feed, extracted 3 criminals (see Figure 4.9). So in this case the possibility of identifying a criminal using a random feed is very low; less than 1 in 100 graphs contained any criminals.

Next, I used a random feed of size 18 for each iteration, so as to compare the results with that obtained by running the SPNSA with the top managers as the algorithm feed. Just 3 of the 1000 graphs generated contained exactly 1 criminal each; the rest of the graphs had none. Even with this larger random feed, only two criminals occurred; 10068 (Kevin Hannon) occurred in two
different random graphs and 1369 (Ben Glisan) occurred only once. With 18 random nodes at each iteration, SPNSA only manages to obtain 2 criminals in the total of 1000 random graphs. In contrast using a list of possible suspects; top managers as the algorithm feed, SPNSA identified 7 criminals in one graph (see Figure 4.8).

4.4.3 Extracting subgraphs using leave-one-out method

The leave-one-out method is widely used in various fields of research as a data sampling method for an algorithm and can be used to estimate performance of a predictive model. Past research shows that one can set the number of data points to be removed from sample data and use it for validation. This is also called delete-p cross validation.

I name this method as leave-$C_i$-out. Leave-$C_i$-out refers to dropping one criminal ($C_i$) from the list of criminals and running the shortest paths network search algorithm on the remaining criminals in the undirected BCC Netgraph. This method is a test of the ability of the algorithm to produce sub-networks that contain the convicted criminal not included in the algorithm feed. I name the criminal that is left out during each iteration as $C_i$. A criminal who has two email accounts has two different BCCNet IDs and during the leave-$C_i$-out process both their IDs are dropped from the algorithm feed.

The results of the leave-$C_i$-out method are given in Table 4.11. In 5 out of 9 cases the criminal who is left out occurs in the network formed by the shortest paths algorithm. For example, when Michael Kopper (12708) is dropped from the algorithm feed (list of criminals) the result obtained is a sub-network that contains Michael Kopper and the connections of Michael Kopper (see Figure 4.10).

Figure 4.10: Enron undirected BCC Netgraph shortest paths network when Michael Kopper (12708) is left out from the criminal feed list. The nodes highlighted in red are all criminals.
Table 4.11: Leave-$C_i$-out Method

<table>
<thead>
<tr>
<th>$C_i$ Out</th>
<th>BCCNet. ID</th>
<th>$C_i$ occurs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrew Fastow</td>
<td>686, 687</td>
<td>✗</td>
</tr>
<tr>
<td>Lea Fastow</td>
<td>11010, 11009</td>
<td>✓</td>
</tr>
<tr>
<td>Kevin Hannon</td>
<td>10068</td>
<td>✗</td>
</tr>
<tr>
<td>Kenneth Rice</td>
<td>9994</td>
<td>✗</td>
</tr>
<tr>
<td>Rex Shelby</td>
<td>15224, 15225</td>
<td>✓</td>
</tr>
<tr>
<td>Michael Kopper</td>
<td>12708</td>
<td>✓</td>
</tr>
<tr>
<td>Ben Glisan</td>
<td>1369</td>
<td>✓</td>
</tr>
<tr>
<td>Joe Hirko</td>
<td>8716</td>
<td>✗</td>
</tr>
<tr>
<td>S. Yaeger</td>
<td>861</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 4.11 shows the result for Leave-$C_i$-out method. It is a test to see if the criminals that are left out still occur in the network formed by the SPNSA. $C_i$ represents each criminal. The sign ‘✓’ indicates the $C_i$ appeared in the shortest paths network community while ‘✗’ are given to $C_i$ who do not appear in the extracted network. A. Khan (BCCNet. ID 205) was not included in this table, because, as shown in Figure 4.7 A. Khan (205) had no connections to other criminals, and further, appeared in a separate component containing 3 nodes.

4.5 Conclusion

The experiments in this chapter illustrate the efficacy of the shortest paths network search algorithm on a larger dataset by comparing the size of the sub-graphs formed and the number of criminals that can be identified in a sub-graph. The existing community detection algorithms in igraph did show the communities that had the criminals, however an investigator would need to manually find the community to which a criminal belongs. Retrieving the neighbourhood sub-networks of the criminals in the community, as identified by the existing community detection algorithms, resulted in dense networks, which were hard to visualise and possibly, even harder to analyse. The criminals’ connections in these sub-networks were hard to be identified.

The shortest paths network search algorithm (SPNSA) shows the criminals’ connections to other nodes for all subsets of the Enron email database. Three different investigation methods were tested using the SPNSA. In all three scenarios, the sub-networks formed by SPNSA were sparse and hence suitable for an investigator to see the connections as well as conduct further investigations. The SPNSA algorithm was able to extract and show the abnormalities through the sub-networks formed; components that contained the criminals’ connections with other nodes,
the 3-clique component could be easily detected (see Figure 4.10).

When the 18 top managers of Enron were used as the algorithm feed, the resulting sub-network identified 4 convicted criminals that were not managers and so not part of the algorithm feed. The leave-one-out method shows the SPNSA algorithm is successful in returning some unknown criminals (not in the algorithm feed) in the resulting sub-network. However, using random nodes as the SPNSA algorithm feed resulted in only 2 criminals, whether using 6 or 18 as the size of the random feed, and this only rarely. This leads me to conclude that in an investigation process, some pre-findings about a crime incident will give more information about related suspects and as this experiment shows, the performance of the SPNSA could increase when relevant suspects related to a crime incident are used as the algorithm feed.

The SPNSA allows the investigator to feed in the early suspects or suspicious entities into the criminal list of the algorithm, a function that is not available through other community detection algorithms. The quality of a criminal investigation can be improved when inputs can be specified as done in this chapter and hence, SPNSA could be a very useful preliminary investigation tool for an investigator. In the next chapter, my research continues to calculate and identify the criminal’s reliance on the other nodes in a network and the criminal reliance connection pattern.
Chapter 5

Shortest Paths Criminal Connection Analysis using Reliance Measure

*Part of this chapter is presented in [100].

5.1 Introduction

In this chapter, I investigate further the shortest paths sub-networks that were extracted using the shortest paths network search algorithm described in Chapter 3, section 3.3.1 and introduce a new method to enhance a criminal investigation process. My focus is on the reliance of a source node (suspect) on other nodes in the network. For this purpose, I explore the existing network node pair-dependency. Past research that uses dependency analysis is discussed in section 5.2. Section 5.3 contains the definitions and mathematical terms used, while section 5.4 describes the mathematical calculations of the proposed new reliance formula which is a type of dependency analysis. I compare the reliance values with other pair-dependency formula given by Brandes [20], Freeman [61], Geisberger et al. [67]. I also show the difference in formula between the reliance measure and Geisberger et al.’s formula in section 5.5.

The Enron ‘TO/CC’ and ‘BCC’ email transactions [44] and the Noordin Top terrorist network [57] are described in case study 1, section 5.6 and case study 2, section 5.7 respectively. For both, I show the ranking of nodes in a criminal sub-network using betweenness centrality measures, Markov centrality and Google PageRank to compare with rankings using the novel reliance measure. This analysis and the results obtained are presented in subsections 5.6.1 for the Enron network and 5.7.1 for the terrorist network. The reliance formula shows stark difference in the importance level of nodes as opposed to the other methods which all tend to show highly similar rankings. Thus, ranking using the reliance measure reveals different people for investigation.
These people are also called the persons of interest or as I name them, the suspects’ crime priority nodes. These nodes are later used recursively to form a criminal reliance sub-network that could aid and improve criminal investigations.

5.2 Backgroud

The betweenness centrality of a node is a measure that computes the number of geodesics (shortest paths) going through that node. The idea of the betweenness centrality of a node was first proposed by Anthonisse [7] in 1971 by taking the highest number of shortest flows through the node in a directed network. Later, in 1977, Freeman [64] defined the betweenness centrality of a node by taking the fraction of the shortest paths passing through the node of interest, from each source to each destination node. Although the concept of betweenness was first introduced by Anthonisse [7], the calculation introduced by Freeman is more widely used in many real world applications such as social network analysis [21], complex route planning [154], computer network analysis [50] etc. In 2001, Brandes proposed a new algorithm using node pair-dependency to compute betweenness centrality for large networks [20].

The dependency of node $i$ on node $j$ in a network is measured by the total influence of node $j$ on node $i$ [99]. Here I discuss past research that uses dependency analysis either to study the network structure or to find influential nodes. First I focus on the effect of the dependencies between nodes on the network structure. Direct dependencies of nodes are computed by Feizi et al. [59] using network deconvolution which reverses the effect of the transitive information flow across all indirect paths to obtain the original direct dependencies. A study of the robustness of a network by analysing the dependency and connectivity of links is conducted by Parshani et al. [123]. Their research predicts system failure, where failure of a node that is dependent of other nodes could cause the failure of the system. The method by Parshani et al. [123] uses a process to iteratively cascade failures and see the difference in the network structure by removing a fraction of the nodes that cause network failure [123].

Network metrics such as closeness and betweenness centrality were used by Zimmerman et al. [168] to predict failure in computer windows server files. Similar to the result by Parshini et al. [123], the analysis by Zimmerman et al. [168] shows that central nodes with many direct connections are more prone to failure. Some researchers have found that the causes for failures that arise in software [2] and internet services [35] are due to the use of a product-product dependency model. Dependency analysis is used by Kenett et al. [86] to identify similarities in sub-networks (directed networks) developing over time. Pierson et al. [124] use dependency analysis to predict political event sequences, showing that decision making in political process over time is highly
dependent on earlier sequences of connections [123]. Dependency links in all of the research above refer to the edges connecting a node to all of its immediate neighbours.

