Characterising Network Paths Through Their Role in Induced Substructures

A thesis submitted in partial fulfilment

of the requirement for the degree of Doctor of Philosophy

Lauren Hudson

September 2021

Cardiff University
School of Computer Science & Informatics

For Nannan

Abstract

Paths are vital in facilitating network connectivity and have been traditionally characterised by global graph theoretic measures. However, motivated by large or dynamic complex networks, alternative analysis methods have been become popular, based on assessing the presence of induced substructures. These typically involve profiling networks based on the under of over representation of particular induced triads. We examine in detail how induced triads support paths and network connectivity. We begin by considering a triadic census derived from all possible shortest paths as compared to a triadic census from the full network. We find distinct differences, and present a classification for induced triads based on the extent to which their edges can be used in a shortest path. This leads to a new binary classification for edges, called overt or covert, based on supporting flooding across induced triads. We develop these concepts to create local centrality measures that are computationally efficient and which can be used to express the potential for containment or spread from a path. We extend these measures to introduce a convenient edge criticality measure, and compare it against conventional criticality metrics. Results are demonstrated through networks from the literature and synthesised networks.

Acknowledgements

Thank you to Roger, Stuart, Liam and Walter for your continued support throughout this project.

Contents

Abstract **ii**

Acknowledgements **iii**

Contents **iv**

List of Figures **viii**

List of Tables **xvii**

List of Algorithms **xix**

1 Introduction **1**

1.1 Background 2

1.2 Hypothesis and Research Questions 8

1.3 Thesis Structure and Contributions 9

2 Background and Literature Review **13**

2.1 Introduction 13

2.2 Defining Edges in a Triad 14

2.3 Centrality Metrics for Complex Networks 22

2.3.1 Edge Based Centrality Metrics 23

2.3.2 Substructure Based Centrality Metrics 26

2.4 Criticality Metrics for Networks 30

2.5 Data sets 37

2.6 Conclusions 40

3 Understanding graph Connectivity through Paths and Induced Triads 42

3.1 Overview 43

3.2 Methodology for Path-based Census 47

3.2.1 Triadic Census on Graphs 47

3.2.2 Triadic Census on Paths 54

3.3 Results 62

3.4 Summary 71

4 The Centrality of Edges Based on their Role in Induced Triads 73

4.1 Overview 74

4.1.1 Edge Centrality and Graphlets 74

4.2 Defining the Role of Edges 76

4.3 Relationship with Transitivity 79

4.4 Profiling Data Sets using Overt and Covert Centrality 81

4.4.1 Computing Overt and Covert Centrality 83

4.4.2 Results 86

4.5 Comparing Overt and Covert Centrality with Existing Metrics 129

4.6 Summary 136

5 The Role of Overt and Covert Edge Centrality in Paths 139

5.1 Overview 140

5.2 Methodology 143

5.2.1 Overt and Covert Path Weights 144

5.2.2 Choosing Minimal Paths 145

5.2.3 Overt-Length and Covert-Length Trade-off 148

5.2.4 Length-Overt Trade-off 151

5.2.5 Summary of Notation 154

5.3 Results 154

5.3.1 Observations on Increasing Edge Density 156

5.3.2 ER networks 160

5.3.3 Hub and Spoke format networks 174

5.4 Summary 177

6 Assessing Edge Criticality through Overt and Covert Centrality 180

6.1 Overview 181

6.2 Considering Overt and Covert Centrality Simultaneously 185

6.3 Methodology 190

6.4 Results 193

6.5 Summary 211

7 Final Conclusions and Future Work **213**

7.1 Research Questions and Contributions 213

7.2 Future Research 219

7.3 Final Remarks 221

Bibliography **223**

List of Figures

1.1 The 13 isomorphism classes of connected triad [75], using Holland and Leindhardt's MAN labelling [39]. The first unit represents the number of mutual (reciprocated) ties, the second unit the number of asymmetric ties and the third unit the number of null ties in the triad. Sometimes a fourth character is present to distinguish between triads that would otherwise have the same labelling, and represents additional characteristics of the triad. T is used in transitive triads (see Chapter Two [75]), C represents a cycle formed by the edges present in the triad. D represents down, U represents up. For example, `030C` has zero mutual ties, three asymmetric ties, zero null ties and is cyclic . . 5

1.2 Two triads `021D` and `021U` (see Figure 1.1). The role of the central node j differs greatly in between both these triads 6

1.3 Different roles edges can play in connected triads. In Triad One, the edge (i, j) can support flooding the triad, but it does not in Triad Two. In Chapter Four we define edges as overt or covert, where in this example (i, j) acts as overt in Triad One and covert in Triad Two. Were this to represent communication on edge (i, j), then when (i, j) is overt (Triad One) 8

2.1 Example of two triplets, where T_1 is an open triplet and T_2 a closed triplet 15

2.2 Example for computing the local clustering coefficient C_v of vertex v. 17

2.3 Example of three triads T_1, T_2 and T_3 where T_1 is transitive, T_2 is intransitve and T_3 is vacuously transitive. 19

2.4 Example of transitive triad T, where edges (i, j) and (i, k) play different roles in terms of flooding the triad 21

2.5 Automorphism Orbits of all possible undirected graphlets of up to five vertices. Each automorphism orbit class is labelled by a new number, whilst a new isomorphism class of graphlet is labelled by G_k 29

3.1 Graph $G = (V(G), E(G))$ where $V(G) = \{x, u, v, w\}$ and $E(G) = \{(x, u), (u, v), (v, w)\}$ 48

3.2 Graph $G = (V(G), E(G))$ where $V(G) = \{u, v, u_1, u_2, u_3\}$ and $E(G) = \{(u, v), (u_1, u), (u_2, u), (v, u_3)\}$ 53

3.3 Graph G, with the `030T` triad t highlighted in red. 57

3.4 Graph G', a modification to G through adding the blue edge (u_1, u_3). . 57

3.5 Induced subgraph $G'[p']$ of shortest path p' on G'. 58

3.6 Graph $G = (V(G), E(G))$ on vertices $V(G) = \{u_1, u_2, u_3, u_4, u_5\}$ and edges $E(G) = \{(u_1, u_2), (u_2, u_3), (u_3, u_4), (u_4, u_5)\}$ 60

3.7 Census Measures of Electrical circuit networks. 63

3.8 Census Measures of Food web networks. 64

3.9 Census Measures of Organise networks. 65

3.10 Census Measures of Regulatory networks. 65

3.11 Census Measures of miscellaneous networks. 66

3.12 `030T`, `120D`, `120U` and `300`. The red edges represent a 2-path, with the corresponding shorcut in blue 66

3.13 A heirarchy graph G, with examples of `021C`, `021U` and `021D` highlighted in blue, red and green respectively 70

3.14 The addition of the edges (u_2, u_3) and (u_3, u_2) to G from Figure 3.13 . 70

4.1 Different roles edges can play in connected triads. In Triad One, the edge (i, j) can support flooding the triad, but it does not in Triad Two 76

4.2 Triad t where (i, j) is overt and (j, k) covert. If t represents communication, then (i, j) enables message spread within the triad, whilst (j, k) does not. 77

4.3 All connected triad types with overt and covert edges indicated by O and C respectively 78

4.4 Example of three triads T_1, T_2 and T_3 where T_1 is transitive, T_2 is intransitve and T_3 is vacuously transitive 79

4.5 Transitive triad t, where t necessarily contains the overt edge (i, j). . . 80

4.6 Triad t where the edge (i, j) is overt yet t is an intransitive triad. . . . 81

4.7 The simple graph G containing overlapping triads in which the same edge acts as both overt and covert 82

4.8 93

4.9 US Airports [42] 95

4.10 Open Flights [67] 96

4.11 s420 [5] 97

4.12 s838 [5] 98

4.13 Mangwet [74] 99

4.14 Baywet [74] 100

4.15 Little Rock Lake [42] 101

4.16 Ythan [1] 102

4.17 St. Marks Seagrass [14] 103

4.18 Grassland [14] 104

4.19 p2p-gnutella04 [47] 105

4.20 p2p-gnutella05 [47] 106

4.21 p2p-gnutella06 [47] 107

4.22 p2p-gnutella08 [47] 108

4.23 p2p-gnutella09 [47] 109

4.24 C. Elegans [40] 110

4.25 Drosophila Medilla 1 [58] 111

4.26 Mouse Visual Cortex 2 [58] 112

4.27 Mouse Retina 1 [58] 113

4.28 Rattus Norvegicus [58] 114

4.29 Cross Parker Consulting [63] 115

4.30 Freemans EIES n48 1 [63] 116

4.31 Freemans EIES n48 2 [63] 117

4.32 Cross Parker Manufacturing [63] 118

4.33 Eva [41] 119

4.34 Bitcoin Alpha [47] 120

4.35 Bitcoin OTC [47] 121

4.36 Email EU Core [47] 122

4.37 Prison Inmate [5] 123

4.38 UCIrvine [63] 124

4.39 WikiVote [47] 125

4.40 E. coli transcription [5] 126

4.41 Yeast transcription [5] 127

4.42 Political Blogs[2] 128

4.43 The correlation matrix comparing the spearman correlation of each metric across all 34 data sets 130

4.44 The correlation matrix comparing the spearman correlation of each metric across all 34 data sets, with insignificant correlation results indicated by green cells (p >0.05) 131

4.45 Scatter plots showing the relationship between number of nodes in a network and mean covert and overt edge centrality 133

4.46 Scatter plots showing the relationship between density of a network and mean covert and overt edge centrality 133

4.47 Scatter plots showing the relationship between clustering of a network and mean covert and overt edge centrality 134

4.48 Scatter plots showing the relationship between reciprocity in a network and mean covert and overt edge centrality 134

4.49 Scatter plots showing the relationship between reciprocity in a network and mean covert and overt edge centrality 135

4.50 Scatter plots showing the relationship between average degree centrality and mean covert and overt edge centrality 135

4.51 Scatter plots showing the relationship between average reaching centrality and mean covert and overt edge centrality 136

4.52 Scatter plots showing the relationship between average diameter and mean covert and overt edge centrality 136

5.1 The graph G, with paths $p_1 = (u_1, u_2, u_3, u_4, u_5)$ and $p_2 = (u_1, u_6, u_7, u_5)$. The potential spread of a message sent down p_1 is highlighted in blue, whereas the potential spread of a message sent down p_2 is highlighted in red 142

5.2 Path $p = (u_1, u, v, u_2)$. 157

5.3 Path p with the addition of edge (v, u_3) in red. 158

5.4 Path p with the addition of (u_4, u), (u, u_5) or (u_6, v). 158

5.5 Path p with the addition of (u_4, u) and (u, u_4) in blue. 159

5.6 The average values (in a clockwise order) of $I_s^o(G)$, $I_s^c(G)$, $I_c^s(G)$ and $I_o^s(G)$, sampling over 50 instances of an ER-network 161

5.7 Example path $p = (u_1, u_2, u_3, u_4, u_5, u_6)$ where $w_c(p) = 0$. 165

5.8 Percentage of all least overt paths with non-zero t_s^o and t_o^S that occur within one of 50 simulations at each vertex-density pair 167

5.9 Frequency histograms plotting occurrence of t_s^o values, separated by number of vertices in the network 168

5.10 Frequency histograms plotting occurrence of t_o^s values, separated by number of vertices in the network 169

5.11 The average values of $I_s^o(G), I_s^c(G), I_c^s(G)$ and $I_o^s(G)$ (in clockwise order) against the corresponding length/weight of the shortest path, sampling over 50 instances of an ER-network 170

5.12 Percentage of all least covert paths with non-zero t_s^c and t_c^S that occur within one of 50 simulations at each vertex-density pair 171

5.13 Frequency histograms plotting occurrence of t_s^c values, separated by number of vertices in the network 172

5.14 Frequency histograms plotting occurrence of t_c^s values, separated by number of vertices in the network 173

5.15 The average values of $I_s^o(G)$, $I_s^c(G)$, $I_c^s(G)$ and $I_o^s(G)$ (clockwise order), sampling over 100 instances of the random k network construction . . 175

5.16 The average values of $I_s^o(G)$, $I_s^c(G)$, $I_c^s(G)$ and $I_o^s(G)$ (clockwise order), sampling over 100 instances of a scale-free network construction . . . 176

6.1 Graph G, where the removal of any black edge will not disconnect the graph, but the removal of the red edge (u, v) will disconnect the graph 182

6.2 G', the resulting graph by removing (u, v) from G in Figure 6.1. G' has seperated into two connected components 182

6.3 KDE Contour plots to show the probability of paired overt-covert weight edges 188

6.4 KDE Contour plots to show the probability of paired overt-covert weight edges 189

6.5 KDE Contour plots to show the probability of paired overt-covert weight edges 190

6.6 Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in all Airport data sets 198

6.7 Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (left) against the proportion of edges removed, p, in all Electrical circuit data sets 199

6.8 Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in all Food web data sets 200

6.9 Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (left) against the proportion of edges removed, p, in all Food web data sets 201

6.10 Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in all Internet data sets 202

6.11 Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in all Internet data sets 203

6.12 Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in all Neural data sets 204

6.13 Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in all Neural data sets 205

6.14 Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in all Organise data sets 206

6.15 Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in all Organise data sets 207

6.16 Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in all Social data sets 208

6.17 Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in all Social data sets 209

6.18 Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in all Regulatory data sets 210

6.19 Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in WWW data set 211

7.1 030T 220

7.2 Tetrad T 221

List of Tables

2.1 Recap of key themes in Section 2.2 34

2.2 Recap of key themes in section 2.3 35

2.3 Recap of key themes in Section 2.4 36

2.4 Networks and metrics used to describe them. $|V|$ denotes the number of vertices in the network, $|E|$ the number of edges. $Clustering$ represents the mean global clustering coefficient. $Reciprocity$ is a measure of global reciprocity. RC, DC and BC denote the global reaching centrality, mean degree centrality and mean betweenness centrality respectively. $Diameter$ is taken using the undirected version of the network. Self loops have been removed from the original data set, so we are testing on the largest connected component of the original data set. Decimal places are rounded to two significant figures 38

3.1 Triad codes and their corresponding triad type under function $TriType$ [8] 49

3.2 13 connected triad types categorised into Type I, Type II and Type III. 65

5.1 A summary of the notation used in Section 5.2 154

5.2 Triadic census over all generated networks, across specific densities. ER is sampled over $p = 0.1, 0.5$ and 0.9. Random k is sampled over $\alpha = 0.1, 0.5$ and 0.9. Scale free is sampled over $\beta = 0.91, 0.95$ and 0.99, with α and $\gamma = \frac{1-\beta}{2}$ in each case 163

5.3 Percentage of all least overt paths with non-zero t_s^o for each network size that have the corresponding t_s^o value, corresponding to Figure 5.9 168

5.4 Percentage of all least overt paths with non-zero t_o^s for each network size that have the corresponding t_o^s value, corresponding to Figure 5.10 169

5.5 Percentage of all least covert paths with non-zero t_s^c for each network size that have the corresponding t_s^c value, corresponding to Figure 5.13 173

5.6 Percentage of all least overt paths with non-zero t_C^s for each network size that have the corresponding t_C^s value, corresponding to Figure 5.14 173

List of Algorithms

1 CENSUS(G): Performs Triadic Census $T(G)$ [8] 50

2 EDGECENSUS(G, u, v): Performs Triadic Census $T_G(u, v)$ of an edge (u, v). 51

3 EDGETRIADICCENSUS(G): Computes the Edge Triadic Census $A(G)$ on graph G. 53

4 PATHINDUCEDTRIADICCENSUS(G): Computes the Path Induced Triadic Census $I(G)$ on a graph G. 56

5 TRIADICEDGEBETWEENNESSCENSUS(G) 60

6 OVERTCENTRALITY(G, u, v) 85

7 COVERTCENTRALITY(u, v) 85

8 NETWORKCENTRALITY(G) 86

9 OVERTPATHWEIGHT(p): Computes the total overt weight $w_o(p)$ of a path p. 144

10 COVERTPATHWEIGHT(p): Computes the total covert weight $w_c(p)$ of a path p. 145

11 OVERTLENGTHTRADEOFF(G, u, v): Computes $t_s^o(u, v)$. 149

12 AVERAGEIMPROVEMENTOVERTLENGTH(G): Computes the average improvement in edges $I_s^o(G)$. 151

13 LENGTHOVERTTRADEOFF(G, u, v): Computes $t_s^o(u, v)$. 153

14 EDGESORTOCC((G)): Sorts edges in descending order according to their OCC. 191

15 CRITICALITYOCC(G): Produces the susceptibility index and size of largest connected component as edges are removed from G in descending order. 193

Chapter 1

Introduction

This thesis proposes new measures and techniques that consider how the links present within induced triads relate to network connectivity, specifically concerning paths. Triads are a cornerstone of complex network analysis, regarded as a useful unit for analysis. However the composition of a triad imposes different connectivity roles on its constituent edges and considering the edges of induced triads in a network gives an alternative basis to assess paths, connectivity and edge criticality. We formalise this as a new concept and examine it in detail.