Dependency is also used to identify the connections between words. Quirk [127] employs a language parser to parse onto a target sentence and extract a word and its translation in a form of source-target hierarchical trees. Attardi et al. [11] model a dependency tree based on terms from questions and answers. The dependency relations between terms are used to extract the relevant answer passage. Cui et al. [49] propose a fuzzy relation matching to improve dependency relation based on statistical models to solve irrelevant passage retrieval due to retrieval of unmatched passage terms to a list of questions. Bipartite graphs are used by Alpcan and Bambos [4] to model dependencies where an edge represents the inter-dependencies or intra-dependencies of an organisation’s business units, security threats and people that contribute to the security risk. For example, the edge between people and business units reflects the dependence (responsibility) of employees on products, services or applications, and the dependence (vulnerability) of a product to a security threat [4]. Here too, the dependency refers to direct links between two entities or nodes.

Researchers have also used dependency values to find influential nodes. Dependency exists when there is a request or information traffic flow between the source and destination. Dependency can also be valued by the level of contributions where the contribution of a node to some function, for example writing a paper, is used to measure the dependency of the node on the network where the function determines the path type [97]. Kourtellis et al. [90] introduced a randomised algorithm by adapting betweenness centrality to quantify the dependency or influence of nodes using $K$-path centrality measure that does not exactly use shortest paths but $K$ social hops. Betweenness centrality is also used to measure the influence of a node that controls the communication of another node [36]. Shetty and Adibi [138] explore email connections and set properties of dependency of one email on the other. In the sequence of email transactions [138], the second depends on the first with some conditions such as if both appear within the same time frame, if a major part is copied from one email to another, if that email is forwarded or if the links in the email are based on a certain event.

Wang et al. [155] perform risk assessment by mining the dependency of a source on a destination IP address using internet protocol traffic flow and presenting a flow dependency graph, with the source and destination data being the criminals’ IP addresses. Shaikh and Jiaxin [135] have proposed a dependency method for terrorist network disruption where they first isolate core nodes identified using combined centrality rank score (CCR); the result of the sum of degree, betweenness and closeness centrality value of each node. The nodes are then ranked using Eigenvector
centrality (EVC), Google PageRank and node dependency; taking the sum of a node’s dependency values using pair-wise dependency methods by Brandes and Freeman [20; 64] and arranging them by sum or column. Finally they [135] analyse the effect of the nodes with the highest score and remove these nodes to disrupt the terrorist network.

Google PageRank and Markov centrality are other measures used to rank and identify influential nodes. Random walks are used to calculate Markov centrality scores [157]. The centrality score of a node is calculated by first taking the average path length of a random walk that starts at that node and arrives (for the first time) at some other node, averaged over all other nodes. Then the inverse of the average distance between each node and every other node is used to score the centrality [157; 46]. The PageRank algorithm, used by Google to rank important web pages uses the assumption that a page is important when it is linked by many pages or if it is linked by many other important pages [82; 24]. The mathematical equivalent of this concept is the eigenvector centrality measure [82].

5.3 Preliminaries

This section includes the graph-theoretic terminology [72] for defining the betweenness centrality measures introduced by Freeman [64], Brandes [20] and Geisberger et al. [67]. Our research focuses on these three main definitions; for broader explanation and analysis, see Brandes [21] and Newman [115].

Let $G = (V, E)$ be a graph where $V$ is the set of vertices $V = \{v_1, v_2, \ldots\}$ (also called nodes) and $E$ the set of edges $E = \{e_1, e_2, \ldots\}$ (representing the connections between the vertices), with the total number of vertices and edges given by $|V| = n$ and $|E| = m$, respectively. An edge that has the same start node and end node is called a self-loop or a loop. If more than one edge is associated with a pair of nodes, these are called multiple edges. For our purpose, I exclude all self-loops and the multiple edges are considered as one edge.

A path is a sequence of edges that connects multiple nodes [115]. Given a path $(s, t)$, I call $s$ the source node and $t$ the destination, end node or target node. In between the source and the target, lies the alternating sequence of nodes and edges, for instance, $s, e_1(s, v_1), v_1, e_2(v_1, v_2), v_2, \ldots, e_t(v_i, t), t$. Here $e(u, v)$ denotes the edge connecting nodes $u$ and $v$. In the graph $G$, the length of an $(s, t)$-path is the number of edges it contains, and the distance, $\mu(s, t)$, from $s$ to $t$ is defined as the minimum length of any $(s, t)$-path if one exists and undefined otherwise [21]. Let the number of shortest paths from $s$ to $t$ be given by $\sigma_{st}$, and let $\sigma_{st}(v)$ be the number of shortest paths from $s$ to $t$ that pass through $v$. 
The pair-dependency $\delta_{st}(v)$ of a pair of nodes $s$ and $t$ on an intermediate node $v$ is the proportion of the shortest paths from $s$ to $t$ that contain $v$, that is:

$$\delta_{st}(v) = \frac{\sigma_{sv}(v)}{\sigma_{st}}. \quad (5.1)$$

The betweenness centrality of $v$ is then the sum of all such pair-dependencies [64]:

$$BC(v) = \sum_{s \neq v \neq t \in V} \delta_{st}(v). \quad (5.2)$$


The dependency of $s$ on an intermediate node $v$ is given by:

$$\delta_{vs}(v) = \sum_{w \in P_s(w)} \frac{\sigma_{sw}(1 + \delta_{vs}(w))}{\sigma_{sw}}. \quad (5.3)$$

Here $\{w : v \in P_s(w)\}$ is the set of all nodes $w$ where $v$ is an immediate predecessor of $w$ in a shortest path from $s$ to $w$, that is $v \in P_s(w)$.

According to Brandes [20], $\delta_{vs}(v)$, the dependency of $s$ on $v$ is positive, that is $\delta_{vs}(v) > 0$ only when $v$ lies on at least one shortest path from $s$ to $t$ and on any such path there is exactly one edge $\{v, w\}$ with $v \in P_s(w)$. Brandes’ algorithm has been used by researchers to measure the centrality of words from a group of texts [20], identify a central node for paths between metabolites or other molecules [77], identify the highest betweenness of an edge that separates two communities [148], etc.

In 2008, Geisberger et al. applied a linear scaling to Brandes’ algorithm by introducing the length function (unit edge weight) [67]. Geisberger et al. [67] thus introduced a better approximation for betweenness centrality, their main motivation being to apply the betweenness approximation measure on large networks without overestimating the values for the nodes near a pivot or parent node. In [67], Geisberger et al. state that Brandes’ algorithm [20] overestimates the betweenness values for the nodes near the pivot. To overcome this problem, Geisberger et al. proposed a linearly scaled betweenness method by adding a length function into the aggregation in Brandes’ scheme, thus giving a scaling for each value in the aggregation [67].

Given a shortest path from source $s$ to a node $w$ with node $v$ a predecessor of $w$ on this path, the length function is the ratio of the distance $\mu(s, v)$ of $v$ from $s$ to the distance of $w$ from $s$, $\mu(s, w)$. Thus Brandes’ algorithm changes to:
\[
\delta_{st}(v) = \sum_{w : v \in P_{s}(w)} \mu(s, v) \left( \frac{\sigma_{sv}}{\sigma_{sw}} \times (1 + \delta_{st}(w)) \right). \quad (5.4)
\]

With this change to Brandes’ algorithm, Geiseberger et al., could apply betweenness centrality to real world situations such as choosing improved highway-node routings \cite{68, 67}. While Brandes gives good exact results for small networks, often it is not possible to get exact results in reasonable running time for large networks. The Geisberger et al. formula gives better approximations in relation to large networks such as dynamic highway-node routing.

### 5.4 The reliance measure

In our proposed reliance formula, I start with a specific set of source nodes \( \{s_1, s_2, \ldots\} \) and calculate the reliance (a.k.a. ‘dependency’) value for each intermediary node \( v \) that occurs on paths starting from each \( s_i \) to all possible end nodes \( t \). Thus, I consider all shortest paths from a particular source \( s \) to all possible end nodes \( t \) where \( v \neq t \). In the first part of the formula, I calculate the proportion of the shortest paths linking source \( s \) to all nodes \( t \) that contain \( v \).

**Definition 4** (Importance Rate). Given a graph \( G = (V, E) \), \( s \) a source node and \( v \) an intermediate node on some path \( (s, t) \) from \( s \) to an end node \( t \), the importance rate \( IR_{(s,t)}(v) \) measures the importance of \( v \) to \( s \) such that \( s \) may continue communicating with \( t \) and is given by:

\[
IR_{(s,t)}(v) = \delta_{st}(v). \quad (5.5)
\]

Here, \( \delta_{st}(v) \) is the pair dependency of \( s \) and \( t \) on \( v \) as given in Equation (1).

I have named this pair dependency as the importance rate of \( v \) as it indicates how often the node \( v \) is relied on to complete a path to reach the destination \( t \) in proportion to all paths from \( s \) to \( t \). The second quantity I introduce is a ‘trust’ value that the source \( s \) places on \( v \) to pass messages to any \( t \) along the shortest paths.

**Definition 5** (Trust). Given a graph \( G = (V, E) \), \( s \) a source node and \( v \) an intermediate node on some path \( (s, t) \) from \( s \) to an end node \( t \), the trust of \( s \) on \( v \), relative to the path \( (s, t) \) denoted by \( T_{(s,t)}(v) \), is given by:

\[
T_{(s,t)}(v) = \frac{\mu(s, v)}{\mu(s, t)}, \quad \text{for } t \neq s, s \neq v \neq t, \quad (5.6)
\]

where \( \mu(s, u) \) is the minimum length from \( s \) to \( u \) along the path \( (s, t) \), \( u \in V \), if one exists and undefined otherwise.
This trust concept is illustrated with an example. Figure 5.1 shows a small network.

Figure 5.1: A small network. This network has 8 nodes and 8 edges. The floating-point numbers near to each node represent the trust value of source node 1 on each intermediate node \{2, 3, 5, 7\} in the path 1 → 2 → 3 → 5 → 7 → 8.

In the graph in Figure 5.1 let 1 be the source node and 8 the destination or end node. A path from source 1 to destination 8 is:

\[ 1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 8 \]

Verbiest et al. [151], [152] in their research incorporating path length and trust aggregation mention that the shorter the distance from source \(s\) to \(v\), the more the source \(s\) trusts \(v\). De Meo et al. [51] design an algorithm to compute edge centrality using k-path length with an assumption made that the influence between two nodes reduces when the distance between them increases. Indeed, a common way to start a criminal investigation process is to identify the closest node to a criminal or source because the shorter the distance from the source, the higher the chances the node is the source’s subordinate [104]. I claim that in the context of money laundering it can be the longer the distance from a source to an intermediate node, the more the source needs to trust that node to make the next financial transaction. Thus, if node 1 in Figure 5.1 is the source node, then the trust of source node 1 on node 7 to pass money on to destination node 8 (\(T_{1,8}(7) = \frac{4}{5}\)) at distance 4 from node 1 is higher than the trust on node 3 (\(T_{1,8}(3) = \frac{2}{5}\)) at distance 2.

Figure 1 could represent the layering of illegal money within a money laundering syndicate, with node 1 the money laundering suspect. The layering process involves multiple transactions of money through various channels [13, 14]. In such a money layering process, money below the threshold is distributed to different financial institutions or accounts. This contributes to the growth in the length of the paths that are used to transport the money from a source to a destination. The source uses longer sequences of channels to divide and distribute smaller amounts of money making it more difficult for law enforcement authorities to identify the influential people.
I propose a method to identify the node that the source relies on the most to detect key people for further investigation.

It is difficult to obtain actual financial transaction data from banks or other financial institutions due to a strict privacy policy, thus I continue to use the Enron email network to identify key nodes via the proposed reliance formula, concentrating on only the static properties of the emails and not on their contents. To rank the nodes important to the source, I incorporate the importance rate and the trust value of an intermediate node to the source to define the reliance of the source $s$ on an intermediate node $v$.

**Definition 6** (Reliance). Given a graph $G = (V,E)$, with $|V| = n$, a source node $s$ and an intermediate node $v$ on some path $(s,t)$ from $s$ to an end node $t$, the reliance of $s$ on $v$ along the path $(s,t)$, $r_{(s,t)}(v)$, is the product of the importance rate $IR_{(s,t)}(v)$ and the trust $T_{(s,t)}(v)$:

$$r_{(s,t)}(v) = \delta_{st}(v) \times \frac{\mu(s,v)}{\mu(s,t)}, \text{ for } v \in (s,t), t \neq s, s \neq v \neq t,$$

(5.7)

Finally, the total reliance of source $s$ on $v$ over all paths from $s$ to all possible end nodes $t$, is:

$$R_{(s)}(v) = \sum_{v \in (s,t),t \neq s,s \neq v \neq t} r_{(s,t)}(v) \frac{1}{(n-2)}.$$

(5.8)

Here, $|V| = n$, that is, there are $n$ vertices in the graph, and since the start node $s$ and the intermediate node $v$ are fixed and I take $\delta_{st}(v)$ for all $t \neq s, s \neq v \neq t$, there are $(n-2)$ possible values for $t$. Thus, I normalise the reliance value, $R_{(s)}(v)$ with $(n-2)$.

### 5.5 Comparison between different dependency techniques

In this section, I compare the reliance model to the dependency techniques of Freeman [64], Brandes [20] and Geisberger et al. [67]. The graph in Figure 5.2 shows the dependency or the reliance value of node 1 on other intermediate nodes of the network in Figure 5.1.

**Example 1.** Difference between dependency techniques and reliance formula
Figure 5.2: The result of applying the different dependency formulae and the reliance formula on the network shown in Figure 5.1. Here, for the purpose of comparison I choose the source as node 1 and take all shortest paths from this source node to all possible end nodes. The value of reliance of node 1 on other intermediate nodes is compared to the other dependency values.

The dependency and reliance of node 1 on itself is zero. Similarly, the reliance of node 1 on node 8 is also zero because there is no shortest path from source 1 to any possible end nodes that contains 8 as an intermediate node. The biggest difference that is obvious from the graph in Figure 5.2 is the reliance value of node 1 on node 3. Freeman and Brandes’ results show the dependency value of node 3 as less than that of node 2. Both the Geisberger et al. and the reliance methods show that node 1 relies more on node 3 than on node 2. The network graph shown in Figure 5.1 is undirected and node 3 is at the crucial position of separation between two groups. Thus, based on the position of the nodes, the dependency or the reliance value of node 1 on node 3 should be higher.

Geisberger et al., through their experiments in [67], also show that their technique gives a better estimation of betweenness value when compared to Brandes’ algorithm. The ranking pattern of Geisberger et al.’s technique and our reliance model is the same and the dependency or the reliance value of node 1 on node 3 is the highest. Although the ranking is the same, the technique by Geisberger et al. is different from our reliance formula as the reliance calculation focuses on the reliance of a specific source on other nodes. Another difference is that the dependency estimation value of Geisberger et al.’s technique gets higher than our reliance model when the graph gets bigger.

Example 2. Difference between the reliance formula and Geisberger et al.’s formula.
Figure 5.3: A sample graph. This sample graph is used to compute the dependency of node 1 on other nodes.

Figure 5.3 shows a network with 8 nodes. I calculate the dependency of node 1 on other nodes using Geisberger et al.’s technique.

\[
\delta_{s=1s}(v) = \sum_{w:v \in P_s(w)} \frac{\mu(s,w)}{\mu(s,w)} \left[ \sigma_{sv} \times (1 + \delta_{ss}(w)) \right],
\]

\[
= \left\{ \sum_{v \in (s,t), t \neq s, s \neq t} \left[ \frac{\mu(s,v)}{\mu(s,t)} \times \delta_{st}(v) \right] \right\} + \left\{ 2 \left\{ \frac{\mu(s,2)}{\mu(s,7)} \sigma_{s2} \right\} + 2 \left\{ \frac{\mu(s,2)}{\mu(s,7)} \sigma_{s7} \right\} + 5 \left\{ \frac{\mu(s,2)}{\mu(s,8)} \sigma_{s8} \right\} \right\},
\]

\[
= R_s(v) + 2 \left\{ \frac{\mu(s,2)}{\mu(s,6)} \sigma_{s6} \right\} + 2 \left\{ \frac{\mu(s,2)}{\mu(s,7)} \sigma_{s7} \right\} + 5 \left\{ \frac{\mu(s,2)}{\mu(s,8)} \sigma_{s8} \right\} + 2 \left\{ \frac{\mu(s,2)}{\mu(s,7)} \sigma_{s7} \right\} + 5 \left\{ \frac{\mu(s,2)}{\mu(s,8)} \sigma_{s8} \right\} \right\},
\]

(5.9)

The calculation using the formula by Geisberger et al. is composed of our reliance value plus extra estimation for nodes 6, 7 and 8. This extra estimation:

\[
\left\{ 2 \left\{ \frac{\mu(s,2)}{\mu(s,6)} \sigma_{s6} \right\} + 2 \left\{ \frac{\mu(s,2)}{\mu(s,7)} \sigma_{s7} \right\} + 5 \left\{ \frac{\mu(s,2)}{\mu(s,8)} \sigma_{s8} \right\} \right\}
\]

is due to the repetitive dependency calculation of the source node on nodes 6, 7 and 8. Thus the direct use of the Geisberger et al. formula for the purpose of calculating reliance would result in an overestimation of the value of certain nodes. While Geisberger et al.’s formula is a good means of estimating values for large networks, our purpose here is to further refine a possible network for criminal investigation. The next section consists of the application of the reliance formula to two different case studies, the Enron Network and Noordin Top Terrorist Network.
5.6 Case Study 1: Enron Network

In Chapter 4, I first divided the Enron email dataset, which consists of 1,887,305 email transactions into two groups; emails that were sent using the fields ‘TO/CC’ and ‘BCC’ due to its enormous size. The SPNSA implementation in Chapter 3 was based on only the ‘BCC’ network to identify a trust network, using money laundering criminals as the algorithm feed. The ‘BCC’ email transactions by nature contains recipients that are kept secret [61]. In this chapter, both the ‘TO/CC’ and ‘BCC’ undirected email transaction networks are used to produce reliance sub-networks for all suspects. Through dividing the email transactions, I am able to compare the important nodes that a suspect relies on, whether or not the connection (email) is kept secret (bcc-ed). To form an undirected network, I make the broad assumption that an email sent from A to B or from B to A implies an undirected relationship between A and B. I give more importance to the sequence of emails exchanged by the nodes rather than the content of the emails.