We discuss relevant background in complex networks (Section 1.1), focusing on the role of edges, and role of triads in complex networks. Figure 1.3 exemplifies the motivation for the different roles edges play in triads, which is the main issue that we consider as a metric to analyse paths and connectivity in networks. We then move to discussing the research questions addressed in this thesis (Section 1.2) before discussing the thesis structure and relevant contributions given within each chapter (Section 1.2).

There is now a wealth of data available that captures the online and real-world social relationships within different groups and organisations, offering the potential for us to gain a much deeper understanding of the roles of individual nodes within different community structures. Accordingly a number of data sets are used throughout the thesis, as outlined in Section 2.5.

1.1 Background

A complex network is a type of mathematical graph, composed of nodes (or vertices) representing objects or entities, with edges between vertices representing their relationship. The point of differentiation with classical graph theory occurs due to the scale and possible dynamism of complex networks, where for example nodes and edges may frequently change, appear or disappear. To incorporate these phenomena, new forms of analysis beyond static graph theory have been introduced, often involving techniques from theoretical physics.

Throughout this thesis we will refer to the mathematical object as a graph. We always assume a graph $G = (V(G), E(G))$ is built on a vertex set $V(G)$ and an edge set $E(G)$, where $(u, v) \in E(G)$ is an edge if $u, v \in V(G)$ are connected. When we assign what the nodes and edges represent (i.e in terms of data sets) we refer to a graph as a network. Graphs can be undirected or directed, depending on whether the edges have direction (i.e the vertex pair representing the edge is ordered). Throughout the rest of the thesis we will discuss directed graphs; thus for simplicity we will simply called these graphs, and specifically refer to graphs which are not directed as undirected graphs.

The importance of complex networks has grown with the progression of science and availability of data sources through which modelling can take place. Typically, complex networks are used in a real-world context to understand non-trivial features relevant to a particular scenario. Examples where the study of complex networks has been useful include the Internet, emails and social networking. For example, Mastrobuoni and Patacchini [51] use various network metrics to identify key figures within mafia networks. They use various metrics to identify different roles of players within the mafia, and are even able to compare these statistics with various factors such as age, in order to profile the typical characteristics of mob bosses. Due to the scale and complexity of complex networks we cannot often easily observe the whole network, and instead methodologies in this field rely on metrics to characterise and profile them in other ways.

Much of the development in studying complex networks has involved generalising the concept of an edge. For example, Przulj [66] generalises the concept of node degree (i.e how many edges are attached to a node) by generalising to the number of substructures attached to a node, and uses this to generalise the degree distribution of a network. Edges in a graph can be thought of as induced substructures of the most basic kind, each involving just a pair of nodes. They represent an atomic substructure on which further insights can be built.

Complex network theory has taken this a step further by looking at the induced substructures from larger subsets of nodes, known as *graphlets*. For example, graphlets involving three nodes have been commonly studied, representing *triads* of nodes and the relationship between them.

We formally define an induced triad in Definition 1:

Definition 1. *Let $G = (V(G), E(G))$ be a graph. Then let $V(T) \subset V(G)$ be a subset of three vertices in G . An induced triad, T, is itself a graph with vertex set $V(T)$ and edge set $E(T)$ such that $(u, v) \in E(T)$ if an only if $(u, v) \in E(G)$.*

Similarly to graphs, induced triads can be undirected or directed. For simplicity, unless explicitly stated as undirected, when we talk about a substructure (dyad, triad or tetrad) we are referring to the directed induced substructure throughout the rest of the thesis.

There are 16 possible isomorphism classes of triads: those which are connected (i.e contain at least two edges, such that every vertex in a triad is incident with at least one edge) and those which are unconnected (such that not every vertex in a triad is incident with one edge). The 13 possible connected triads are shown in Figure 1.1, each with their own unique label using Holland and Leindhardt's MAN labelling [39]. The first unit represents the number of mutual (reciprocated) ties, the second unit the number of asymmetric ties and the third unit the number of null ties in the triad. Sometimes a fourth character is present to distinguish between triads that would otherwise have the same labelling, and represents additional characteristics of the triad. T is used

in transitive triads (see Chapter Two [75]), C represents a cycle formed by the edges present in the triad. D represents down, U represents up. For example, `030C` has zero mutual ties, three asymmetric ties, zero null ties and is cyclic.

We are interested in triads because they are the smallest non-trivial substructure. *Dyads* (substructures on two vertices) are constructed on two vertices and the edges between them. Therefore, although dyads can be used to identify reciprocity between vertices in a network, their interpretation stops at identifying interactions between vertices within a group setting.

Triads, in contrast, can identify a variety of patterns such as such as social balance [34, 13]. That is, then tendency for the edge (i, k) to exist given the existence of (i, j) and (j, k) within a triad. Triads also allow more information to be encoded relevant to a scenario, such as pairs of nodes that may act together to provide influence on a third node, for example. In fact interest in triads naturally arise naturally in many different contexts, concerning issues such as representation of logical structure, redundancy or social concepts in behavioural networks. For example, consider Figure 1.2. This example contains two triads where the central node plays a very different role in both triads. In `021D` node j holds influence (or potential power) over the other nodes. However in `021U` no single node holds influence over the others, and here j has no influence.

Larger substructures, such as *tetrads*, may present local characteristics which cannot be present in triads due to their small size, such as *bridging* within the substructure [23] (in tetrads, bridging occurs when three vertices are connected, yet only one is adjacent to a fourth). However, increasing the number of vertices in a substructure exponentially increases the number of isomorphism classes for the substructure. Whilst there are just 16 isomorphically different triads, there are 218 different tetrads. This makes it extremely difficult to observe and compare the behaviour of all possible classes of tetrad in a graph: indeed, much of the literature relating to tetrads focuses on a specific subset of induced tetrad (for example, [52], [23]) or investigates only undirected tetrads (such

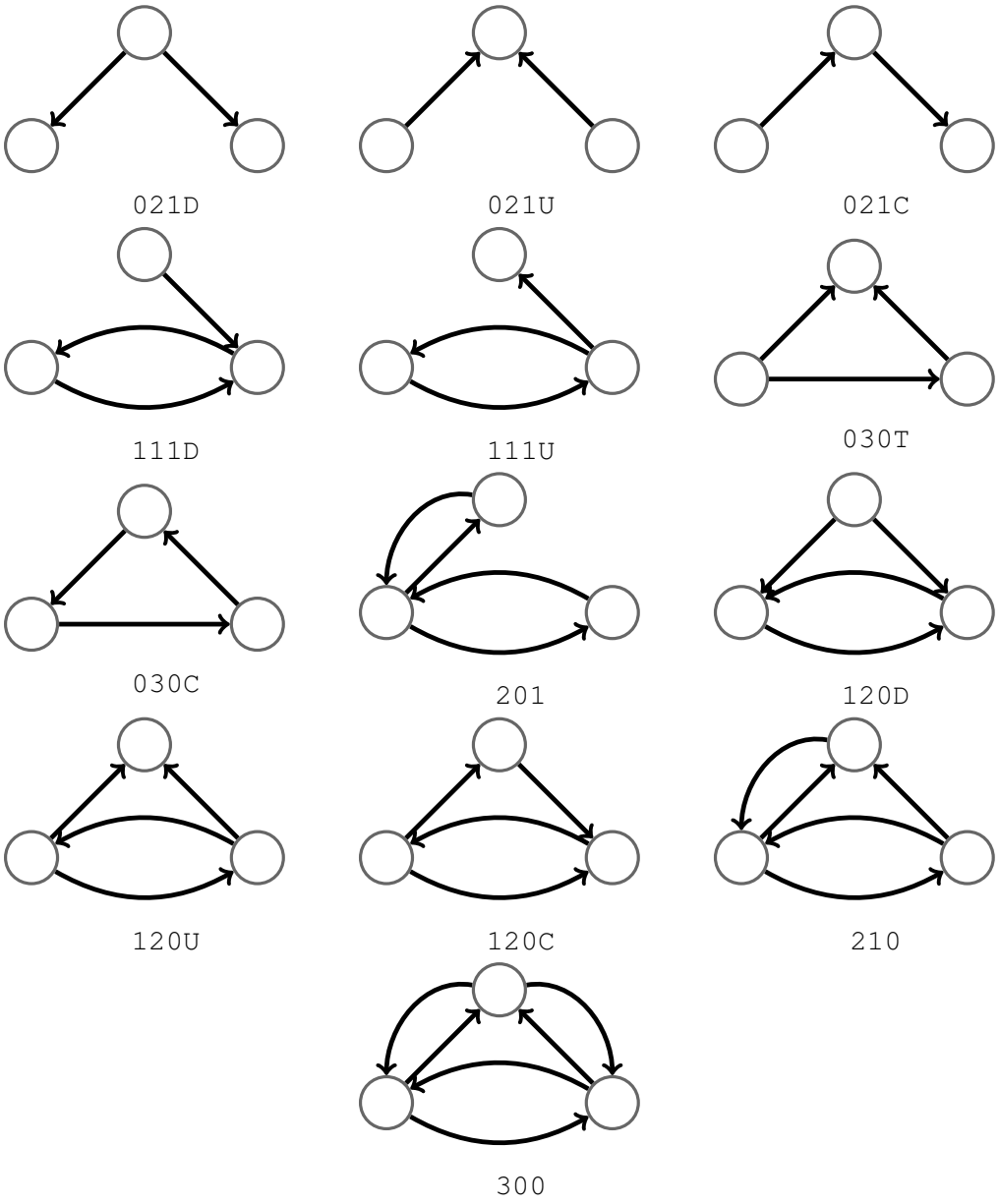


Figure 1.1: The 13 isomorphism classes of connected triad [75], using Holland and Leindhardt's MAN labelling [39]. The first unit represents the number of mutual (reciprocated) ties, the second unit the number of asymmetric ties and the third unit the number of null ties in the triad. Sometimes a fourth character is present to distinguish between triads that would otherwise have the same labelling, and represents additional characteristics of the triad. T is used in transitive triads (see Chapter Two [75]), C represents a cycle formed by the edges present in the triad. D represents down, U represents up. For example, `030C` has zero mutual ties, three asymmetric ties, zero null ties and is cyclic.

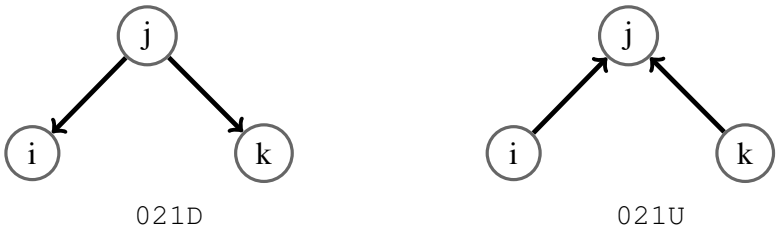


Figure 1.2: Two triads `021D` and `021U` (see Figure 1.1). The role of the central node j differs greatly in between both these triads.

as Milo et al. [56]). Further, finding and classifying all possible tetrads in a graph is more computationally complex than finding and classifying all possible triads because there are fewer nodes and edges in a triad.

Further understanding networks through triads motivates our research, given that triads are widely used as units of analysis for studying the entire network in many complex network scenarios. This includes the analysis of *motifs* [71], being induced triads that are significantly over represented in complex networks, which has gained to much popularity in recent years. Motifs were first introduced by Shen-Orr et al. [71] who found motifs present in networks across biochemistry, neurobiology, ecology, and engineering and noted that the motifs found in one classification of network may differ from those found in another. This approach takes networks, typically those of scale, and profiles then to determine the under or over representation of induced triads relative to the induced triads found from sampling a corpus of comparable randomly generated networks. The profile of motifs induced by a network provides a useful characterisation of its latent features, from which distinct classes of network have been established. This form of network analysis has mainly supported inter-network comparison based on assessing the relative volume of induced substructures. Famously Milo et al [56] study similarities in local structure of networks through profiling triadic motifs, outlining several superfamilies of otherwise unrelated networks.

However, as a consequence of this analysis approach, there are many papers in the literature where interpretation of results stops with identification of graphlets that represent motifs. While these may have domain-specific interpretations, it is also the

case that networks are often compared between different application areas. In these cases, it is rational to question in general terms, what the differences between induced substructures may mean in terms of general connectivity, for example what are the implications of a `111U` (see Figure 1.1) triad dominating in a complex network as compared to a `102U` triad?

These questions motivate further consideration of the role of different induced triads in network connectivity. We believe there is a need for a general method to interpret the implications of particular triads being present in a network structure, based on the configuration of their edges. Edge structure within triads affects the potential paths between nodes at a local level. It is possible that the simple nature of triads has led to the mis-conception that there is little to be gained from considering the role of edges within them. However, the position and direction of edges within the triad in relation to the triad's other edges is highly influential with respect to connectivity.

A useful example of our approach to classify edges within a triad can be seen in Figure 1.3. Consider Figure 1.3 in terms of communication along a path which includes the edge (i, j). Here the edge (i, j) holds different significance in Triad One than in Triad Two - specifically communication on edge (i, j) can subsequently be conveyed by j to k in Triad One but this is not the case in Triad Two. Thus if communication across (i, j) is sensitive, Triad One offers less potential for the message to spread than Triad Two. Conversely, if third party contagion is desirable, (i, j) in Triad One would be preferential. In Chapter Four, we will formally define the terms *overt* and *covert* to classify an edge based on its direction in relation to the direction and existence of its neighbours; which can be applied to describe whether an edge enables or hinders communication with a third party. Note also that because edges often participate in multiple induced subgraphs, it is possible that an edge may play different roles in different induced triads. Thus as networks increase in complexity, edges provide different degrees of freedom in supporting the nature of an intended communication in terms of potential spread or containment.