As feed for the SPNSA, I choose certain employees of Enron, as in Chapter 4, based on the possibility of them being involved in money laundering and they are henceforth called suspects. The first set of suspects consists of the Enron finance managers [6] (see Table 4.10) while a larger list comprises all top managers [6] (see Table 4.9) plus the finance managers. As mentioned previously, a manager may be indexed by more than one node if he or she has more than one email account. Both the email transaction groups, the ‘TO/CC’ and ‘BCC’, have distinct ID sets for each email address and this is designed as such because some nodes that exist in the ‘BCC’ group do not exist in the ‘TO/CC’ group. The network formed using the ‘BCC’ email transaction subset has 19,716 nodes and 65,532 edges while the network formed using the ‘TO/CC’ email transaction subset has 26,027 nodes and 252,863 edges. All self-loops and multiple edges have been removed from these networks.

5.6.1 Node ranking for on the Enron ‘BCC’ network

Here, I first compare the node ranking results of betweenness centrality proposed by Brandes (Equation 5.3) and Geisberger et al. (Equation 5.4) with the results produced using the reliance measure (Equation 5.8) on the Enron ‘BCC’ network. The total reliance of all suspects on each \( v \) in the respective BCC sub-network is used for this comparison. The nodes are ranked based on the decending order of betweenness centrality (Brandes’ and Geiseberger et. al) values.
Enron finance manager BCC sub-network node ranking

Five Enron employees who worked as finance managers with the Enron company between the years 1990-2001 were picked as the SPNSA’s feed. The finance managers that are used as the algorithm feed are Andrew Fastow (686, 687), Sherron Watkins (16929), Ben Glisan (1369), Rick Causey (15077) and Jeff McMahon (8071) (see Table 4.10). The sub-network generated by SPNSA using this feed is named as the Enron Finance Manager BCC sub-network and has 30 nodes and 53 edges. The betweenness centrality measures by Brandes and Geisberger et al. calculates the centrality value from all sources on \(v\). To compare the reliance value with the betweenness centrality measures, I take the total reliance of all finance managers, \(F = \{686, 687, 16929, 1369, 15077, 8071\}\), on each \(v\) in the Enron finance manager BCC sub-network. The total reliance value for each \(v\) is thus \(\sum_{s \in F} R(s)(v)\). This compares well with the betweenness centrality value (in the case of both Brandes and Geiseberger et al.) for \(v\) which is \(\delta_s * (v)\) for all \(s\) in the sub-network, and not just \(s \in F\) and hence, illustrates clearly the need for a more specific measure than the betweenness centrality measure. The comparison of node ranking is depicted in Figure 5.4 for nodes that the finance managers rely on.

![Image of node ranking](image)

**Figure 5.4:** The Enron finance manager BCC sub-network node ranking. The total reliance (Equation 5.8) of all finance managers (686, 687, 16929, 1369, 15077, 8071) on each \(v\) in the Enron Finance Manager BCC sub-network is used for this comparison. The nodes are ranked based on the descending order of betweenness centrality (Brandes’ (Equation 5.3) and Geiseberger et al (Equation 5.4)) values. There is a clear difference in node ranking between the betweenness centrality measures of Brandes and Geisberger et al. (there is no difference between them) and the total reliance. For example, node 2473 is more heavily relied on by the finance managers than 686, where as both the betweenness centrality measures rank them the other way around.

Highly ranked nodes can be important nodes for a primary investigation. To compare the results with the reliance measure, I normalise both sets of the betweenness centrality results of Brandes and Geisberger et al. by dividing each node’s betweenness value with the maximum value in each set respectively. In Figure 5.4, all three measures identified the same node as having the highest value. The interesting point is to note that some of the nodes (for example, 11010, 12935 etc.) ranked lower by the betweenness centrality measures, are ranked as being more important.
by the reliance measure. One node in this list, node 11010 (lfastow@pop.pdq.net) belongs to Lea 
Fastow who was convicted as an Enron money laundering criminal [23].

Reliance ranking is also compared with the other two ranking values; PageRank and Markov 
centrality. The results are shown in Figure 5.5. The nodes are ranked based on the descending 
order of PageRank scores.

![Figure 5.5: The Enron finance manager BCC sub-network node ranking.](image)

The bar chart in Figure 5.5 shows the differences in node ranking using Markov centrality, 
PageRank and the reliance measure. Some nodes (3983, 687 and 15077) are valued by the Markov 
centrality and PageRank but not relied on at all by the finance managers (see figure 5.5 towards 
the end of the bar chart). Similar to the betweenness centrality measures in Figure 5.4, PageRank 
and Markov centrality ranking were not able to pick 11010 (Lea Fastow), who is not in the finance 
manager list, as important, in contrast to the reliance formula.

**Enron manager BCC sub-network node ranking**

The same method is repeated for all the Enron managers (finance and other top managers); a 
larger algorithm feed to see again if there is a difference in ranking. The list of finance managers 
can be found in Table 4.10 and other top managers in Table 4.9. A shortest paths sub-network 
is formed using all these managers as the SPNSA feed and is named as the Enron Manager BCC 
sub-network. This is an undirected graph and has 121 nodes and 314 edges. First the nodes in 
this sub-network are ranked based on the descending order of betweenness centrality values.
Figure 5.6: The Enron manager BCC sub-network node ranking. Three different methods that is betweenness centrality by Brandes (Equation 5.3), by Geisberger et al. (Equation 5.4) and reliance value using reliance measure (Equation 5.8) are used to show the difference between node ranking. Note that only the nodes with significant reliance value are displayed here.

Even more than in Figure 5.4, Figure 5.6 shows that the betweenness centrality measures give a different ranking; the most important node (17697) that was picked by the reliance formula is different from the node (15932) valued the most by Brandes and Geisberger et al. The most important node (15932) valued by Brandes and Geisberger et al. is one of the least scored nodes by the reliance measure and vice versa. Moreover, the betweenness centrality values for nodes 348, 441 and 9395 are almost the same whereas the reliance measure shows dramatically different rankings. These differences could allow an investigator to pick suitable people for further investigation. Next I applied Markov centrality and PageRank method to compare with the Enron manager’s node reliance ranking. The result is shown in Figure 5.7. The nodes are ranked based on the descending order of PageRank values.

Figure 5.7: The Enron manager BCC sub-network node ranking. The total reliance of all managers on each \( v \) in the Enron Manager BCC sub-network is used for this comparison. The nodes are ranked based on the descending order of PageRank values. There is a clear difference in node ranking using the three different methods. For example PageRank, Markov centrality and the reliance measure ranks node 19075, 6673 and 17697 as having the highest value respectively.
It is clear that an investigation using either the betweenness centrality, the PageRank or the Markov centrality ranking, as opposed to the reliance ranking, will lead to a different outcome in terms of possible people to investigate. For example, note that Lea Fastow is not in the algorithm feed for the experiment using all managers or finance managers but Figure shows that Lea Fastow, the wife of Andrew Fastow and one of the money laundering criminals, is heavily relied on by the finance managers.

5.6.2 Extracting a suspect’s reliance sub-network from the Enron Network using the reliance measure

Here, I show the results of identifying the Enron money laundering suspects’ important nodes based on suspect-intermediate reliance value and the steps to extract suspects’ reliance sub-network. Both the Enron ‘BCC’ and ‘TO/CC’ email transactions were used here. The finance managers are used as the algorithm feed followed by the top managers. The shortest paths network formed using the SPNSA with 5 finance managers as the algorithm feeds is shown in Figure 5.8.

Figure 5.8: The Enron Finance Manager ‘BCC’ sub-network formed using the SPNSA and only emails that have recipients in BCC field. The finance managers that are used as the algorithm feed are given in Table 4.10. The finance managers who are captured in this shortest path network are highlighted in red. This is an undirected graph with 30 nodes and 53 edges.

I calculate the reliance of each finance manager as source node on the intermediate nodes in the network in Figure 5.8. In the Table 5.1, I show the nodes that the finance managers rely on the most according to our reliance measure.
Table 5.1: Nodes that the finance managers rely on the most in the Enron Finance Manager BCC sub-network

<table>
<thead>
<tr>
<th>Finance Managers (FM)</th>
<th>The node that the FM relies the most</th>
</tr>
</thead>
<tbody>
<tr>
<td>686</td>
<td>1369</td>
</tr>
<tr>
<td>687</td>
<td>2473</td>
</tr>
<tr>
<td>16929</td>
<td>15406</td>
</tr>
<tr>
<td>1369</td>
<td>686</td>
</tr>
<tr>
<td>15077</td>
<td>19075</td>
</tr>
<tr>
<td>8071</td>
<td>2473</td>
</tr>
</tbody>
</table>

Table 5.1 shows the nodes that the finance managers rely on the most based on our reliance measure.

The nodes 1369, 2473, 15406, 686 and 19075 represent email addresses that are named as the persons of interest (based on our reliance value ranking) to the finance managers. The next step in our analysis is to take these nodes that the finance managers rely on the most and use them as a new feed in the SPNSA to continue our search for an investigative sub-network. The sub-network formed using these crime priority nodes is depicted in Figure 5.9.

Figure 5.9: Second phase Finance Manager BCC sub-network: This sub-network is formed using the nodes (1369, 2473, 15406, 686 and 19075). These nodes (highlighted in red) are the ones that the finance managers rely on the most. This is an undirected graph and has 19 nodes and 32 edges.

On applying the reliance algorithm on the sub-network in Figure 5.9, I obtained 4 other important nodes (1369, 10917, 15406 and 19075) to investigate. The process is continued by using these 4 nodes as the SPNSA feed for the third phase, with the persons of interest of the third
phase being (7971, 15406). There is no intermediate node between 7971 and 15406, thus there is no further person of interest and the process stops. Finally, using the persons of interest obtained at each phase, the finance managers’ reliance sub-network for investigation looks like the network in Figure 5.10.

Figure 5.10: **Enron finance managers’ reliance ‘BCC’ sub-network**: The figure above shows a sub-network formed using our reliance measure algorithm repetitively. The finance managers are used as the first phase feed to the SPNSA, followed by the next set of nodes that are of the highest reliance to the finance managers. These nodes are called the finance managers’ persons of interest and they are, in turn, used as the third phase feed and so on. The process stops when the same nodes obtain the highest reliance value or when there is no out-component from the persons of interest, resulting finally in a hierarchal network as in this figure.

Next, the same method is repeated for the Enron managers’ shortest paths network shown in Figure 5.11.

Figure 5.11: The Enron Manager BCC sub-network formed using the SPNSA and only emails that have recipients in BCC fields only. The feed for the algorithm are all the Enron managers including finance managers; (see Table 4.9 and 4.10). Here, I assume that I do not know the criminal list and I use all managers even though some happen to appear in the criminal list. The managers who are captured in this shortest path network are highlighted in red. This is an undirected graph and has 121 nodes and 314 edges.

As in the previous case with the finance managers, first I find the nodes that the managers rely on the most using the reliance measure, then repeat the steps until the nodes that I feed in have the same person that they rely on the most or don’t have any out-component, giving the Enron managers’ reliance ‘BCC’ sub-network as depicted in Figure 5.12.
Figure 5.12: Enron managers’ reliance ‘BCC’ sub-network formed using the nodes that all top managers and finance managers from the BCC network rely on the most.

In this way an investigator may be able to identify the list of crime priority nodes (based on the reliance ranking) to the suspects and use these persons of interest for further investigation. This experiment will help to form a hierarchical network that eases the work of an investigator to identify the important people and their connections for a criminal investigation.

Next, I apply the SPNSA and the reliance formula to explore the Enron ‘TO/CC’ email network. Similar to the above experiments, I used the finance managers and then all managers’ as the suspects. Figure 5.13 shows the reliance sub-network for the finance managers in the Enron ‘TO/CC’ email network.

Figure 5.13: Enron finance managers’ reliance ‘TO/CC’ sub-network: The figure shows a sub-network formed using the nodes that all finance managers in ‘TO/CC’ network rely on the most. The finance managers [6] that are used as the algorithm feed are given in Table 4.10. Note that some employees’ email addresses that exist in the ‘BCC’ network do not occur in the ‘TO/CC’ network, for example email address (andrew.fastow@ljminvestments.com) that belongs to Andrew Fastow occurs only in the ‘BCC’ network. Thus, the use of separate ID for the nodes in both the networks.

The following, Figure 5.14 is the result of applying the reliance method using all managers.
5.6.2.1 Discussion

The process of searching for the reliance sub-network allows us to find the people that the finance managers or all managers rely on the most. On comparing the reliance sub-network of finance managers in ‘BCC’ network (see Figure 5.10) and ‘TO/CC’ network (see Figure 5.13) I found that more people were relied on by the finance managers in the bcc-ed email transactions than in the ‘TO/CC’. In addition the person that the finance managers relied on in the BCC network was different to the ‘TO/CC’ network.

However, the number of persons of interest that all managers rely on in the ‘BCC’ network (see Figure 5.12) is less than in ‘TO/CC’ network (see Figure 5.14). Three people appear to be the persons of interest in both the Enron managers’ ‘BCC’ and ‘TO/CC’ reliance sub-network. They are Jeff Dasovich (‘BCC’ ID 7971, ‘TO/CC’ ID 12868), Mike Mcconnel (‘BCC’ ID 12935, ‘TO/CC’ ID 20582) and Greg Whalley (‘BCC’ ID 6673, ‘TO/CC’ ID 10836). Here Greg Whalley seems to be relied on by more managers in both the ‘BCC’ and ‘TO/CC’ network.

The experiment shows that finance managers rely on more people in the ‘BCC’ network compared to the ‘TO/CC’ network while it is the other way around when the test is done with all the managers. By its very nature, ‘BCC’ email transactions contain recipients that are kept secret. This shows finance managers kept more recipients secret in their daily email conversations. The proposed method used to extract the reliance sub-network could reduce the time needed for exploring all the nodes in a large network and help to speed up the investigation process. The method shown in the experiment above is adaptable to other types of transaction data, for example, it could be applied to cell phone communication transactions; calls or text messages, financial transactions, goods and product transactions, etc. In the second case study in the next section, I
explore Noordin Top data set, that contains terrorist relationships [57].

5.7 Case Study 2: Noordin Top Terrorist Network

The Noordin Top terrorist network [57] dataset consists of different types of criminal connections and for this thesis I focus on the terrorist-people relationship information such as classmates, soulmates, kin and friends. Two different subsets from this dataset were used for the reliance formula analysis.

I look particularly into the past research that applies social network analysis on the Noordin Top Terrorist Network as this data has been widely used for forming communities and to track changes in network structure over time. Some examples are discussed here. For the purpose of monitoring and disrupting dark networks [58], four types of networks; a trust network that contains direct connections, an operational network that shows the participation of terrorists in meetings, operations, etc., a communication network and a business & finance network were all combined. Changes in the network structure were monitored by capturing terrorists’ occurrences over time based on three different measures; density, degree and betweenness centrality, and normalised standard deviations of the centrality scores [58]. Noordin Top Terrorist affiliation community analysis was conducted by Alzahrani and Horadam [5] using an existing random walk based algorithm Infomap that is found to be able to identify criminal cliques. In other research, various measures such as node degree, edge overlap, node participation in different layers, clustering coefficient, reachability and eigenvector centrality were used to characterize multiplex networks in this dataset [16].

In addition to this, terrorists in this dataset have been ranked using some key node ranking techniques. Fox and co-authors [63; 62] rank the terrorists by first calculating the global decision weight of certain decision criteria; e.g. role in organisation, closeness, degree centrality etc. using Analytic Hierarchy Process (AHP) method [62]. They [62] then apply TOPSIS (Technique of Order Preference by Similarity to Ideal Solution) [163] to score the decision criteria and make matrix comparisons of these criterias’ scores to rank the terrorist targets. Key players in this dataset were also predicted by Butt et al. [31], incorporating certain centrality measures (degree, betweenness, closeness centrality and eigenvector) with some classifying techniques; k-nearest neighbours (kNN), Gaussian mixture model (GMM) and support vector machine (SVM). Liebig and Rao [95] use the bipartite clustering coefficient to find important nodes in this multi structured terrorist network. Brown [30] in his thesis finds key players of this terrorist network by integrating some centrality measures (betweenness, closeness, and total degree) into a network fragmentation and diffusion process.
In this chapter, I rank the terrorists using source-intermediate reliance value. A list of source nodes or the suspects from each group of connections are extracted randomly using a random node generator. In a crime investigation process, suspect nodes would be the people who are suspected for involvement in a crime. For example in the first case study in section 5.6, I used finance managers or all top managers in assumption of having committed money laundering. The reason for having a specific seed in that case (Section 5.6) was the enormous size of the BCC network (19,716 nodes and 65,532 edges). In the case of Noordin Top terrorist network, the size of the network is much smaller (≤ 61 nodes and < 200 edges), and, in addition, the network contains only criminals and their connections [57]. Thus, I use a random node generator assuming that the generated nodes (algorithm feed) are a mix of criminals and other people. First I use the SPNSA with the random feed to extract sub-networks with possible people to investigate and then I use the reliance formula to identify the crime priority nodes for the reliance ranking application.

While no particular set of suspects was picked to form a criminal reliance sub-network in this case, with a set of suspects, the method would allow an investigator to use the obtained terrorist reliance sub-network as an aid for terrorist investigation. Next, in subsection 5.7.1, I use the terrorist-friendship and terrorist-classmate networks only for the purpose of node ranking.

### 5.7.1 Node ranking for the Noordin Top terrorist network

I present 4 different bar charts here to show the network node importance value calculated using the betweenness, Markov centrality, PageRank and the reliance measure.

(a) The terrorist-friendship sub-network node ranking using betweenness centrality and reliance measure.

(b) The terrorist-classmate sub-network node ranking using betweenness centrality and reliance measure.

Figure 5.15: The total highest reliance of all nodes in the terrorist-friendship and terrorist-classmate sub-network on each \(v\) is used for this comparison. For figures a and b, the nodes are ranked based on the descending order of Brandes’ betweenness centrality values and the reliance and betweenness values are shown for different subsets of terrorists, that vary in size and membership. For each graph, terrorists with zero reliance and ranked by betweenness (Brandes) values as below the terrorist with the lowest reliance, are not included.
(a) The terrorist-friendship sub-network node ranking using Markov centrality, Pagerank and reliance measure.

(b) The terrorist-classmate sub-network node ranking using Markov centrality, Pagerank and reliance measure.