Figure 1.3: Different roles edges can play in connected triads. In Triad One, the edge (i, j) can support flooding the triad, but it does not in Triad Two. In Chapter Four we define edges as overt or covert, where in this example (i, j) acts as overt in Triad One and covert in Triad Two. Were this to represent communication on edge (i, j), then when (i, j) is overt (Triad One).

Throughout this thesis, we only discuss connected triads and so refer to these as triads from now on.

1.2 Hypothesis and Research Questions

We are interested in network connectivity: which is one of the most fundamental, basic concepts of graph theory. A graph is connected if every vertex is reachable from every other vertex via a path. Connectivity asks for the minimum number of elements that need to be removed to disconnect the graph, separating it into multiple subgraphs.

Triads are fundamental because they are the smallest, non-trivial substructure and act as the building blocks for graphs, yet existing literature is preoccupied with identifying motifs (triads which occur more frequently than at random). We propose to instead explore the link between triads and connectivity. In particular, we look to the role the edges (representing relationships between two vertices) of a triad play in graph connectivity.

We hypothesise that *classifying the role of edges in providing connectivity in networks with respect to induced triads, can provide additional insights to conventional graphlet-based analysis, that are beyond standard metrics of network connectivity.*

This hypothesis is intended to complement existing complex network analysis approaches where the unit of reference is the triad. From this perspective, it is natural to consider edges in a network taking into account the role that they play in triads. This leads us to develop new concepts that characterise edges relative to induced triads. It also provides a new bridge between complex-network analysis based on the presence of triads, and graph theoretic approaches to network analysis which address a network's structure through features aligned to edges and nodes.

The following specific research questions are addressed.

RQ1: The role of triads in paths: if connectivity in a graph is dependent on the existence of paths, then what is the relationship between triads and paths?

RQ2: The role of edges in triad connectivity: if connectivity within the triad is dependent on the arrangement of its edges, then how do we identify which edges are most fundamental to enabling connectivity within the triad?

RQ3: The role of edges of a triad in paths: combining RQ1 and RQ2, how can the edges which enable connectivity within the triad affect paths, and thereby connectivity within a graph?

These research questions are addressed using both theoretical and empirical observations and wide ranging data sources (Section 2.5).

1.3 Thesis Structure and Contributions

In this section we discuss the thesis structure, where each heading represents a chapter of the thesis. We highlight the relevant contributions in each chapter which investigate research questions RQ1-3.

Chapter Two - Background and Literature Review: The related literature and works can be found in Chapter Two. This is divided into three major sections. In

preparation for introducing overt and covert as classifications for edges in triads (Chapter Four), we highlight the relevant definitions of edges in triads such as local clustering coefficient and transitivity. Secondly, since we develop overt and covert classification for edges as a centrality metric in Chapter Four, (i.e. a measure of the importance of a node or edge within a network) to highlight edge importance, we introduce other key centrality metrics for complex networks. In particular, we highlight centrality metrics related to induced substructures and centrality metrics related to edges. We note that our centrality metric relates to both, highlighting its novelty in the literature. Finally, we discuss criticality metrics to determine importance of edges in terms of network structure.

Chapter Three - Alternative Census Measures for Triads Based on Their Occurrence Across Paths: Chapter Three serves as an introduction to the behaviour of triads across paths. Since paths are inherent to connectivity within a graph, and our new classification of edges introduced in Chapter Four can be applied to manage spread, we take a triad census of triads which occur along paths. We offer two new census measures to assess importance of triads to shortest paths, and demonstrate that they allow insight beyond the standard Graph Triadic Census measure. This leads to our first contribution:

C1 *New triad census measures to establish the proportion of triads that occur more frequently along shortest paths, as compared with the entire network. This addresses research question* **RQ1**.

Chapter Four - The Centrality of Edges Based on their Role in Induced Triads: In Chapter Four, we introduce the classification of edges as either overt or covert. This leads to our second contribution:

C2 *A new classification for edges based on their role in supporting connectivity within triads. This addresses research question* **RQ2**.

We note that an edge can be present in multiple triads and play different roles simultaneously. This leads to arguments establishing how many triads as edge acts as overt or covert. We present this as an entirely local new centrality measure for edges to determine their importance: satisfying the gap in the literature between centrality measures concerning substructures and centrality metrics concerning edges. We outline this as the third contribution:

C3 *A new local centrality metric for edges based on our new edge classification in C2. This addresses research question* **RQ2.**

We use our centrality metric to profile 34 real world networks by taking frequency distributions of the two centrality metrics. Further, we compare our new centrality metrics with some existing centrality metrics by taking the Spearman's rank correlation.

The Role of Overt and Covert Edge Centrality in Paths: As our centrality metrics highlight the importance of edges in terms of their ability to enable connectivity within the triad we apply our metrics in Chapter Five to paths to investigate the effect on graph connectivity through investigating the impact on paths. We control for overtness/covertness of paths in various generated networks, observing the difference in between these controlled paths and shortest paths. We offer this as an method to minimise the spread of information to unintended parties when sending a message between two individuals through a series of intermediaries, discussing our results in terms of cost/benefits to a system. This is our fourth contribution:

C4 *A new method to understand spread of a message through applying the new centrality metric C2 to path problems in networks. This addresses research question* **RQ3**.

Chapter Six - Edge Criticality in Networks Based on Overt and Covert Centrality: Chapter Six also explores the link between overt/covert edges and graph connectivity,

but instead of constructing paths, we investigate how removing edges breaks a graph into multiple connected components. We combine the overt and covert centrality into a single metric to determine an edge's criticality. Critical edges are important to network structure as the network breaks down when these edges are removed. We follow by comparing our overt and covert criticality metric to existing criticality metrics. Chapter Six offers contribution C5.

C5 *A new locally derived criticality metric for edges based on the centrality metric in C2 to observe the importance of an edge with respect to maintaining overall network connectivity. This addresses research question* **RQ3**.

These contributions have formed the following peer-reviewed paper:

> Hudson, L., Whitaker, R., Allen, S., Turner, L., & Felmlee, D. (2021). The Centrality of Edges Based on their Role in Induced Triads. ASONAM 2021 - The 2021 IEEE/ACM International Conference on Advances in Social Network Analysis and Mining (08-11 November 2021). The Hague, Netherlands.

Chapter 2

Background and Literature Review

2.1 Introduction

Complex networks are a fundamental area studied through mathematics, physics and computer science. These networks differ from other classes of networks such as random networks and lattices, exhibiting non-trivial characteristics such as long tailed degree distributions [6]. With the emergence of new technology that enables access to large data sets and the computational tools to analyse them, the study of complex networks has become more accessible. As such this area of research is interdisciplinary and therefore relates to a large body of research. We focus on the areas most relevant to our work. These concern: i) characterising edges in triads (Section 2.2), ii) centrality metrics for complex networks (Section 2.3), iii) criticality metrics for networks (Section 2.4) and iv) potential data sets (Section 2.5).

In Section 2.2 we address the classifications of triads based on the existence of particular edges, most notably the Global [49] and Local Clustering Coefficient [76]. Clustering is a feature dominant in complex networks, in particular social networks, where vertices form tightly knit groups [76, 37]. This is relevant to our development of a new classification for edges (Chapter Four), based on the role of edges within triads.

In Section 2.3 we discuss various centrality metrics. Centrality characterises edges or vertices as important (or central), differing based upon their interpretations of importance. We highlight centrality metrics built on edges, noting there are not many and

furthermore, most are generalisations of vertex-based variations [72]-[73] on betweenness centrality [26]. We also highlight centrality metrics built on substructures, such as [19, 66, 10], noting that these do not take into account the role of edges. In Chapter Four we utilise our classification for edges based on their role within induced triads as a new centrality metric, which bridges the gap in the literature between centrality metrics on edges and centrality metrics on substructures.

Section 2.4 relates to edge criticality [79]: an alternative to centrality in determining the importance of an edge to the overall structure of the network. This is fundamental to Chapter Six, where we construct a new criticality metric from our centrality metrics, and compare this with existing measures. We note that in social networks, counter intuitively weak ties are the most critical to network structure and their removal dissolves a network more quickly [61], through breaking it into a larger number of connected components. We explore some existing criticality metrics built on cliques (an easily computable metric), most notably the Bridgeness Index [77] and the Betweenness Centrality and Clique model [79] to establish those most critical edges.

Finally, in Section 2.5 we identify and display features of the 34 data sets to test our edge metrics on.

2.2 Defining Edges in a Triad

The role of vertices within triads has been extensively studied, most notably in terms of global [49] and local clustering coefficient [76]: measures to highlight the degree of ties between neighbours of neighbours of a vertex. In real world networks (particularly social networks) verticies tend to cluster together, forming tightly knit groups [76, 37]. We begin by highlighting these concepts and their role in supporting network connectivity. We note that these measures cannot be applied to weighted or directed networks; moving to developments taken to rectify this such as the weighted clustering coefficient [64] or transitivity [75] in weighted networks [64]. We discuss the limitations of these measures,

highlighting their dependence on vertices and lack of information relative to edges.

Global Clustering Coefficient

Measuring the global clustering coefficient was first attempted by Luce and Perry [49]. They turn to adjacency matrices of graphs to calculate the number of open and closed triplets (though they do not explicitly call them open and closed), summing the non-diagonal cells of the squared adjacency matrix to calculate the number of open triplets, and summing the diagonal cells of the cubed adjacency matrix to calculate the number of closed triplets [64]. A triplet is defined as three vertices that satisfy the property of having at least two undirected edges between them (in other words - a connected triad, although triplets are built on undirected networks in this context). A triplet is closed if there are three edges between the vertices, or open otherwise. For example, consider Example 2.2.1.

Example 2.2.1. *Suppose there exists two triples T_1 and T_2, both with vertex set: $\{i, j, k\}$ and edge sets $E(T_1) = \{(i, j), (j, k)\}$ and $E(T_2) = \{(i, j), (j, k), (i, k)\}$ as shown in Figure 2.1. Then T_1 is an open triplet, whereas T_2 is a closed triplet.*

Figure 2.1: Example of two triplets, where T_1 is an open triplet and T_2 a closed triplet.

Recalling Figure 1.3 from Chapter One, we discuss the different roles edges can play in connecting triads, in terms of ability to flood the triad (overt or covert, later formally defined in Chapter Four). This classification relies on the presence of a particular edge within a triad relative to the edge we are classifying, i.e: relying on the existence of a

neighbour of a neighbour. Open and closed triplets are a similar, primitive definition; also relying on the presence of a neighbour of a neighbour, but instead classifying the whole triad rather than the triad's edges. This definition is for undirected graphs, though can be extended to directed graphs, as seen with transitive triads.

The global clustering coefficient C of a graph G is the fraction of closed triplets which exist of all possible triples in the entire network, i.e:

$$C = \frac{\text{Number of Closed Triplets}}{\text{Number of All Triplets}} \tag{2.1}$$

C is bound between 0 and 1. In a random network, $C \longrightarrow 0$ as the graph increases in size; whereas in a completely connected network $C = 1$.

One major limitation of the standard global clustering coefficient [49] is that it can't be applied to weighted networks. Were the weights of edges to represent the strength of ties, then the global clustering coefficient would not recognise this and therefore may characterise similarly two networks with very different weight distributions [64]. Tore and Opsahl [64] offer a solution to this with their global clustering coefficient for weighted networks (see Section 2.2). Further, the global clustering coefficient cannot be applied to directed networks, where reciprocity of relationships is another important factor. Transitivity [75] as defined on directed networks extends this, and can be applied to networks in the same manner as global clustering coefficient (see Section 2.2).

Local Clustering Coefficient

In contrast to the global clustering coefficient, the local clustering coefficient (Watts and Strogatz) [76] measures the cliquishness in neighbourhoods. The definition for local clustering coefficient is still built on the existence of closed triplets, but allows to distinguish the degree of clustering pivoting on a single vertex.

Suppose that for some graph G with vertex set $V(G)$ and edge set $E(G)$, a vertex

$v \in V(G)$ has k_v neighbours; then at most $\frac{k_v(k_v-1)}{2}$ edges can exist between them (this occurs when every neighbour of v is connected to every other neighbour of v). Then the local clustering coefficient C_v on vertex v denotes the fraction of these allowable edges that exist, i.e:

$$C_v = \frac{|\{(i,j) \in E(G) | i,j \in N(v)\}|}{\frac{k_v(k_v-1)}{2}} = \frac{2|\{(i,j) \in E(G) | i,j \in N(v)\}|}{k_v(k_v-1)} \tag{2.2}$$

where $N(v)$ represents the neighbourhood of v.

For example, consider Example 2.2.2.

Example 2.2.2. *Suppose there exists some vertex $v \in V(G)$ with neighbourhood $N(v) = \{u_1, u_2, u_3\}$ where $n_i \in V(G)$. Suppose that $(u_1, u_2), (u_2, u_3) \in E(G)$ as shown in Figure 2.2.*

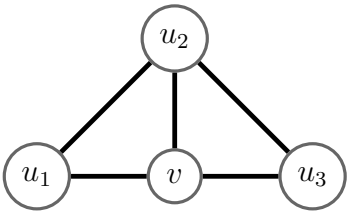


Figure 2.2: Example for computing the local clustering coefficient C_v of vertex v.

Then the maximum possible number of edges between neighbours of v is 3, i.e the edges $(u_1, u_2), (u_2, u_3), (u_1, u_3)$. Since only two of these exist, then the local clustering coefficient for v is $C_v = \frac{2}{3}$.

Local clustering coefficient [76] suffers from similar problems to the global clustering coefficient, namely that it cannot be applied to weighted or directed networks in its current formulation in Equation 2.2 [64]. There have been several attempts to generalise local clustering for weighted networks such as in [7, 48, 62, 80]. Further, local clustering coefficient is biased by a vertex's degree. The higher the degree of a vertex, the greater the number of neighbours and therefore the higher the likelihood it is contained in more

open triplets, decreasing the clustering coefficient [64] This correlation may be even stronger in preferential attachment networks [64]

Clustering Coefficient for Weighted Networks

A major limitation of the both clustering coefficients is that they cannot be applied to weighted networks [64]. Opsahl and Panzarasa [64] generalised the global clustering coefficient to account for weighted networks. To calculate a generalised global clustering coefficient, C_w, for a graph G they propose:

$$C_w = \frac{\text{Total Weight of Closed Triplets}}{\text{Total Weight of Triplets}} \tag{2.3}$$

The authors take care to ensure that under their measure, the properties $C_w \longrightarrow 0$ as network size increases and $C_w = 1$ in a completely connected network still hold. They extend their argument for directed networks under the guise of a generalised transitivity measure, which we outline in Section 2.2.