Figure 5.16: The total highest reliance of all nodes in the terrorist-friendship and terrorist-classmate sub-network on each \( v \) is used for this comparison. For figures a and b, the nodes are ranked based on the descending order of PageRank scores and the reliance, Markov centrality and PageRank scores are shown for different subsets of terrorists, that vary in size and membership. For each graph, terrorists with zero reliance and ranked by PageRank scores as below the terrorist with the lowest reliance, are not included.

The terrorist-friendship sub-network consists of 61 nodes and 91 edges and the terrorist-classmate sub-network has 39 nodes and 175 edges. First I isolated the highest reliance of each node on the intermediate node in the shortest paths between that node and all other nodes in the sub-network. Then all the reliance values of the same intermediate node were summed and normalised with the highest value in the list. The terrorist node importance level comparison experiment shows that our reliance formula is able to present variations in the importance level of nodes when most nodes that are ranked by the other methods in this chapter have similar values. The differences in the rankings here are even more extreme than for the Enron dataset.

5.7.1.1 Discussion

The difference in ranking of terrorist nodes in the section above proves that each of the individual relationships; terrorist-friendship, terrorist-soulmates, terrorist-kinship and terrorist-classmates can be explored separately to obtain the crime priority nodes (persons of interest). Similar to the reliance sub-network shown in the Enron case study above, the terrorist reliance sub-network can be identified. A path in the terrorist reliance sub-network is the sequence of people who could be engaged in a particular terrorist mission or attack. However, in this thesis, no target suspects were picked to match any particular terrorist activity thus I have not shown the suspects’ source-intermediate reliance sub-network here. Note that in a real world application, first extracting the important nodes using the reliance measure is essential and then forming the terrorist reliance sub-network can aid to identify suspects’ reliance connections.
5.8 Conclusion

The work presented in this chapter introduces a new reliance measure to rank nodes in a network and therefore identify nodes of interest. This reliance measure is different from betweenness centrality because the betweenness centrality measure calculates the centrality value of all sources on a node whereas the reliance measure calculates the reliance of a list of specific sources on a node. I compare the reliance ranking with other centrality measures such as Google PageRank, Markov centrality and betweenness centrality. Reliance identifies a very different subset of nodes from those identified by the other measures. The ranking based on the reliance measure can also be used to identify the nodes that particular persons of interest rely on most heavily.

The SPNSA (see Chapter 3 section 3.3.1) is able to produce a small and manageable network. In this chapter, I extended the research and analysed the connections between nodes in the network; reliance of one node on another that leads to the identification of important nodes. These important persons are used to form a reliance sub-network that aids in further investigation. In our experiment using the Enron dataset as the first case study, the reliance sub-network of the finance managers shows that the number of priority nodes of the finance managers in the ‘BCC’ network is more than in the ‘TO/CC’ network. This reveals that the finance managers’ connections are concentrated more in the ‘BCC’ network than in the ‘TO/CC’ network. In the second case study, I applied the reliance formula on the Noordin Top terrorist network and distinguished important nodes.

This reliance method could reduce the time needed for exploring nodes in a large network and hence may speed up an investigation process. For the best outputs, prior to the application of the shortest paths network search algorithm and the reliance formula, it is essential to choose the most relevant first suspects to a crime. The analysis would also yield more criminal to criminal connections from choosing the most applicable or appropriate algorithm feed to a crime incident.
Chapter 6

Conclusion

This research involved the study and the application of complex network analysis to solve the problems of dealing with large numbers of nodes and their complex relationships especially with regards to money laundering networks. The detailed literature review shows that an illegal network is usually embedded in daily operation as links the sum of which becomes complex and the discovery of the hidden criminal community turns into a difficult task. This thesis focused on money laundering criminal community detection. Previous research illustrates that money laundering detection is highly dependent on knowledge based techniques and manual processes to identify illegal money transfers. In this thesis, instead of using financial data that is difficult to obtain due to confidentiality concerns, I found a publicly available dataset that consisted of email transactions, and included persons convicted of money laundering. I further explored the criminals’ complex relationships that lay in the email network and that had not previously been shown by data mining techniques. A criminal can disguise and escape being caught for an illegal money transfer by not exceeding the threshold amount but this research proves that an investigator is able to reveal the links between the criminals and their accomplices through an email network. However, the email database that was obtained for this research is large and tracking a criminal association from this large set of relationships is tedious, time consuming and has been done manually in the past by other researchers.

To overcome this problem, I developed an algorithm to extract the relationships between criminals and represent them in a network graph. This algorithm, named the shortest paths network search algorithm (SPNSA) is built by integrating network centrality measures and shortest paths. SPNSA requires a set of algorithm feed that consists of criminals or suspects to a crime incident. The network extracted through SPNSA is able to show all the connections that are associated with the list of criminal or suspects who are concentrated within the highly connected people (meaning the central people in a network). In this research, the SPNSA is applied to the
Enron email dataset to retrieve the connections of the Enron money launderers. The list of the Enron money launderers was obtained through the available online news articles and published papers.

At the beginning of this research, the statistical analyses applied to the Enron email dataset assisted in narrowing the research focus to only the BCC email network, that is, emails where 1 or 2 recipients were bcc-ed. The ‘BCC’ email network is a directed network that comprises of the connections from senders to recipients some of whom are concealed from the other recipients, thus implying a trust relationship. The Enron money laundering criminals were used as the algorithm feed for the SPNSA to extract money launderers’ trust connections from this large ‘BCC’ network. The result shows that the SPNSA is able to extract 3 criminals from the directed 1-BCC network (emails with 1 recipient bcc-ed) and 6 criminals from the directed 2-BCC network (emails with 2 recipients bcc-ed). These criminals are 5 edges away from the central nodes leading to the conclusion that the location of these criminals is within the highly linked people. However the main aim was to see if the SPNSA could retrieve criminals without knowing who they were. In this case, when the investigator only suspects that there is money laundering activity, I suggested the use of first suspects; the employees or people closely connected with Enron account management or other top managers. Here, I assume that the criminals belonging to either of these suspect lists are not known by the investigator.

This sort of feed is not possible with any known community detection algorithms. Thus, the effectiveness of the SPNSA is compared to the existing community detection algorithms and k-Neighbourhood approach, to the results using a different algorithm feed for SPNSA and applying both the community detection algorithms and the SPNSA to larger subsets of the Enron email data; undirected ‘BCC’ and ‘TO/CC’. Different algorithm feed represents different scenarios. In this evaluation, when all criminals are known and when the investigator only suspects that certain people are involved in a crime are investigated. Then a new scenario is added to this algorithm evaluation, that is when the investigator fails to detect one of the criminals. The existing community detection algorithms and the K-neighbourhood approach produced large and complex sub-networks, while the SPNSA extracted sparse and small sub-networks that could feasibly identify possible people for further investigation. The SPNSA validation without criminal information identified 4 convicted criminals in the resulting sub-network and the test where an investigator fails to identify one criminal, the SPNSA was successful in retrieving the left out criminals in 5 out of 9 cases. I conclude that the SPNSA’s performance is far more efficient when compared to existing community detection algorithms and the k-Neighbourhood approach. I further conclude that SPNSA using relevant algorithm feed related to a crime incident, gives a
network of connections that are significant. This is partially validated by using random feeds, the results of which show the probability of retrieving criminals from a large network using random nodes as the algorithm feed is very low.

Besides extracting possible people to investigate from a large network, I further identify important nodes to a criminal or suspect in the shortest paths network formed. A common method of identifying important nodes is by ranking on the basis of node centrality measures. Suspiciousness in a communication network can be highly dependent on the number of times a person is contacted, the number of meetings involving this person and also the location of this person. When communication links are seen as a network graph, these attributes of real life can be measured as the number of times a node occurs between criminals and other nodes or the distance of a node from these criminals. In this research, I proposed a method of identifying the importance of a node, given a set of nodes of interest. In order to track important people in a criminal path that connects a criminal to its acquaintances, I explored the mathematical concept of pair-dependency on the intermediate nodes, adapting the concept to criminal relationships and introducing a new source-intermediate reliance measure. For the purpose of illustration, besides the Enron ‘BCC’, ‘TO/CC’ email transactions, the Noordin Top Terrorist network was also used. I compared the performance of the reliance measure with other importance ranking methods such as Google PageRank, Markov Centrality as well as betweenness centrality. The results show that the reliance measure led to a different prioritisation in terms of possible people to investigate. I also used this reliance measure to form a new network; a criminal reliance sub-network for further investigation. For this, the nodes with highest reliance that occur between the criminals or suspects and other nodes are gathered. These nodes become a set of new suspicious nodes and they are repeatedly used as algorithm feed in the SPNSA to find new communities. The network search will stop when there is no link found from the suspicious node to other nodes.

In conclusion, in this thesis, I demonstrated the SPNSA’s performance in multiple scenarios leading to an investigator identifying criminals’ connections within a large network. The probability of obtaining criminals and their connections using random feed is low when compared to the use of crime suspects. The SPNSA works well given a set of relevant algorithm feed prior to the relationship grouping. I showed that the criminals’ importance ranking that were previously dependent on centrality measures can now be improved by using source-intermediate reliance ranking. This new shortest paths network search algorithm (SPNSA) and the reliance measure can be applied to one-to-one, or one-to-many relationships, for example, hyperlinks that connect different web pages, client-server links or predator-prey relationships.
Appendix A

Tables

Tables A.1 - A.6 are results of 1-BCC shortest paths network analysis from section 3.3.4.

Table A.1: Path connecting Andrew Fastow to other nodes

<table>
<thead>
<tr>
<th>End Node</th>
<th>Length of shortest path</th>
<th>Intermediate Nodes in the path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Louise Kitchen</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Steven Kean</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Mark Taylor</td>
<td>2</td>
<td>Louise Kitchen</td>
</tr>
<tr>
<td>Vince Kaminski</td>
<td>2</td>
<td>Louise Kitchen</td>
</tr>
<tr>
<td>David Forster</td>
<td>2</td>
<td>Louise Kitchen</td>
</tr>
<tr>
<td>Mike Mcconnell</td>
<td>2</td>
<td>Louise Kitchen</td>
</tr>
<tr>
<td>Sherri Sera</td>
<td>2</td>
<td>Steven Kean</td>
</tr>
<tr>
<td>Kevin Hannon</td>
<td>2</td>
<td>Steven Kean</td>
</tr>
<tr>
<td>Sara Shackleton</td>
<td>3</td>
<td>Louise Kitchen, Mark Taylor</td>
</tr>
<tr>
<td>James Derrick</td>
<td>3</td>
<td>Steven Kean, Sherri Sera</td>
</tr>
<tr>
<td>Jeff Skilling</td>
<td>3</td>
<td>Steven Kean, Kevin Hannon</td>
</tr>
<tr>
<td>Mark Mcconnell</td>
<td>3</td>
<td>Louise Kitchen, Mike Mcconnell</td>
</tr>
<tr>
<td>Alhamd Alkhayat</td>
<td>3</td>
<td>Louise Kitchen, Mike Mcconnell</td>
</tr>
<tr>
<td>Kate Cole</td>
<td>4</td>
<td>Steven Kean, Sherri Sera, James Derrick</td>
</tr>
<tr>
<td>Jan Moore</td>
<td>4</td>
<td>Louise Kitchen, Mike Mcconnell, Mark Mcconnell</td>
</tr>
<tr>
<td>Rod Hayslett</td>
<td>5</td>
<td>Louise Kitchen, Mike Mcconnell, Mark Mcconnell, Jan Moore</td>
</tr>
<tr>
<td>Alhamd Alkhayat</td>
<td>5</td>
<td>Steven Kean, Sherri Sera, James Derrick, Kate Cole</td>
</tr>
<tr>
<td>Tim Despain</td>
<td>6</td>
<td>Louise Kitchen, Mike Mcconnell, Mark Mcconnell, Jan Moore, Rod</td>
</tr>
</tbody>
</table>

Chapter 3 Table A.1 shows every path that begins with Andrew Fastow (686) and ends with the End Node. If the path length is greater than 1, then the nodes occurring in the path are also listed.
Table A.2: Path connecting Kevin Hannon to other nodes

<table>
<thead>
<tr>
<th>End Node</th>
<th>Length of shortest path</th>
<th>Intermediate nodes in the path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jeff Skilling</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Sherron Watkins</td>
<td>2</td>
<td>Jeff Skilling</td>
</tr>
<tr>
<td>James Derrick</td>
<td>3</td>
<td>Jeff Skilling, Sherri Sera</td>
</tr>
<tr>
<td>Vince Kaminski</td>
<td>4</td>
<td>Jeff Skilling, Sherri Sera, Steven Kean</td>
</tr>
<tr>
<td>Kate Cole</td>
<td>4</td>
<td>Jeff Skilling, Sherri Sera, James Derrick</td>
</tr>
<tr>
<td>Sara Shackleton</td>
<td>5</td>
<td>Jeff Skilling, Sherri Sera, James Derrick, Kate Cole</td>
</tr>
<tr>
<td><a href="mailto:t..hodge@enron.com">t..hodge@enron.com</a> (annonymous)</td>
<td>5</td>
<td>Jeff Skilling, Sherri Sera, James Derrick, Kate Cole, Sara Shackleton</td>
</tr>
<tr>
<td>Mark Taylor</td>
<td>6</td>
<td>Jeff Skilling, Sherri Sera, James Derrick, Kate Cole, Sara Shackleton, Mark Taylor</td>
</tr>
<tr>
<td>Rod Hayslett</td>
<td>6</td>
<td>Jeff Skilling, Sherri Sera, James Derrick, Kate Cole, <a href="mailto:t..hodge@enron.com">t..hodge@enron.com</a></td>
</tr>
<tr>
<td>Louise Kitchen</td>
<td>7</td>
<td>Jeff Skilling, Sherri Sera, James Derrick, Kate Cole, Sara Shackleton, Mark Taylor</td>
</tr>
<tr>
<td>Andrew Fastow</td>
<td>7</td>
<td>Jeff Skilling, Sherri Sera, James Derrick, <a href="mailto:t..hodge@enron.com">t..hodge@enron.com</a>, rod.hayslett, Tim Despain</td>
</tr>
<tr>
<td>David Forster</td>
<td>7</td>
<td>Jeff Skilling, Sherri Sera, James Derrick, Kate Cole, Sara Shackleton, Mark Taylor</td>
</tr>
<tr>
<td>Tim Despain</td>
<td>7</td>
<td>Jeff Skilling, Sherri Sera, James Derrick, Kate Cole, <a href="mailto:t..hodge@enron.com">t..hodge@enron.com</a>, Rod Hayslett</td>
</tr>
<tr>
<td>Mike Mcconnell</td>
<td>8</td>
<td>Jeff Skilling, Sherri Sera, James Derrick, Kate Cole, Sara Shackleton, Mark Taylor, Louise Kitchen</td>
</tr>
<tr>
<td>Mark Mcconnell</td>
<td>9</td>
<td>Jeff Skilling, Sherri Sera, James Derrick, Kate Cole, Sara Shackleton, Mark Taylor, Louise Kitchen, Mike Mcconnell</td>
</tr>
<tr>
<td>Alhamd Alkhayat</td>
<td>9</td>
<td>Jeff Skilling, Sherri Sera, James Derrick, Kate Cole, Sara Shackleton, Mark Taylor, Louise Kitchen, Mike Mcconnell</td>
</tr>
</tbody>
</table>

Continued on next page
Table A.2 – continued from previous page

<table>
<thead>
<tr>
<th>End Node</th>
<th>Length of shortest path</th>
<th>Intermediate nodes in the path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan Moore</td>
<td>10</td>
<td>Jeff Skilling, Sherri Sera, James Derrick, Kate Cole, Sara Shackleton, Mark Taylor, Louise Kitchen, Mike McConnell, Mark McConnell</td>
</tr>
</tbody>
</table>

Chapter 3 Table A.2 shows every path that begins with Kevin Hannon (10068) and ends with the End Node. If the path length is greater than 1, then the nodes occurring in the path are also listed.

Table A.3: Path connecting Lea Fastow to other nodes

<table>
<thead>
<tr>
<th>End Node</th>
<th>Length of shortest path</th>
<th>Intermediate nodes in the path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike McConnell</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>David Forster</td>
<td>2</td>
<td>Mike McConnell</td>
</tr>
<tr>
<td>Mark McConnell</td>
<td>2</td>
<td>Mike McConnell</td>
</tr>
<tr>
<td>Alhamd Alkhayat</td>
<td>2</td>
<td>Mike McConnell</td>
</tr>
<tr>
<td>Mark Taylor</td>
<td>3</td>
<td>Mike McConnell, David Forster</td>
</tr>
<tr>
<td>Louise Kitchen</td>
<td>3</td>
<td>Mike McConnell, David Forster</td>
</tr>
<tr>
<td>Steven Kean</td>
<td>3</td>
<td>Mike McConnell, Alhamd Alkhayat</td>
</tr>
<tr>
<td>Jan Moore</td>
<td>3</td>
<td>Mike McConnell, Mark McConnell</td>
</tr>
<tr>
<td>Vince.Kaminski</td>
<td>4</td>
<td>Mike Mcconnell, David Forster, Louise Kitchen</td>
</tr>
<tr>
<td>Sara Shackleton</td>
<td>4</td>
<td>Mike Mcconnell, David Forster, Mark Taylor</td>
</tr>
<tr>
<td>Sherron Watkins</td>
<td>4</td>
<td>Mike Mcconnell, Alhamd Alkhayat, Steven Kean</td>
</tr>
<tr>
<td>Kevin Hannon</td>
<td>4</td>
<td>Mike Mcconnell, Alhamd Alkhayat, Steven Kean</td>
</tr>
<tr>
<td>rod.hayslett</td>
<td>4</td>
<td>Mike Mcconnell, Mark McConnell, Jan Moore</td>
</tr>
<tr>
<td>James Derrick</td>
<td>5</td>
<td>Mike Mcconnell, Alhamd Alkhayat, Steven Kean, Sherri Sera</td>
</tr>
<tr>
<td>Tim Despain</td>
<td>5</td>
<td>Mike Mcconnell, Mark McConnell, Jan Moore, Rod Hayslett</td>
</tr>
<tr>
<td>Jeff Skilling</td>
<td>5</td>
<td>Mike Mcconnell, Alhamd Alkhayat, Steven Kean, Kevin Hannon</td>
</tr>
</tbody>
</table>

Continued on next page
Chapter 3. Table A.3 shows every path that begins with Lea Fastow (11010) and ends with the End Node. If the path length is greater than 1, then the nodes occurring in the path are also listed.