Transitive Triads

A further limitation of the global clustering coefficient is that it cannot be applied to directed networks. Transitivity instead defines a more refined concept for triplets, which can be applied to undirected networks and share results with global clustering coefficient [59, 21]. We note that transitivity is not always consistent in the literature: sometimes the global clustering coefficient is called the transitivity coefficient, which is counter-intuitive because (as we will see) Waserman and Faust's [75] definition of transitivity applies only to directed networks, whereas the global clustering coefficient is only applicable to networks which are undirected. However, were we to map the edges of a directed triad to undirected edges, Wasserman and Faust's [75] definition of a transitive triad would map to a closed triplet, whilst an intransitive or vacuously

transitive triad would map to an open triplet; hence the definition is equivalent under this mapping. For consistency, we will always refer to Wasserman and Faust's definition [75] of transitivity and refer to local [76] and global clustering [49] coefficients as the undirected version defined in Equations 2.1 and 2.2 in Sections 2.2 and 2.2 respectively.

Transitive triads are an example of structures that relate network theory to human behaviour and relationships between groups of three people. Balance theory is the study of attitude change, first introduced by Heider [34], which was extended to triadic analysis in social networks by Harary and Cartright [13]. In particular, triads in this context can highlight possible relations such as "A friend of a friend is also my friend ". These relations are naturally transitive, i.e if i ~ j and j ~ k then i ~ k where ~ represents the relationship of 'liking'. This extends to triads through Waserman and Fausts' definition of transitivity in triads [75].

Definition 2. *A triad (i, j, k) is transitive if whenever there is a directed edge from i to j and from j to k, then there is a directed edge from i to k [75]. If no edge (i, j) or (j, k) exists then a triad is vacuously transitive. Otherwise, it is intransitive.*

Consider Example 2.3.

Example 2.2.3. *Suppose there exists three triads T_1, T_2 and T_3 on vertex set: $\{i, j, k\}$ and edge sets $E(T_1) = \{(i, j), (j, k)(i, k)\}$, $E(T_2) = \{(i, j), (j, k)\}$ and $E(T_3) = \{(i, j), (i, k)\}$ as shown in Figure 2.3. Then T_1 is a transitive triad, whereas T_2 is intransitive. T_3 is vacuously transitive as there is an edge from i to j and i to k, yet no edge from j to k.*

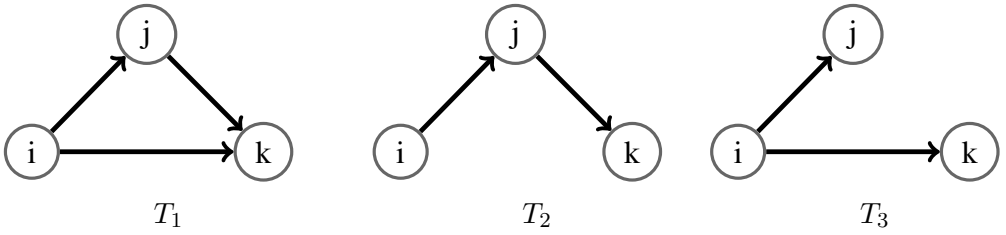


Figure 2.3: Example of three triads T_1, T_2 and T_3 where T_1 is transitive, T_2 is intransitve and T_3 is vacuously transitive..

Tore and Opsahl [64] define T_w in order to apply a global clustering coefficient to directed networks. They further their generalised clustering coefficient for weighted networks (Equation 2.3 to a generalised transitivity coefficient for weighted networks through defining T_w as:

$$T_w = \frac{\text{Total Weight of Transitive Triplets}}{\text{Total Weight of Triplets}} \tag{2.4}$$

When T and T_w are defined this way, they exhibit the same properties as global clustering coefficient and generalised global clustering coefficient for weighted networks, namely: $T \longrightarrow 0$ and $T_w \longrightarrow 0$ as the network size increases in random networks, and $T = 1$ and $T_w = 1$ in completely connected networks.

These measures are designed to employ triads to assess global characteristics of a network, but in doing so they focus on particular triads and the importance of particular edges is not captured.

Clustering focuses on the existence of a third edge when the two other edges are present between three vertices. All measures are either vertex based (i.e local clustering focuses on a single vertex) or focus on the whole triad (i.e the global clustering coefficient focuses on prevalence of triads in a network). They focus on existence of edges to determine if vertices are present in closed triads. They do not focus on the role of individual edges within these triads or differentiate between them. This lack of classification of edges makes it impossible to understand the local edge impact on spread throughout a network. Whilst global and local clustering discuss the cliquishness of groups within a network, they do little to control the spread of messages throughout the cliques. For example, the triad `030T` is transitive, yet the edge which enables the transitivity of the triad itself cannot flood the triad, as shown in Example 2.2.4. Here, triad T is transitive because there exists an edge from i to j. However, were we to traverse the edge from (i, j) then you could traverse no further edges. The same is not true for edge (i, k), where you are able to flood all vertices in the triad by traversing

this edge. In other words, not all edges in the triad are equal in this regard.

Example 2.2.4. *Consider the triad* $T = \texttt{030T}$ *on vertex set* $V(T) = \{i, j, k\}$ *and edge set* $E(T) = \{(i, j), (i, k), (j, k)\}$ *as in Figure 2.4.*

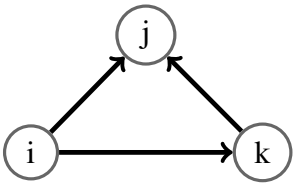


Figure 2.4: Example of transitive triad T, where edges (i, j) and (i, k) play different roles in terms of flooding the triad.

Structural balance theory was born from Heider's [34] study on the ability of friends of friends to become friends. But what if we wanted to ensure that a message was delivered to one friend without being delivered to the third friend in the closed triad? In this regard not all friends are equal, and early modelling fails to consider such differentiation.

In Example 2.2.4, we could say we wanted to send a message from i to j without the same message reaching k, this is entirely possible due to the nature of (i, j) in relation to other edges in the triad T. The same does not hold for getting a message from i to k. Alternatively, we could pose that i wants to pass a message to both friends in as little steps as possible. In this case, it is best to pass a message to k who will in turn deliver the message to j.

Noting this case, throughout the literature we see there is little discussion of the role of edges in triads with respect to supporting local connectivity. We turn our attention to centrality metrics, which have the ability to classify importance, and show there is little to bridge the gap between centrality measures on edges and centrality measures on induced substructures. However, this leaves a fundamental concept open for consideration that has possible application in controlling for spread on content or artifacts across networks.

2.3 Centrality Metrics for Complex Networks

Generally speaking, centrality metrics are able to characterise particular edges or vertices as important. They differ based upon alternative interpretations of importance.

Starting with the basic concepts, the *degree* of a vertex in a graph is the number of edges incident with that vertex. In directed graphs, there are two forms of degree: namely the in-degree (i.e the number of edges directed into the vertex) and the out degree (the number of edges directed out of the vertex). *Degree centrality* [27] is simply the degree of a vertex: those with a higher degree have more links and typically are seen as more important to a network. Many vertices with a high degree centrality also score highly in other centrality measures.

Degree centrality is useful for determining how many links a vertex has, but does not always tell us about the importance of a vertex to a group, as the edges may not be very important in terms of connectivity. An alternative measure to degree centrality that considers connectivity is *closeness centrality* [9]. Closeness centrality was first developed by Bavelas [9] who wanted to out the relative position of a vertex in a graph relative to other vertices. It is the average distance (in terms of length of a shortest path) of a vertex from every other vertex in the network. A similar measure is local reaching centrality [57]. This is the number of vertices in a network reachable via a path from a singular vertex. Both of these measures concern the connectivity: vertices which are more central are those which are more connected to other vertices in the network.

All of the aforementioned centrality metrics are vertex based. In the upcoming sections we discuss particular centrality metrics built on substructures and other centrality metrics that concern edges, noting there is little literature that addresses both. This supports our work in Chapter Four, where we introduce a centrality metric built on the definition of edges in substructures.

2.3.1 Edge Based Centrality Metrics

Vertex betweenness centrality was first proposed by Freeman [26] and later refined by Freeman et al. [28] by generalising the measure to network flows. The betweenness centrality of a vertex u is the number of shortest paths that pass through u. In this approach centrality is defined as having more control over the flow of information. In [28] Freeman references the following to highlight its application: *"that a person who is more close to other people will have several benefits over others, be it: greater information access higher status, power or influence."*

There are several limitations to betweenness centrality, namely that it exclusively focuses on shortest paths [28]. This may not necessarily be true for human communication, which may not rely on shortest paths to link a pair of people [28]. Further, the standard betweenness centrality measure takes into account shortest paths of any length; yet very long paths may not be a realistic representation of friendship [12]. The measure only works for connected graphs, as the distance between vertices which are not connected is undefined [12]. Borgatti [11] discusses this further, surmising that the way the flow of information as defined by Freeman [26] is unsuitable for the dissemination of many categories of spread: such as infection or gossip, which is often untargeted and unlikely to reach the target through a shortest possible path. There are many variations on the definition for betweenness centrality to address these issues, particularly other alternative approaches considering the role of edges including various forms of network flow (e.g., [28, 11, 12]). However, many of these are vertex-based measures, and we will instead focus on edges.

Girvan and Newman [29] generalise Freeman's measure for edges, in order to find which edge is more between other pairs of vertices [29]. A high betweenness centrality represents a bridge-like connector. The formal definition for the betweenness centrality

of an edge (u, v) is given by $B(u, v)$, where:

$$B(u,v) = \sum_{i,j \in V(G)} \frac{\sigma_{ij}(u,v)}{\sigma_{ij}} \tag{2.5}$$

and where σ_{ij} is the number of shortest paths from i to j, and $\sigma_{ij}(u, v)$ is the number of shortest paths from i to j passing through edge (u, v).

Other edge based centrality measures are generalisations of vertex-based variations of betweenness centrality. For example, Sun et al. [72] instead modify betweenness centrality to include all paths (not just the shortest ones). Meo et al. [15] restrict their attention to paths flowing through an edge where the path length is restricted. They build their measure κ-path edge centrality based on Alahakoon et al.'s [4] vertex based κ-path centrality, a modification to betweenness centrality. Meo et al. [15] define the κ-path edge centrality $L_\kappa(u, v)$ of edge (u, v) by:

$$L_\kappa(u,v) = \sum_{s \in V(G)} \frac{\sigma_s^\kappa(u,v)}{\sigma_s^\kappa} \tag{2.6}$$

where s are all possible source vertices, σ_s^κ is the number of paths of length at most κ which start on source vertex s , and $\sigma_s^\kappa(u, v)$ is the number of these paths flowing through (u, v).

Fortuna et al. [25] take a centrality metric that makes long paths undesirable. The efficiency of a network is built on the assumption that the further apart two vertices are, the less efficient their communication, and thus is the inverse proportion to distance [45, 43]. The authors extended Latora and Marchio [44, 46] vertex based centrality metric based on network efficiency to an edge based measure. Fortuna et al. [25] define the information centrality $C^I_{(u,v)}$ of edge (u, v) by:

$$C^I_{(u,v)} = \frac{NE(G) - NE(G'_{(u,v)})}{NE(G)} \tag{2.7}$$

where G' denotes the graph obtained by removing the edge (u, v) from G, $NE(G)$ is

the network efficiency of G [45, 43], i.e:

$$NE(G) = \frac{1}{n(n-1)} \sum_{v_i \neq v_j \in V(G)} \frac{1}{d_{v_i v_j}} \tag{2.8}$$

Their measure is built on the reduction in network efficiency through the removal of an edge in order to determine how central such edge is.

Teixeira et al. [73] replace shortest paths flowing through an edge with minimum spanning trees, although this is only applicable to undirected and weighted networks present an edge betweenness based on the fraction of minimum spanning trees an edge is present. They define the spanning edge betweenness $\delta_G(u, v)$ as:

$$\delta_G(u, v) = \frac{\tau_G(u, v)}{\tau_G} \tag{2.9}$$

where τ_G is the number of minimum spanning trees for G and $\tau_G(u, v)$ is the number of these that run through edge (u, v).

In general, there are far fewer examples of edge-based centrality metrics than vertex based centrality metrics, and most of the centrality metrics discussed are simply generalisations of vertex based variations on betweenness centrality. Though these variations on betweenness centrality may paint a better picture of how social interactions work in the real world than betweenness centrality in its original form, they do not explicitly outline how the edges interact with one another. We can assess which edge is most central in terms of number of (specified) paths flowing through it, but we don't necessarily know much about adjacent edges to this edge (other than that there is another edge that forms a path with this edge). In other words, of the discussed literature, there is an apparent lack of edge based centrality metrics that take into account local features. This means we potentially ignore some information that may be critical to exploring how messages spread, as we know in complex networks such as social networks, local dissemination of information is crucial. Further, the lack of local information means that measures

require knowledge of the overall network (for example, knowledge of all shortest paths in the network, or the minimum spanning trees), making them computationally complex.

2.3.2 Substructure Based Centrality Metrics

While there are few edge based centrality metrics, there are far fewer subgraph-based centrality metrics. In this section we address Estrada and Rodríguez-Velázquez [19] subgraph centrality measure which aims to satisfy this gap in the literature. We also look to Przulj [66] graphlet degree distribution, which generalises the degree of a vertex by considering the number of orbits (pieces of induced substructures) incident with a vertex. The authors offer this to be used in degree distributions, although this could be used as a centrality metric to generalise degree centrality [27]. However, first we define network motifs [54]: those substructures which occur more frequently than expected in a network, which can be used to profile the networks themselves. [71, 55, 56].

Motifs and Triadic Census

A Triadic Census of a network counts the number of triads present for each of the 16 triad isomorphism classes given in Figure 1.1 [75]. Taking a Triadic Census provides a method of summarising information about a whole network into a vector of length 16. This was first introduced by Holland and Leindhart [38]. Triads which are over-represented by a Triadic Census on a network are known as network motifs. These were first discovered in gene-regulatory networks by Shen-Orr, Milo, Alon et al. [54] and generalised to any network by Milo et al. [55]. This approach involves normalising the frequency of induced triads against those which might occur in a sample of random networks. A motif profile allows us to categorise and identify particular types of network. A considerable number of studied have been conducted on motifs, though motif detection and remains computationally challenging, with many contributions to the literature in motif-mining algorithms. Other contributions focus on applying network

motifs to uncover deep information on network profiling [71, 55, 56]. Currently there appears to be limited interpretation of substructures in their own right. In other words, many analyses present the over or under representation of induced substructures with limited assessment of the agency that the induced substructures provide to the edges involved within them.

Subgraph Centrality Measure

Estrada and Rodríguez-Velázquez [19] introduce a subgraph centrality measure for complex networks. They define the subgraph centrality of a vertex $i \in V(G)$ as the sum of all closed walks starting and ending at vertex i, $C_s(i)$:

$$C_s(i) = \sum_{k=0}^{\infty} \frac{\mu_k(i)}{k!} \tag{2.10}$$

where $\mu_k(i)$ represents the local spectral moment, or the number of walks of length k starting and ending on vertex i. Their work is biased so that short walks have more influence on the subgraph centrality, to align with motifs in real networks being based on small induced subgraphs [19]. They define their subgraph centrality metric on closed walks in order to enable computation from the corresponding adjacency matrix of a graph, however this has some drawbacks. A path of length k could be a trivial walk that passes through the same vertex multiple times and therefore yields a smaller subgraph than a substructure on k edges.