<table>
<thead>
<tr>
<th>End Node</th>
<th>Length of shortest path</th>
<th>Intermediate nodes in the path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrew Fastow</td>
<td>6</td>
<td>Mike Mcconnell, Mark Mcconnell, Jan Moore, Rod Hayslett, Tim Despain</td>
</tr>
<tr>
<td>Kate Cole</td>
<td>7</td>
<td>Mike Mcconnell, Alhamd Alkhayat, Steven Kean, Jeff Skilling, Sherri Sera, James Derrick</td>
</tr>
<tr>
<td>t..hodge (annonymous)</td>
<td>7</td>
<td>Mike Mcconnell, Alhamd Alkhayat, Steven Kean, Sherri Sera, James Derrick, Kate Cole</td>
</tr>
</tbody>
</table>

Table A.4: Frequency of node connecting Andrew Fastow to other nodes

<table>
<thead>
<tr>
<th>Node</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Louise Kitchen (11370)</td>
<td>10</td>
</tr>
<tr>
<td>Steven Kean (17597)</td>
<td>6</td>
</tr>
<tr>
<td>Mike Mcconnell (12935)</td>
<td>5</td>
</tr>
<tr>
<td>Sherri Sera (16926)</td>
<td>3</td>
</tr>
<tr>
<td>Mark Mcconnell (11895)</td>
<td>3</td>
</tr>
<tr>
<td>Mark Taylor (11939)</td>
<td>2</td>
</tr>
<tr>
<td>James Derrick (7513)</td>
<td>2</td>
</tr>
<tr>
<td>Jan Moore (7609)</td>
<td>2</td>
</tr>
<tr>
<td>Kevin Hannon (10068)</td>
<td>1</td>
</tr>
<tr>
<td>Rod Hayslett (15805)</td>
<td>1</td>
</tr>
<tr>
<td>Kate Cole (9643)</td>
<td>1</td>
</tr>
</tbody>
</table>

Chapter 3. Table A.4 shows the frequency of node connecting Andrew Fastow to other nodes.

Table A.5: Frequency of node connecting Kevin Hannon to other nodes

<table>
<thead>
<tr>
<th>Node</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jeff Skilling (8024)</td>
<td>16</td>
</tr>
<tr>
<td>Sherri Sera (16926)</td>
<td>15</td>
</tr>
<tr>
<td>James Derrick (7513)</td>
<td>13</td>
</tr>
</tbody>
</table>

Continued on next page
Table A.5: Frequency of node connecting Kevin Hannon to other nodes

<table>
<thead>
<tr>
<th>Node</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kate Cole (9643)</td>
<td>11</td>
</tr>
<tr>
<td>Sara Shackleton (16383)</td>
<td>7</td>
</tr>
<tr>
<td>Mark Taylor (11939)</td>
<td>6</td>
</tr>
<tr>
<td>Louise Kitchen (11370)</td>
<td>4</td>
</tr>
<tr>
<td>Mike McConnell (12935)</td>
<td>3</td>
</tr>
<tr>
<td><a href="mailto:t..hodge@enron.com">t..hodge@enron.com</a> (17913)</td>
<td>3</td>
</tr>
<tr>
<td>Rod Hayslett (15805)</td>
<td>2</td>
</tr>
<tr>
<td>Steven Kean (17597)</td>
<td>1</td>
</tr>
<tr>
<td>Tim Despain (18360)</td>
<td>1</td>
</tr>
<tr>
<td>Mark McConnell (11895)</td>
<td>1</td>
</tr>
</tbody>
</table>

Chapter 3: Table A.5 shows the frequency of node connecting Kevin Hannon to other nodes.

Table A.6: Frequency of node connecting Lea Fastow to other nodes

<table>
<thead>
<tr>
<th>Node</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike McConnell (12935)</td>
<td>18</td>
</tr>
<tr>
<td>Alhamd Alkhayat (441)</td>
<td>7</td>
</tr>
<tr>
<td>Steven Kean (17597)</td>
<td>6</td>
</tr>
<tr>
<td>Mark McConnell (11895)</td>
<td>4</td>
</tr>
<tr>
<td>David Forster (3964)</td>
<td>4</td>
</tr>
<tr>
<td>Jan Moore (7609)</td>
<td>3</td>
</tr>
<tr>
<td>Sherri Sera (16926)</td>
<td>2</td>
</tr>
<tr>
<td>James Derrick (7513)</td>
<td>2</td>
</tr>
<tr>
<td>Rod Hayslett (15805)</td>
<td>2</td>
</tr>
<tr>
<td>Mark Taylor (11939)</td>
<td>1</td>
</tr>
<tr>
<td>Louise Kitchen (11370)</td>
<td>1</td>
</tr>
<tr>
<td>Jeff Skilling (8024)</td>
<td>1</td>
</tr>
<tr>
<td>Kevin Hannon (10068)</td>
<td>1</td>
</tr>
<tr>
<td>Sherron Watkins(16929)</td>
<td>1</td>
</tr>
<tr>
<td>Kate Cole (9643)</td>
<td>1</td>
</tr>
<tr>
<td>Tim Despain (18360)</td>
<td>1</td>
</tr>
</tbody>
</table>

Chapter 3: Table A.6 shows the frequency of node connecting Lea Fastow to other nodes.
Tables A.7 and A.8 are results of 2-BCC shortest paths network analysis from section ??.

Table A.7: Path connecting Kevin Hannon to other nodes

<table>
<thead>
<tr>
<th>End Node</th>
<th>Length of shortest path</th>
<th>Intermediate nodes in the path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sherri Sera</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Greg Piper</td>
<td>2</td>
<td>Sherri Sera</td>
</tr>
<tr>
<td>Vince Kaminski</td>
<td>2</td>
<td>Sherri Sera</td>
</tr>
<tr>
<td>Rosalee Fleming</td>
<td>2</td>
<td>Sherri Sera</td>
</tr>
<tr>
<td>Andrew Fastow (686)</td>
<td>3</td>
<td>Sherri Sera, Rosalee Fleming</td>
</tr>
<tr>
<td>Louise Kitchen</td>
<td>3</td>
<td>Sherri Sera, Greg Piper</td>
</tr>
<tr>
<td>Travis McCullough</td>
<td>3</td>
<td>Sherri Sera, Greg Piper</td>
</tr>
<tr>
<td>Mark Taylor</td>
<td>3</td>
<td>Sherri Sera, Greg Piper</td>
</tr>
<tr>
<td>Rick Buy</td>
<td>4</td>
<td>Sherri Sera, Greg Piper, Louise Kitchen</td>
</tr>
<tr>
<td>Michelle Cash</td>
<td>4</td>
<td>Sherri Sera, Greg Piper, Louise Kitchen</td>
</tr>
<tr>
<td>Rex Shelby</td>
<td>4</td>
<td>Sherri Sera, Greg Piper, Travis McCullough</td>
</tr>
<tr>
<td>Sara Shackleton</td>
<td>4</td>
<td>Sherri Sera, Greg Piper, Mark Taylor</td>
</tr>
<tr>
<td>Ben Glisan</td>
<td>5</td>
<td>Sherri Sera, Greg Piper, Louise Kitchen, Rick Buy</td>
</tr>
<tr>
<td>Twanda Sweet</td>
<td>5</td>
<td>Sherri Sera, Greg Piper, Louise Kitchen, Michelle Cash</td>
</tr>
<tr>
<td>Kenneth Rice</td>
<td>6</td>
<td>Sherri Sera, Greg Piper, Louise Kitchen, Michelle Cash, Twanda Sweet</td>
</tr>
</tbody>
</table>

Chapter 3 Table A.7 shows every path that begins with Kevin Hannon (10068) and ends with the End Node. If the path length is greater than 1, then the nodes occurring in the path are also listed.

Table A.8: Frequency of node connecting Kevin Hannon to other nodes

<table>
<thead>
<tr>
<th>Node</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sherri Sera (16926)</td>
<td>14</td>
</tr>
<tr>
<td>Greg Piper (6667)</td>
<td>10</td>
</tr>
<tr>
<td>Louise Kitchen (11370)</td>
<td>5</td>
</tr>
<tr>
<td>Michelle Cash (12848)</td>
<td>2</td>
</tr>
<tr>
<td>Mark Taylor (11939)</td>
<td>1</td>
</tr>
<tr>
<td>Rosalee Fleming (15932)</td>
<td>1</td>
</tr>
</tbody>
</table>

Chapter 3 Table A.8 shows part of the frequency of node connecting Kevin Hannon to other nodes.
Bibliography


[70] M. Goldberg, M. Hayvanovych, A. Hoonlor, S. Kelley, M. Magdon-Ismail, K. Mertsalov, 
B. Szymanski, and W. Wallace, “Discovery, analysis and monitoring of hidden social networks 


Co., 1969.

[73] T. H. Haveliwala, “Topic-sensitive pagerank,” in Proceedings of the 11th international con- 

[74] H. Henseler, “Network-based filtering for large email collections in e-discovery,” Artificial 
Intelligence and Law, vol. 18, no. 4, pp. 413–430, 2010.


2009.

[77] P. Holme, M. Huss, and H. Jeong, “Subnetwork hierarchies of biochemical pathways,” Bioin- 

[78] F. Iqbal, B. C. Fung, and M. Debbabi, “Mining criminal networks from chat log,” in Web In-
telligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International 


www.washingtonpost.com/wp-dyn/content/discussion/2006/10/20/ID2006102000645.html 
(accessed 5th September 2013).