Graphlet Degree Distribution

Przulj [66] doesn't build a new centrality metric based on subgraphs, but rather generalises the degree of a vertex, and uses this to generalise the degree distribution of a network. This could be used as a generalisation of degree centrality, although not explicitly used for this purpose in the paper. The author notes that an edge is simply a graphlet built on two vertices (a dyad) an the degree of a vertex is how many graphlets

on two vertices touch the vertex. Therefore, it is possible to generalise degree by extending the definition to a vertex touching a graphlet on more vertices. The authors define 29 specific graphlets (all possible undirected graphlets of orders two to five). This is complicated by the number of automorphism orbits a graphlet has, as a vertex that touches a graphlet can do so in different ways depending the structure of the graphlet. The 29 graphlets defined in [66] yield 73 total automorphism orbits; and the authors count the graphlet degree distributions for each of the 73 orbits.

Graphlet Orbits

Przulj's [66] automorphism orbits and graphlet degree distribution provides an important tool for analysing the local toplogy of vertices; used particularly in bioinformatics in papers such as in [35, 36]. This provides an alternative approach to the global profiling of networks using network motifs. These applications are based on the assumption that the vertex's local topology is related to some property [36].

Formally, an isomorphism on a graph is a bijective homomorphism from one graph to another ie.:

Definition 3. *[16] Let G and H be graphs on vertex set $V(G), V(H)$ and edge set $E(G), E(H)$ respectively. Let θ be a mapping from G to H such that $\theta : V(G) \rightarrow V(H)$. Then θ is an isomorphism if and only θ is bijective and if $\forall i, j \in V(G), ij \in E(G) \Leftrightarrow \theta(i)\theta(j) \in E(H)$.*

An automorphism is an isomorphism from one graph to itself. The set of all automorphisms on a graph G form a group, denoted $Aut(G)$. The automorphism orbit of a vertex i in G is all the set of all other vertices in G that map i under some automorphism from $Aut(G)$, i.e:

Definition 4. *[16] Let G be a graph on vertex set $V(G)$ and edge set $E(G)$, and $i \in V(G)$. Then the automorphism orbit of i, $Orb(i) = \{j \in V(G) | \theta(i) = j$ for some $\theta \in Aut(G)\}$.*

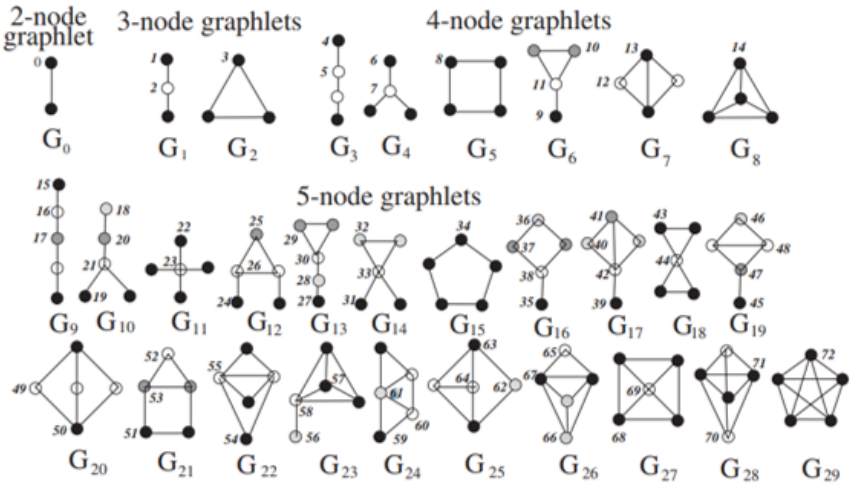


Figure 2.5: Automorphism Orbits of all possible undirected graphlets of up to five vertices. Each automorphism orbit class is labelled by a new number, whilst a new isomorphism class of graphlet is labelled by G_k.

Przulj [66] gives all automorphism orbits for undirected orbits of up to five vertices, shown in the image taken from their paper in Figure 2.5.

Automorphisms preserve structure within a graph, and thus in this case, the automorphism orbits of a vertex are all the other vertices within the same graphlet that are structurally the same. For example, in G_2, all vertices are structurally identical up to relabelling. However, in G_1, the central vertex acts differently from the two outer vertices as it always has a degree of two, whilst the external vertices have a degree of one. The outer vertices are structurally the same and therefore lie in the same orbit, whilst the central vertex lies in a different orbit. Therefore G_1 partitions into two automorphism orbits, whereas G_2 has one. These measures are largely vertex-based rather than edge based. Estrada and Rodríguez-Velázquez [19]'s subgraph centrality metric could be seen as another generalisation of betweenness centrality: this time, where instead of shortest paths flowing through a vertex, the focus is on the number of closed walks flowing through a vertex. There are some limitations to this measure, namely that it takes into account both trivial and non-trivial closed walks. This means that in graphs that contain multi-edges, the subgraph centrality is not specifically picking up graphlets

and could just be picking up dyads on multiple loops. Przulj [66] potentially builds a measure closer to what we are interested in with their generalised vertex degree - counting the number of automorphism orbits of a particular graphlet incident with a vertex. This does highlight the different positions of a vertex within a substructure having a different effect and so counts the orbits separately. However, all these measures are vertex based and predominantly concerned with volume of substructures rather than the behaviour of edges in substructures. Even Przulj, although investigating separate orbits, focuses on the volume of substructures touching a vertex, and not the effect the different positioning of vertices may have. Further, Przulj's work is restricted to undirected networks and therefore hasn't been applied to directed triads.

To conclude, we have looked at the most relevant centrality metrics given our interest in edges and induced substructures. We note that there are few edge based centrality metrics and even fewer induced substructure centrality metrics. Whilst we are able to find examples of centrality metrics on edges and centrality metrics on induced substructures, we have not found evidence for any measure that encompass both. Further, most centrality metrics we've discussed are based on *volume* of paths flowing through and edge (in the case of edge based centrality metrics) or volume of induced substructures incident with a vertex (in the case of substructure based centrality metrics).This aligns with motif profiling of networks, which focuses on the occurrence of induced substructures which appear more frequently than one would expect. This focus on volume ignores the behaviour of edges (based on their location and direction in relation to other edges within a substructure) within induced substructures, which collectively could have a large effect on message spread throughout a network.

2.4 Criticality Metrics for Networks

Edge criticality [79] (otherwise known as edge significance [77]) offers an alternative indication of edge importance in terms of maintaining network structure and connectivity.

This has emerged from percolation theory, which describes the behaviour of a network when vertices or edges are added. Critical edges are determined by gradually removing edges from a network, and observing the effect of their removal on the global network structure. Typically, the size of the giant component or the susceptibility index are used to judge the critical point at which a network breaks down, leading to an increase in the number of connected components and decrease in the size of connected components.

Perhaps somewhat counter intuitively, in social networks edges which may be most critical to network structure may be those with the weakest ties [61, 78]. Weak Ties theory [32] states that overlapping friendship circles strengthen ties between two individuals, and as such their removal will only cause the network to shrink through local disintegration but will not cause the collapse of the overall network [61]. In contrast, the removal of weak ties will cause quick network disintegration as the weak ties act as bridges between the highly connected communities [61]. The weak ties hypothesis [32] means that weak ties exhibit high betweenness centrality. This is unique to social networks and directly contradicts the global efficiency principle [30, 50] where the strength of a tie directly correlates with its betweenness centrality as it is optimised to create maximum network flow [61]. In most technological and biological networks the global efficiency principle holds true, and it is believed that the removal of these strong ties is believed to cause the collapse of a network [61].

Onnela et al. [61] observe the effect of removing ties in mobile phone networks on the network stability, weighting ties based on the aggregated duration of calls between individuals. They compare this to some other hypotheses, namely the dyadic hypothesis and the global efficiency principle [30, 50], showing that the weak ties hypothesis appears to hold true for mobile phone networks, and removing weak ties first causes fastest network collapse. Tie strength can be hard to determine, and the processes for doing so are probably complicated and time consuming [77]. Cheng et al. [77] and Yu et al. [79] offer alternative metrics to tie strength in order to determine an edge's criticality. Both papers [78, 79] base their criticality metrics on cliques, the

former using the bridgeness index [77], and the latter constructing their algorithm, Betweenness Centrality and Clique Model (BCCMod) [79] to determine critical edges. In the next sections (Sections 2.4 and 2.4) we discuss these metrics further. We note that betweenness centrality of edges [29] as defined in Equation 2.5 (which can also be used as a criticality metric on edges) is commonly used as benchmark comparisons within the literature. To understand measures built on cliques, we must first define a clique.

Definition 5. *Suppose G is a graph on vertex set $V(G)$ and edge set $E(G)$. Then the graph G' on vertex set $V(G')$ and edge set $E(G')$ where $V(G') \subset V(G)$ is a clique if and only if there is an edge between every pair of vertices in $V(G')$.*

Bridgeness Index

A bridge is a link between two vertices that joins two components that would be disconnected from each other if the bridge did not exist [3]. Cheng et al. [77] introduce the bridgeness index for edges based on clique sizes of the vertices included in an edge and the edge itself. They propose this as a local measure that is entirely topological, but exhibits the weak ties phenomenon [32]. They discuss the correlation of the bridgeness index with the strength of ties. Edges which act as a bridge are usually weak ties, but connect cliques together, and thus underpin the structure of the network - removing them causes the network to collapse into smaller connected components. The authors run their index on document networks, where two documents are more likely to be connected if they are conceptually relevant.

The bridgeness index [77] B of edge (u, v) in G is given by:

$$B = \frac{\sqrt{S_u S_v}}{S_{(u,v)}} \tag{2.11}$$

where S_u, S_v and $S_{(u,v)}$ denote the clique sizes (i.e the size of the maximum clique containing this vertex or edge [70, 69] of u,v and the edge (u, v) respectively).

The authors discuss that the bridgeness index leads to a faster network disintegration

than when compared with their comparison metrics. Therefore, edges with a high bridgeness index may be more critical to the structure of a network.

Betweeness Centrality and Clique Model

Yu et. al [79] create their criticality metric, The Betweeness Centrality and Clique model (BCCMod), in order to combine both local features of an edge in a network (specifically, degree of vertices and cliques) with global features (i.e betweenness centrality). The Betweeness centrality and clique model [79] of an edge (u, v) in G is defined by:

$$BCC_{MOD} = \frac{k_u k_v BC(u, v)}{\sum_{i=3}^{n} C(u, v)_i} \tag{2.12}$$

where $BC(u, v)$ is the betweenness centrality of (u, v), k_u and k_v are the degrees of vertices u and v respectively, $C(u, v)_i$ is the number of cliques of size i containing (u, v).

The authors show that BCCMod [79] often outperforms (in terms of collapsing the network) various comparable criticality metrics across a variety of networks (including social networks, technological networks, etc) when edges are removed in descending order and the susceptibility index is taken; meaning that edges with a high BCCMod score are the most critical to the structure of a network.

BCCMod [79], bridgeness index [77] and betweenness centrality of edges [29] are computationally complex, as they require knowledge of entire cliques, or all the paths that flow through an edge. However, considering edge criticality does highlight the importance of edges to a network; namely their importance to network structure through observing the disintegration of the network when they are removed. Criticality measures have the power to demonstrate that edges hold communities together in a network, something that cannot be determined from a vertex alone. Criticality measures built on cliques rely on network substructures, however cliques are not as fundamental as small induced substructures, such as triads.

To recap the themes visited across the literature review, we have summarised the information in Tables 2.1, 2.3 and 2.2.

Type of measure: classifying and counting triads based on their edges				
Measure	Formula	Description	Global / Local	Comments
Global clustering coefficient	$C = \frac{\text{Number of Closed Triplets}}{\text{Number of All Triplets}}$	Determines proportion of clustering present in network	Global	Can't be applied to weighted or directed networks.
Local clustering coefficient	$C_v = \frac{2\lvert\{(i,j)\in E(G) \lvert i,j\in N(v)\}\rvert}{k_v(k_v-1)}$	Measures cliquishness in neighbourhoods by pivoting on a single vertex.	Local	Can't be applied to directed networks; inverse correlation between node degree and local clustering coefficient.
Weighted clustering coefficient	$C_w = \frac{\text{Total Weight of Closed Triplets}}{\text{Total Weight of Triplets}}$	Generalised clustering coefficient for weighted networks.	Global	Can't be applied to directed networks.
Weighted Transitivity coefficient	$T_w = \frac{\text{Total Weight of Transitive Triplets}}{\text{Total Weight of Triplets}}$	Weighted clustering coefficient for directed networks.	Global	Can't be applied to directed networks.
Do not focus on edges within the triad play in maintaining the triad connectivity.				

Table 2.1: Recap of key themes in Section 2.2

Type of measure: centrality measures				
Measure	Formula	Description	Substructure / Edge	Comments
Betweenness centrality for edges	$B(u,v) = \sum_{i,j \in V(G)} \frac{\sigma_{ij}(u,v)}{\sigma_{ij}}$	Proportion of shortest paths passing through an edge.	Edge based	Requires global knowledge of all shortest paths.
Motifs	$Z(G') = \frac{F_G(G') - \mu_R(G')}{\sigma_R(G')}$	Finds triads which are over-represented in a network.	Substructure based	Highlights substructure importance; focuses on abundance, little interpretation of substructures in their own right.
Subgraph centrality	$C_s(i) = \sum_{k=0}^{\infty} \frac{\mu_k(i)}{k!}$	Sum of all closed walks starting and ending on the vertex. i	Substructure based	Counts closed loops, concerned with volume rather than behaviour of edges in substructures.
Graphlet Degree Distribution	$d_G^j(k)$ = number of vertices in G touching the graphlet k times on orbit j	Generalises degree of a node to number of graphlets it touches.	Substructure based	Concerned with volume rather than behaviour of edges in substructures.
No overlap between edge based and substructure based centrality measures.				

Table 2.2: Recap of key themes in section 2.3

Type of measure: edge criticality measures			
Measure	Formula	Description	Comments
Tie Strength	N/A	Assigns strength to an edge based on some data (eg. duration of phone calls).	Can be hard to determine; processes for doing so are probably complicated and time consuming.
Bridgeness Index	$B = \frac{\sqrt{S_u S_v}}{S_{(u,v)}}$	Indentifies edges which act as bridges between two otherwise disconnected subgraphs.	Requires knowledge of all cliques in graph.
Betweenness centrality for edges	$B(u,v) = \sum_{i,j \in V(G)} \frac{\sigma_{ij}(u,v)}{\sigma_{ij}}$	Proportion of shortest paths passing through an edge	Requires global knowledge of all shortest paths.
Betweeness centrality and clique model	$BCC_{MOD} = \frac{k_u k_v BC(u,v)}{\sum_{i=3}^{n} C(u,v)_i}$	Combines global knowledge (betweenness centrality) and local knowledge (degree of vertices and clique).	Requires knowledge of all cliques in graph.
No overlap between edge based and substructure based centrality measures.			

Table 2.3: Recap of key themes in Section 2.4

2.5 Data sets

To test, develop and compare concepts, it is useful to adopt open data sources that have been adopted in other studies. Accordingly we consider a range of such data sets that represent a spread of different scenarios. Networks of interest are collated from a variety of sources, including the Cosin Project [14], Stanford's Snap data sets [14], NeuroData's Network Database Connectomes [58], Toreopsahal.com data set collection [63], amongst others [74, 1, 5]. They have been organised by their category, which are: Airport, Electrical Circuit, Ecological Food web, Peer-To-Peer Internet, Neural, Organise, Metabolic, Regulatory, Trust/Social and World Wide Web networks.

We use these data sets as they represent different natural phenomena well established in the literature. Newman's Review [59] provides a good summary of empirical studies of the structure of networks and loosely categorises data structures into four types based on their source: social, biological, information and technological networks. These categories are deliberately loose as some networks can overlap more than one category (for example, the world wide web is both technological and social). We have further sorted our networks into sub-classes of Newman's categories. For example, Food web, Neural and Regulatory Networks are all biological networks.

Classifications of networks is an evolving area of research so we cannot fully describe any category of network in terms of its topology. What we do know is that different structural properties are exhibited by different classes of network. For example, Food web networks are generally very small (less than 200 vertices) due to the difficulty in recording every species in the environment and every interaction between them [14]. Some information networks, such as citation networks, do not contain closed loops [59]. Social networks may exhibit 'community structure ', where vertices are clustered in tightly knit groups, with only weak ties between different groups [29]. There are a number of properties established in the literature that real world networks have in common, namely the 'Small World 'effect [53], long-tailed degree distributions and clustering [76].

Name	$\|V\|$	$\|E\|$	$Density$	$Clustering$	$Reciprocity$	BC	DC	RC	$Diameter$
Airport Flights Networks									
US Airports [42]	1572	28235	0.011	0.50	0.78	0.0013	0.023	0.059	8
Open Flights [67]	2905	30442	0.0036	0.44	0.97	0.0010	0.0072	0.0072	14
Electrical Circuit Networks									
s420 [5]	252	399	0.0063	0.028	0.00	0.0062	0.013	0.66	13
s838 [5]	512	819	0.0031	0.027	0.00	0.0037	0.0063	0.67	15
Ecological Food Web Networks									
Mangwet [74]	97	1492	0.16	0.26	0.062	0.011	0.32	0.11	3
Baywet [74]	128	2106	0.13	0.18	0.029	0.0091	0.26	0.19	3
Little Rock Lake [42]	183	2476	0.074	0.17	0.034	0.00094	0.15	0.82	4
Ythan [1]	92	414	0.049	0.11	0.00	0.0011	0.099	0.79	4
St. Marks Seagrass [14]	49	223	0.095	0.13	0.00	0.0048	0.19	0.74	4
Grassland [14]	88	137	0.018	0.17	0.00	0.00033	0.036	0.97	6
Peer-to-peer Internet Networks									
p2p-gnutella04 [47]	10876	39994	0.00034	0.0031	0.00	0.00021	0.00068	0.6	10
p2p-gnutella05 [47]	8842	31837	0.00040	0.0036	0.00	0.00024	0.00081	0.6	9
p2p-gnutella06 [47]	8717	31525	0.00041	0.0033	0.00	0.00024	0.00083	0.61	10
p2p-gnutella08 [47]	6299	20776	0.00052	0.0054	0.00	0.00030	0.0010	0.63	9
p2p-gnutella09 [47]	8104	26008	0.00040	0.0048	0.00	0.00024	0.00079	0.65	10
Neural Networks									
C. Elegans [40]	297	2345	0.027	0.17	0.17	0.01	0.053	0.13	5
Drosophila Medilla 1 [58]	1770	9624	0.0030	0.15	0.15	0.00083	0.0061	0.14	6
Mouse Visual Cortex 2 [58]	193	214	0.0058	0.010	0.00	0.00	0.012	0.24	8
Mouse Retina 1 [58]	1076	90811	0.079	0.30	0.00	0.00046	0.16	0.56	4
Rattus Norvegicus [58]	503	27667	0.11	0.78	0.34	0.0018	0.22	0.0020	3
Organise Networks									
Cross Parker Consulting [63]	44	521	0.28	0.62	0.77	0.021	0.55	0.047	4
Freemans EIES n48 1 [63]	34	540	0.48	0.69	0.85	0.016	0.96	0.00	2
Freemans EIES n48 2 [63]	34	708	0.63	0.77	0.85	0.012	1.26	0.00	2
Cross Parker Manufacturing [63]	77	1452	0.25	0.67	0.80	0.012	0.50	0.013	3
Eva [41]	4475	4662	0.00023	0.0060	0.0043	0.00	0.00047	0.28	18
Regulatory Networks									
E. coli transcription [5]	328	456	0.0043	0.055	0.00	0.00	0.0085	0.020	13
Yeast transcription [5]	662	1063	0.0024	0.025	0.0019	0.00	0.0049	0.12	15
Trust/Social Networks									
Bitcoin Alpha [47]	3670	22639	0.0017	0.15	0.85	0.00065	0.0034	0.12	10
Bitcoin OTC [47]	5551	32007	0.0010	0.14	0.84	0.00042	0.0020	0.16	14
Email EU Core [47]	986	24929	0.026	0.37	0.71	0.0014	0.051	0.16	7
Prison Inmate [5]	67	182	0.040	0.23	0.44	0.034	0.082	0.34	7
UCIrvine [63]	1893	20292	0.0057	0.088	0.64	0.00080	0.011	0.29	8
WikiVote [47]	7066	103663	0.00208	0.082	0.056	0.00	0.0042	0.089	7
World Wide Web Networks									
Political Blogs[2]	1222	19021	0.013	0.22	0.24	0.0013	0.025	0.13	8

Table 2.4: Networks and metrics used to describe them. $|V|$ denotes the number of vertices in the network, $|E|$ the number of edges. $Clustering$ represents the mean global clustering coefficient. $Reciprocity$ is a measure of global reciprocity. RC, DC and BC denote the global reaching centrality, mean degree centrality and mean betweenness centrality respectively. $Diameter$ is taken using the undirected version of the network. Self loops have been removed from the original data set, so we are testing on the largest connected component of the original data set. Decimal places are rounded to two significant figures.

We have applied various standard metrics on our networks in effort to describe their structure further. These results are presented in Table 2.4. Here we compare network connectivity (through measuring their density, global clustering coefficient and mean degree centrality), directionality (in terms of the reciprocity and reaching centrality of a network) and also review the importance of vertices in paths (in this case, betweenness centrality).

Density represents the fraction of edges that exist out of all possible edges. Clustering coefficient [49] also indicates density of ties, but differs in that it indicates to what degree groups of vertices cluster together. A graph could have low density but could have an area of high clustering. Similarly, degree centrality represents the number of ties a vertex has, indicating importance in terms of popularity. From Table 2.4 we infer that generally Organise networks have a high level of clustering as well as a higher degree centrality, and are more dense than the other networks we compare. This could indicate that organise networks form tightly knit groups. Similarly, Airport networks also have a high level of clustering but a far lower degree centrality and are sparse. In contrast, e-circuit networks have low clustering and degree centrality. Peer-to-peer Internet networks and Regulatory networks present similar structure. Other networks sit somewhere in the middle. For example, food web networks present a mid level of clustering and degree centrality but have low density. Social networks are sparse, but have mid levels of clustering. Neural networks vary greatly depending on which network we are referring to: for example, the network Rattus Norvegicus [58] has a higher level of clustering, whereas the network Mouse Visual Cortex [58] has very low levels of clustering.

Betweeness centrality [26] also measures the importance of vertices, but unlike degree or reaching centrality, betweeness centrality measures how many shortest paths pass through a vertex. Betweeness centrality is low across all networks we studied. Diameter is also linked to shortest paths, measuring the longest shortest path in the network. Diameter varies greatly across our networks: with the largest diameter belonging to the Yeast Transcription Regulatory network (with a diameter of 15), and the shortest being shared by Freemans EIES 1 and 2 [63] networks (with a diameter of two). High diameter seems to occur in networks where density is lower.

Reciprocity represents the likelihood of vertices in a network being mutually linked [60]. Reciprocity is high across those networks in which clustering is high (Airport and Organise networks, with the exception of Eva [41]), but is also found to be high

in trust/social networks (with the exception of WikiVote [47]). This may be due the mutuality of to trust/social links. Reciprocity is low across networks where clustering and density (such as peer-to-peer Internet, Electrical circuit and Regulatory networks), but is also found to be low in food web networks. This is likely due to the way the food web networks are structured: a lack of reciprocity exists due to the likelihood that no two species predate each other. In networks where reciprocity is higher, reaching centrality tends to be lower and visa versa. Reaching centrality [57] is another directional metric, but instead of measuring mutual ties, captures the degree of hierarchy in a network. Reaching centrality is generally high in electrical circuit networks, ecological food web networks, peer-to-peer Internet networks and the Mouse Retina [58] network. Trust/social and Neural networks, as well as Political Blogs [2] and Eva [41] tend to present mid-levels of reaching centrality, with low levels being observed in Organise, Eegulatory and Airport networks.

2.6 Conclusions

In this chapter we have explored literature relating to the study of complex networks that encompasses the contributions made within this thesis. We first explored alternative classifications for triads based on the role of their edges. We looked to the global clustering coefficient [49], which works on the basis of closed and open triplets in a network. We also looked to the local clustering coefficient [76], and explored Tore and Opsahl's [64] generalisation of the local clustering coefficient for weighted networks. We note that clustering coefficient is not sufficient for directed networks, and so explore the more refined concept of transitivity [75] in networks; defining what it means for a triad to be transitive, vacuously transitive or intransitive. Tore and Opsahl [64] also generalise a transitivity coefficient for weighted complex networks. We conclude that whilst these measures are based on the existence of edges to classify triads, they do not focus on the role individual edges play within these triads or differentiate between the edges. This could have potential implications concerning the spread of messages

when edges with different roles are layered on top of one another. There is generally a gap in the literature for the role of edges, motivating our new classification for edges *(Contribution C1)* based on their role within triads in Chapter Four.

Additionally, in Chapter Four we transform our classification for edges into a local centrality metric for edges *(Contribution C2)*. In this literature review we explore various edge-based centrality metrics. Most notably, we look to Girvan and Newman [29] for their generalisation of Freeman's [26] vertex-based centrality metric to edges. We note that other edge-based centrality measures are generalisations of variations on the vertex-based betweenness centrality [72][15][25]. Since our classification of edges is based on their roles within induced substructures, our centrality metric in Chapter Four is not only an edge based centrality metric, but one also grounded in induced substructures. In this chapter we highlight other induced-substructure based centrality metrics such as Estrada and Rodríguez-Velázquez's subgraph centrality metric [19] or Przulj's graphlet degree distribution [66]. We conclude that there is little to no overlap between edge based and subgraph based centrality metrics.

Finally, we looked at edge criticality as an alternative measure of edge importance to edge centrality. Edge criticality [79] focuses on the importance of edges to maintaining structure in the overall network. We learn that in social networks (and potentially further classes of network) weak ties are fundamental to network structure [61], and whilst the removal of strong ties may result in a reduction in network size, the removal of weak ties quickly disintegrates the network into connected components [61]. We look to existing criticality metrics that are easily computable, specifically the bridgeness index of an edge [77] and the betweenness centrality and clique model [79]. Both of these measures are built on cliquishness of edges. In addition to our centrality measure's application to minimising spread (Chapter Five), it may highlight those edges most critical to the structural integrity of a network.

Chapter 3

Understanding graph Connectivity through Paths and Induced Triads

Within a graph, the way in which edges in support connectivity is influenced by their role in paths between vertices. This presents an interesting starting point for detailed analysis. Specifically when the unit of analysis is the triad, as often used in analysis of complex graphs, do paths of different types (such as shortest paths) tend to favour composition involving edges from particular types of triads? As far as we can establish, this is not known. It is also relevant to developing subsequent metrics on the connectivity role of edges in triads, as introduced in Chapter Four.

In Chapter One, we introduced triads and showed the 16 isomorphism classes for triads in Figure 1.1. In this chapter we explore which of the classes of connected triad occur most frequently across paths. A Triadic Census on a graph G counts the number of each class of induced triad present in G. This provides a method of summarising information about a whole graph into a vector of length 13. To assess the induced triads that are incident with important paths, we construct a novel measure: *the triadic edge betweenness census*, that counts the induced triads occurring on all shortest paths in a graph. This happens when the shortest path and an induced triad share at least one edge in common. This is an edge based Triadic Census measure, and for completeness in this context, we also construct the vertex-based equivalent: the *induced Triadic Census*.

Through this approach, we can compare these census measures against census measures

for the whole graph, using the data sets discussed in Section 2.5. This gives a basis to understand the under or over representation of induced triads that belong to shortest paths, as compared to the whole graph. Accordingly we are able to discover which triads are more prevalent in supporting shortest paths.

We also classify a triad according to how many of its edges can simultaneously occur in the same shortest path, enabling us to observe in which graphs the proportion of these induced triads occurring across shortest paths (i.e *the triadic edge betweenness census (Definition 25*) may be greater than across the proportion across the whole graph. This chapter serves as an introductory chapter to the rest of the thesis, where we further explore the role of the individual edges within triads (Chapter Four). Later we apply this to shortest paths to observe the effect on message spread (Chapter Five) and develop it further in the context of edge criticality (Chapter Six).

3.1 Overview

Triadic Census was first introduced by Holland and Leindhardt [39], who used it to construct various subgraph configurations (subgraphs in which only some of the edges between vertices are of interest [39]) and compared subgraph counts in real-world graphs against those found in random graphs. A Triadic Census is valuable as it summarises the information about a graph into a vector of length 13. Further, structural properties about a graph, such as transitivity [75], can be tested using a Triadic Census (for an overview, see Waserman and Faust [75]).

Faust [20] argues that much of a Triad Census can be explained by lower order graph properties, specifically the dyad census and graph density, because they constrain triadic outcomes. Dyads are the building blocks of triads and thus, for example, networks with a high volume of asymmetric dyads (edges with no reciprocated links) will contain mostly triads of type `030T` and `030C` [20]. Therefore, these two lower order properties constrain the triad census. However, Faust [20] also argues that in social networks,

substantial triadic tendencies occur which are only partially explained by the dyad census. Whilst the region of possible triad census which can occur based on network density and dyad census is relatively small, and comparison of networks with different dyad censuses largely explains their differing triad censuses, there is large variability between triad census in networks with similar dyad profiles. Therefore, we can deduce that triad census of a network offers new information, that cannot be wholly derived from the dyad census and network density, about a network in a concise way.

Triads which are over-represented in a Triadic Census are known as network motifs [55]. This approach involves normalising the frequency of induced triads against those which might occur in a sample of random networks. The authors do this through introducing the Z-score for a particular triad type, i.e:

Definition 6. *The Z-score for triad type i in graph G is given by:*

$$Z_i = \frac{Nreal(i) - < Nrand(i) >}{std(Nrand(i))} \tag{3.1}$$

where $Nreal(i)$ is the number of times triad i appears in graph G and $< Nrand(i) >$ and std(Nrand(i)) are the mean occurrence and standard deviation of i in a randomised esemble of random graphs with the same degree sequence as G [55].

The significance profile (SP) of i is the normalised Z-score, computed by dividing through by the sum of Z-scores for all 13 triads, in order to understand the relative presence of a particular triad type i.e:

Definition 7. *The significance profile for triad type i in graph G is given by:*

$$SP_i = \frac{Z_i}{(\sum_{i=1}^{i=13} Z_i^2)^{1/2}} \tag{3.2}$$

where Z_i is the Z-score of triad i in G [55].

Profiling networks by observing their motifs allows us to categorise and identify particular types of network. A considerable number of studied have been conducted on motifs,

though motif detection and remains computationally challenging. At present, Z-scores and significance profiles for networks are based on static networks. Further, exploring the motifs present in a network does is preoccupied with identification: there is little interpretation of substructures in their own right. Finally, observing only structures that appear more frequently than at random ignores triads which may hold significant influence over network structure or connectivity, but are not themselves frequently occurring. Instead, we propose to explore connectivity within a graph through modifying the triadic census.

To use the concept of Triadic Census in the context of connectivity, we need to begin by considering the basic concept of a *path*. In graph theory, a path p is defined as a walk where all vertices are distinct. Paths are of significance to graphs as they represent chains of interaction between individual vertices. In particular, we may want to know whether it is possible to deliver a message from one vertex to another through a collection of intermediaries (something exhibited, for example, in neurological graphs, where neurons communicate via electrical impulses propagated from one neuron to another [24]; or could represent chains of infection from individual to individual). Paths are inherent to many important measures and properties within graphs. The existence of path in a graph underpins the definition for connectivity: whether every vertex is reachable every other through a path.

Definition 8. *A walk from u to v in a graph $G = (V(G), E(G))$ is a sequence of the form $w(u, v) = (u = u_0, u_1, u_2, \ldots, v = u_n)$ where $u_0, u_1, \ldots u_n \in V(G)$ and for $1 \leqslant i \leqslant n, (u_{i-1}, u_i) \in E(G)$.*

Definition 9. *A path from u to v in a graph $G = (V(G), E(G))$ is a sequence of the form $p(u, v) = (u = u_0, u_1, u_2, \ldots, v = u_n)$ where $u_0, u_1, \ldots u_n$ are **distinct** vertices in $V(G)$ and for $1 \leqslant i \leqslant k, (u_{i-1}, u_i) \in E(G)$. The length of the path is given by the number of edges, and denoted by $l(p) = |p(u, v)| - 1$.*

Amongst different types of path, shortest length paths are fundamental. In many real world systems, the shortest path represents the most efficient system to minimise cost or

time moving within the network. For example, many mobile map apps are developed using shortest paths to minimise the time it takes to travel from one point to another. Betweenness centrality [26] is also built on shortest paths, where a vertex [26] (or an edge [29]) is determined more important based on the number of shortest paths that overlap on the edge. The average shortest path length can highlight how tight-knit communities are, for example: small world networks are those which exhibit small average shortest path length [76].

Paths underpin such an important structural property of graphs, and Triadic Census may yield much information about a graph, yet the relationship between a Triadic Census and the triads that a shortest path induces is not well understood. However, the occurrence of triads across paths may have a part to play in the dissemination of information (or infection) in graphs. Therefore, in this chapter we explore which triads are most prevalent (or important) to paths. Some triads may be more present in shortest paths than elsewhere in the graph, or visa versa. In particular – we ask whether the Triadic Census is an accurate or useful proxy for the characteristics of a graph when paths between vertices are the primary concern.

In this chapter, we modify the Triadic Census to investigate the volume of each triad class which have at least one edge in common with a shortest path in a graph. We call this measure Triadic Edge Betweenness Census. Since this is an edge based measure, we construct the vertex based equivalent, Induced Triadic Census, for comparison with Triadic Census. We also construct the edge based equivalent of Triadic Census, All Edge Census, for comparison with Triadic Edge Betweenness Census. In doing so, we investigate which triads may be more present across shortest paths than at a network level, and therefore may have a greater role in maintaining network connectivity. Further, we also investigate which triads may be more present at network level but have a lesser role in shortest paths; which may demonstrate the lack of information about network connectivity gained when solely looking at a network's motifs. We proceed to run our census measures on a handful of networks and make these comparisons, noting that

triads can be categorised according to how many of their edges can be simultaneously contained within the same shortest path.

3.2 Methodology for Path-based Census

3.2.1 Triadic Census on Graphs

The following definitions are introduced to enable Triadic Census to be carried out in the context of paths.

Let $\mathcal{T}$ be the ordered set of triad types of interest (the 13 possible connected classes of triads from Figure 1.1) i.e:

$$\mathcal{T} = (\texttt{021D}, \texttt{021U}, \texttt{021C}, \texttt{111D}, \texttt{111U}, \texttt{030T}, \texttt{030C}, \texttt{201}, \texttt{120D}, \texttt{120U}, \texttt{120C}, \texttt{210}, \texttt{300})$$

Let $t_k(G)$ denote the number of distinct triads (i.e triads which have at most two vertices in common) of type $k \in \mathcal{T}$ present in G.

Definition 10. *A Triadic Census on G is given by $T(G)$, where:*

$$T(G) = (t_k(G))_{k \in \mathcal{T}}$$

To calculate $T(G)$ in this thesis, we apply the Triadic Census algorithm proposed by Vladimir Batagelj and Andrej Mrvar [8], to compute the Triadic Census designed by Holland and Leindhart [37]. Algorithm 1 adapts and simplifies Batagelj and Mrvar's algorithm by ignoring triads `003`, `102` and `120` from calculations, which represent the possible disconnected triads. Before we do this, we first remind ourselves of the relevant information from Batagelj and Mrvar [8], defining the functions $Link$ and $Tricode$ as follows:

Definition 11. *Let $G = (V(G), E(G))$ be a graph and let $u, v \in V(G)$ such that $u \neq v$. Then $Link(u, v)$ [8] is defined as:*

$$Link(u,v) = \begin{cases} 1 & \text{if } (u,v) \in E(G) \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

Definition 12. *Let $G = (V(G), E(G))$ be a graph and let $u, v, w \in V(G)$ such that $u \neq v \neq w$. Then $Tricode(v, u, w)$ [8] is defined as:*

$$\begin{aligned} Tricode(u,v,w) &= Link(u,v) + 2(Link(v,u) + 2(Link(u,w) + & (3.4)\\ & 2(Link(w,u) + 2(Link(v,w) + 2Link(w,v))))) & (3.5) \end{aligned}$$

$Tricode$ outputs an integer value which is mapped to a unique triad type under the function $TriType$. The mapping is given by Table 3.1. To demonstrate the $Link$ and $Tricode$ functions computing the triad types formed by three vertices, we refer to Example 3.2.1.

Example 3.2.1. *Suppose $G = (V(G), E(G))$ is a simple graph on vertices $V(G) = \{x, u, v, w\}$ and edges $E(G) = \{(x, u), (u, v), (v, w)\}$, as shown in Figure 3.1.*

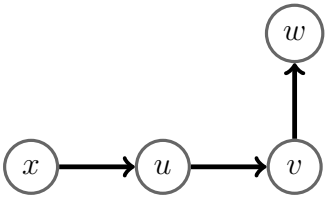


Figure 3.1: Graph $G = (V(G), E(G))$ where $V(G) = \{x, u, v, w\}$ and $E(G) = \{(x, u), (u, v), (v, w)\}$.

Then $Link(u, v) = 1, Link(x, u) = 1$ and $Link(v, w) = 1$ since $(u, v), (x, u)$ and $(v, w) \in E(G)$.

Then following Definition 12, $Tricode(u, v, w) = 1 + 2(0 + 2(0 + 2(1 + 2(0)))) = 9$.

code	type	code	type	code	type	code	type
0	003	16	012	32	012	48	102
1	012	17	021C	33	021U	49	111D
2	012	18	021D	34	021C	50	111U
3	102	19	111U	35	111D	51	201
4	012	20	021U	36	021C	52	111D
5	021D	21	030T	37	030T	53	120D
6	021C	22	030T	38	030C	54	120C
7	111U	23	120U	39	120C	55	210
8	012	24	021C	40	021D	56	111U
9	021C	25	030C	41	030T	57	120C
10	021U	26	030T	42	030T	58	120U
11	111D	27	120C	43	120D	59	210
12	102	28	111D	44	111U	60	201
13	111U	29	120C	45	120U	61	210
14	111D	30	120D	46	120C	62	210
15	201	31	210	47	210	63	300

Table 3.1: Triad codes and their corresponding triad type under function $TriType$ [8].

According to Table 3.1, $TriType(9) =$ 021C. Thus triad (u, v, w) is of type 021C. *Similarly $Tricode(x, u, v) = 9$, and $TriType(9) =$ 021C, thus (x, u, v) is of type* 021C.

Using the $Tricode$ formula and Table 3.1 we introduce the algorithm CENSUS, a Triadic Census ($T(G)$) on a graph G (Algorithm 1).

Algorithm 1 only considers neighbouring vertices v once u is determined (and similarly, w must be a neighbour of u and v). This prevents the need to run the $Tricode$ function on every triplet of vertices in a graph through only considering those which form a triad. We define the neighbourhood of a vertex in Definition 13:

Definition 13. *Let $G = (V(G), E(G))$ be a graph and let $u \in V(G)$. Then the neighbourhood of u, $N(u)$ is defined by:*

$$N(u) = \{v \in V(G) | (u, v) \in E(G)\} \tag{3.6}$$

Looping over all triples of vertices must be done with care, as each may appear in six different orderings. To ensure every non-null triad is counted exactly once in Algorithm 1, Batagelj and Mrvar [8] index vertices by integers. Hence, we give each vertex in the graph a unique integer value, and say the ordering if canonical if for $u, v, w \in V(G)$: $u < v < w$,

Algorithm 1 CENSUS(G): Performs Triadic Census $T(G)$ [8]

Input: Graph G.
Output: Triadic Census vector.
$t_k = 0 \; \forall k \in \mathcal{T}$ ▷ Initialise counts
for $u \in V(G)$ **do**
  for $v \in N(u)$ such that $u < v$ **do** ▷ Count each pair once only
    for $w \in N(u) \cup N(v) - \{u, v\}$ **do** ▷ Only consider connected triads
      if $v < w$ or ($u < w$ and $w < v$ and $(u, w), (w, u) \notin E(G)$) **then**
        $k = TriTypes[Tricode(u, v, w)]$
        $t_k = t_k + 1$
return $T(G) = (t_k)_{k \in \mathcal{T}}$

We are also interested in finding the Triadic Census for a single edge in a graph. This later enables us to build more complex census measures along paths, such as the Path Induced Triadic Census (Definition 22) or Triadic Edge Betweenness census (Definition 25). An edge census (Definition 16) on edge (u, v) finds all (non-trivial) triads which contains (u, v). To construct an edge census, we first define the *induced subgraph* of G on a subset, S:

Definition 14. *Let $G = (V(G), E(G))$ be a graph. For $S \subseteq V(G)$, the induced subgraph of G on S, denoted $G[S]$, is the graph with vertex set S and edge set $E(G[S]) = \{(u, v) | u, v \in S, (u, v) \in E(G)\}$.*

Defining the induced subgraph of G on S allows us to construct a Triadic Census on an induced subgraph, i.e:

Definition 15. *A Triadic Census $T_G(S)$ for a subset of vertices $S \subseteq V(G)$ is defined as:*

$$T_G(S) = T(G[S]) \tag{3.7}$$

where $G[S]$ denotes the induced subgraph of G on S.

We are now in a position to define a Triadic Census on a single edge:

Definition 16. *A Triadic Census $T_G(u, v)$ on an edge $(u, v) \in E(G)$ is defined by:*

$$T_G(u, v) = \sum_{w \in V(G) - \{u,v\}} T_G(\{u, v, w\}) \tag{3.8}$$

To calculate $T_G(u, v)$, we apply EDGECENSUS (Algorithm 2), a modification to CENSUS (Algorithm 1).

Algorithm 2 EDGECENSUS(G, u, v): Performs Triadic Census $T_G(u, v)$ of an edge (u, v).

Input: Graph G, $(u, v) \in E(G)$.
Output: Triadic Census vector.

1: $t_k = 0 \; \forall \; k \in \mathcal{T}$ ▷ Initialise counts
2: **for** $w \in N(u) \cup N(v) - \{u, v\}$ **do** ▷ Only consider connected triads
3: $\quad k = TriTypes[Tricode(u, v, w)]$
4: $\quad t_k = t_k + 1$
5: **return** $T_G(u, v) = (t_k : k \in \mathcal{T})$

We want to investigate the difference between the Triadic Census performed on shortest paths as compared to the whole graph for a range of data sets. Specifically, we want to explore which triads may be found more frequently than we would expect across shortest paths, as compared to the whole graph. To achieve this we introduce two forms of census: (Definition Path Induced Triadic Census $I(G)$ (Algorithm 22) and Triadic Edge Betweenness Census $B(G)$ (Algorithm 25). Path Induced Triadic Census will be built on vertices and so can be compared with Triadic Census (a census on the whole

graph). Triadic Edge Betweenness Census will be built on edges, and therefore we require an edge based metric equivalent to Triadic Census. We therefore build the adaptation: Edge Triadic Census $A(G)$ 17):

Definition 17. *The Edge Triadic Census on G is defined as $A(G)$ where:*

$$A(G) = \sum_{(u,v)\in E(G)} T_G(u,v) \tag{3.9}$$

We compute the Edge Triadic Census ($A(G)$) by running a Triadic Census on each edge in turn. Hence, $A(G)$ counts the number of triads of each type that each edge is contained within.

Constructing EDGETRIADICCENSUS requires summing two censuses, which we define below (Definition 18):

Definition 18. *Let S and T denote two Triadic Censuses on a graph G, where $S = (s_k)_{k\in\mathcal{T}}$ and $T = (t_k)_{k\in\mathcal{T}}$. Then the sum of censuses, $S + T$ is defined by:*

$$S + T = (s_k + t_k)_{k\in\mathcal{T}}. \tag{3.10}$$

EDGETRIADICCENSUS (Algorithm 3) is an adaptation on CENSUS (Algorithm 1) to compute $A(G)$. The complexity of the algorithm depends on the number of edges in the graph, since it runs EDGECENSUS on each edge. That is: for each edge in the graph, each $w \in N(u) \cup N(V) - \{u, v\}$ must be found, the $Link$ function must be applied six times (once per possible edge between v, u and w) and the $Tricode$ function applied once.

Algorithm 3 EDGETRIADICCENSUS(G): Computes the Edge Triadic Census $A(G)$ on graph G.

Input: Graph G.

Output: Edge Triadic Census vector $A(G)$.

1: $A = (0)_{k \in \mathcal{T}}$
2: **for** $(u, v) \in E(G)$ **do**
3: $\quad A = A +$ EDGECENSUS(G, u, v)
4: **return** A

Note that individual triads containing more than one edge contribute to $A(G)$ multiple times. Consider Example 3.2:

Example 3.2.2. *Suppose there exists some graph* $G = (V(G), E(G))$ *on vertices:* $V(G) = \{u, v, u_1, u_2, u_3\}$ *and edges* $E(G) = \{(u, v), (u_1, u), (u_2, u), (v, u_3)\}$ *as shown in Figure 3.2. Then the Triadic Census on the edge* (u, v) *(Definition 16) is given by*

$T_G(u, v) = (0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$. *That is,* (u, v) *is contained in three triads:* $t_1 = (u, v, u_3), t_2 = (u_1, u, v)$ *and* $t_3 = (u_2, u, v)$*, which are all of the form* `021C`*.*

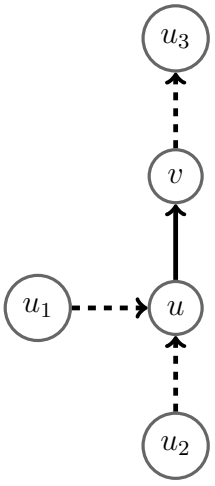


Figure 3.2: Graph $G = (V(G), E(G))$ **where** $V(G) = \{u, v, u_1, u_2, u_3\}$ **and** $E(G) = \{(u, v), (u_1, u), (u_2, u), (v, u_3)\}$**.**

The edge (u_1, u) *is contained in two triads:* t_2 *(above, of form* `021C`*) and* $t_4 = (u_1, u, u_2)$

of type `021U`. *Hence* $T_G(u_1, u) = (0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$. *Note that* t_1 *has been counted twice, once in* $T_G(u, v)$ *and once in* $T_G(u_1, u)$.

Similarly, edge (u_2, u) *is contained in two triads:* t_3 *(above, of type* `021C`*) and* t_4 *(above, of type* `021U`*.) Hence* $T_G(u_2, u) = (0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$. *Note that* t_3 *has been counted twice* $T_G(u, v)$ *and once in* $T_G(u_2, u)$*; and* t_4 *has been counted twice, once in* $T_G(u_1, u)$ *and once in* $T_G(u_2, u)$.

Edge (v, u_3) *is contained in one triad:* t_1 *(above, of type* `021C`*). Then* $T_G(v, u_3) = (0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$. *Note that* t_1 *has been counted twice* $T_G(u, v)$ *and once in* $T_G(v, u_3)$.

Then the Edge Triadic Census, $A(G) = T(u_1, u) + T(u_2, u) + T(u, v) + T(v, u_3) = (0, 2, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$*, but this has only been constructed from four triads:* t_1, t_2, t_3 *and* t_4.

In general, Edge Triadic Census ($A(G)$) counts every triad once per edge of the triad. Edge Triadic Census and Triadic Census ($T(G)$, Definition 10) assess the relative presence of each triad across a graph. Triadic Census is vertex-centric, whilst Edge Triadic Census uses edges.

We are now in a position two build the two new census measures we will be investigating: Path Induced Triadic Census (22) and Triadic Edge Betweenness Census (Definition 25). These census measures will assess the relative presence of each triad across a graph (with Path Induced Triadic Census being vertex-based, whilst Triadic Edge Betweenness Census is edge-based); offering a direct comparison to Edge Triadic Census $A(G)$ and Triadic Census $T(G)$.

3.2.2 Triadic Census on Paths

Since shortest paths are at the focus of our new measures, we first introduce the set of all shortest paths between a pair of vertices.

Definition 19. *Suppose $G = (V(G), E(G))$ is a graph and let $u, v \in V(G)$. Then a shortest path between u and v is a path between u and v which contains the minimum number of edges. We denote the length of a shortest path from u to v as $\delta_s(u, v)$.*

Definition 20. *Suppose $G = (V(G), E(G))$ is a graph and let $u, v \in V(G)$. Let $P(u, v)$ denote the set of all paths between u and v. Then the set of all paths of least length between u and v, $P_s(u, v) \subseteq P(u, v)$ is denoted by:*

$$P_s(u, v) = \{p \in P(u, v) | l(p) = \delta_s(u, v)\} \tag{3.11}$$

We define a Path Induced Triadic Census between vertices u and v by summing the Triadic Census $T_G(e)$ for each edge e in path $p \in P_s(u, v)$. This is normalised over the number of paths in $P_s(u, v)$.

Definition 21. *Let $P_s(u, v)$ be the set of all shortest paths between $u, v \in G$. Then a Path Induced Triadic Census $I_G(u, v)$ between u and v is given by:*

$$I_G(u, v) = \frac{1}{|P_s(u, v)|} \sum_{p \in P_s(u,v)} T_G(p) \tag{3.12}$$

By running a Path Induced Triadic Census (Definition 21) on every pair of vertices, we can induce a Path Induced Triadic Census on a graph, i.e:

Definition 22. *The Path Induced Triadic Census on G is given by $I(G)$ where:*

$$I(G) = \sum_{u,v \in V(G)} I_G(u, v) \tag{3.13}$$

To calculate $I(G)$, we apply Algorithm 5, PATHINDUCEDTRIADICCENSUS, which performs a TRIADCENSUS (Algorithm 1) on each induced subgraph of $G[p]$ for each

shortest path p, where each shortest path is generated using Dijkstra's shortest path algorithm [17].

Algorithm 4 PATHINDUCEDTRIADICCENSUS(G): Computes the Path Induced Triadic Census $I(G)$ on a graph G.

Input: Graph G.
Output: Triadic Census vector.
1: $I = (0)_{k \in \mathcal{T}}$
2: $\{P_s(u,v) : u, v \in V(G), u \neq v\} =$ DIJKSTRA_ALL_PAIRS(G) ▷ Find all shortest paths using Dijkstra's Algorithm [17]
3: **for** $u, v \in V(G)$ such that $u \neq v$ and $P(u,v) \neq \varnothing$ **do**
4: $\quad T = (0)_{k \in \mathcal{T}}$
5: $\quad$ **for** $p \in P(u,v)$ **do**
6: $\quad\quad T = T +$ CENSUS($G[V(p)]$)
7: $\quad I = I + \frac{1}{|P(u,v)|} T$
8: **return** I

Path Induced Triadic Census has the property of counting triads multiple times according to how many shortest paths the vertices of a triad are contained in. Further, $I(G)$ will only count triads present in the induced subgraph $G[p]$ of a shortest path p on G, which are precisely the triads with all three vertices present in p. We demonstrate in Examples 3.2.3 and 3.2.4 two different scenarios: in Example 3.2.3 we show that $G[p]$ only contains triads where all three of it's vertices are all present in p, and in 3.2.4 we shown that $I(G)$ counts each triad in G the same number of times as the number of shortest paths all three vertices of the triad are contained in.

Example 3.2.3. *Suppose there exists some graph $G = (V(G), E(G))$ as shown in Figure 3.3. The* `030T` *triad $t = (u_2, u_4, u_5)$ is highlighted in red.*

If p is a shortest path from u_1 to u_8 in G, then $p = (u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8)$. Then the induced subgraph of p on G, $G[p]$, contains all the same vertices and edges as G, thus is identical to Figure 3.3. We see in this figure that t is present in $G[p]$.

Now suppose we modify G through adding the edge (u_1, u_3) and call the resulting graph G', as shown in Figure 3.5, where edge (u_1, u_3) is shown in blue. This creates a shortcut

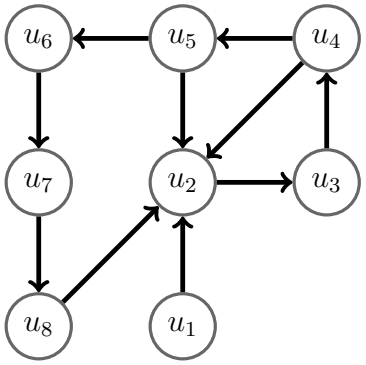


Figure 3.3: Graph G, with the `030T` triad t highlighted in red.

from u_1 to u_3 bypassing u_2, hence u_2 will no longer be contained in shortest paths from u_1 to u_8.

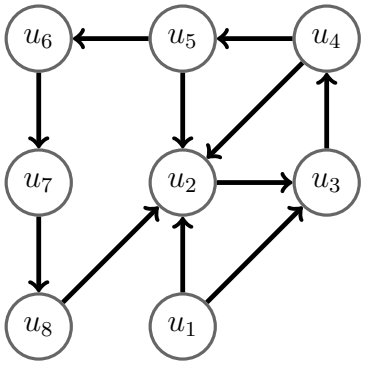


Figure 3.4: Graph G', a modification to G through adding the blue edge (u_1, u_3).

Hence if p' is a shortest path from u_1 to u_8 in G', then $p = (u_1, u_3, u_4, u_5, u_6, u_7, u_8)$. Then the induced subgraph $G'[p']$ is shown in Figure 3.4. We see in this figure that t is not present in $G'[p']$ because $u_2 \in t$ is not present in p', and therefore edges incident with u_2 will not be present in $G'[p']$.

Hence, since an induced subgraph of a shortest path p on G contains the vertices present p, then $G[p]$ will only contain triplets of vertices where all three vertices are present in p.

Example 3.2.4. *Consider Figure 3.3 and triad $t = (u_2, u_4, u_5)$ again. Then the vertices of t are all simultaneously contained in the following shortest paths: $p_1 =$*

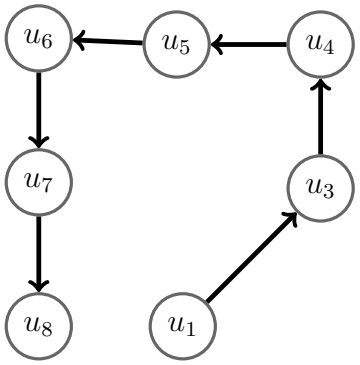


Figure 3.5: Induced subgraph $G'[p']$ of shortest path p' on G'.

$(u_1, u_2, u_3, u_4, u_5), p_2 = (u_1, u_2, u_3, u_4, u_5, u_6), p_3 = (u_1, u_2, u_3, u_4, u_5, u_6, u_7)$ *and* $p_4 = (u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8)$. *Then* t *will be present in the induced subgraphs* $G[p_1], G[p_2], G[p_3]$ *and* $G[p_4]$. *Hence* $I(G)$ *will count* t *four times, once per shortest path of which all three vertices of* t *occur.*

Suppose a message was transmitted through a graph from vertex u to v. We are interested in the number of triads of each type that such a message would encounter or pass through, assuming it is sent along a shortest path. In particular, we are interested in whether certain triad types are more prevalent on shortest paths than in the whole graph. Since there are likely to be several shortest paths between every such pair of vertices, we follow the same approach as the betweenness centrality measure, and take their average census.

We first define a Triadic Edge Betweenness Census on a shortest path, then build to a Triadic Edge Betweenness Census on a graph. Triadic Edge Betweenness Census looks at every triad in the graph G incident with the vertices in path p.

Definition 23. *Let* $p = (u = u_0, u_1, \ldots, v = u_n)$ *be a path of length* n *between* u *and* $v \in V(G)$. *A Triadic Edge Betweenness Census on* p *is given by* $B_G(p)$ *where:*

$$B_G(p) = \sum_{i=1}^{n} T_G(u_{i-1}, u_i) \tag{3.14}$$

We next calculate the Triadic Edge Betweenness Census over all possible shortest paths between u and v. This is normalised over the total number of shortest paths between u and v.

Definition 24. *Let $P_s(u, v)$ be the set of all shortest paths between $u, v \in G$. Then a Triadic Edge Betweenness Census for u, v is given by $B_G(u, v)$:*

$$B_G(u, v) = \frac{1}{|P_s(u, v)|} \sum_{p \in P_s(u,v)} B_G(p) \tag{3.15}$$

Finally, we can define the Triadic Edge Betweenness Census on a graph:

Definition 25. *The Triadic Edge Betweenness Census $B(G)$ on graph G is defined as:*

$$B(G) = \sum_{u,v \in V(G)} B_G(u, v) \tag{3.16}$$

To calculate $B(G)$, we apply Algorithm 5: TRIADICEDGEBETWEENNESSCENSUS. Triadic Census of edges are cached once computed, hence in TRIADICEDGEBETWEENNESSCENSUS, an EDGECENSUS (Algorithm 2) is only completed on each each unique edge once.

$B(G)$ will count triads multiple times, according to how many edges of the triad are contained in each shortest path. To demonstrate this, consider Example 3.2.5.

Example 3.2.5. *Suppose there exists some graph $G = (V(G), E(G))$ where $V(G) = \{u_1, u_2, u_3, u_4,$*
$u_5\}$ and $E(G) = \{(u_1, u_2), (u_2, u_3), (u_3, u_4), (u_4, u_5)\}$ as shown in Figure 3.6. Suppose $p = (u_1, u_2, u_3, u_4, u_5)$ is a shortest path of length four between u_1 and u_5.

There are two factors which affect how many times each triad is counted by $B(G)$: the degree to which shortest paths overlap and how many edges of a triad are contained in a shortest path.

Algorithm 5 TRIADICEDGEBETWEENNESSCENSUS(G)

Input: Graph G.
Output: Triadic Census vector.

$B = (0)_{k \in \mathcal{T}}$
$ec \leftarrow$ empty dictionary ▷ Cache for edge Triadic Censuses
$\{P_s(u, v) : u, v \in V(G), u \neq v\} =$ DIJKSTRA_ALL_PAIRS(G)
for $u, v \in V(G)$ such that $u \neq v$ and $P(u, v) \neq \varnothing$ **do**
    $T = (0)_{k \in \mathcal{T}}$
    for $p \in P_s(u, v)$ **do**
        for $(a, b) \in E(p)$ **do**
            if $(a, b) \notin ec$ **then** ▷ First encounter of edge
                $ec[(a, b)] =$EDGE_CENSUS(G, a, b) ▷ Calculate and cache
            $T = T + ec[(a, b)]$ ▷ Retrieve from cache
    $B = B + \frac{1}{|P_s(u,v)|} T$
return B

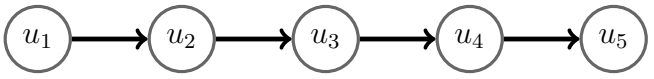


Figure 3.6: Graph $G = (V(G), E(G))$ on vertices $V(G) = \{u_1, u_2, u_3, u_4, u_5\}$ and edges $E(G) = \{(u_1, u_2), (u_2, u_3), (u_3, u_4), (u_4, u_5)\}$.

The triad $t_1 = \{u_1, u_2, u_3\}$ is a triad of type `021C` *in p. Two edges of t_1 are contained in p: (u_1, u_2) and (u_2, u_3). Then $B(G)$ will count t_1 twice in p.*

Similarly, the triads $t_2 = \{u_2, u_3, u_4\}$ and $t_3 = \{u_3, u_4, u_5\}$ will all be counted twice in p. Hence, the number of edges of a triad which are contained in a shortest path affects the number of times each triad is counted. In addition, the number of overlapping shortest paths affects how many times a triad is counted.

There are several subpaths of p which are themselves shortest paths:

- *There are four subpaths of length 1: $p_1 = (u_1, u_2), p_2 = (u_2, u_3), p_3 = (u_3, u_4), p_4 = (u_4, u_5)$. Depending on position in p, each subpath may lie on one or the intersection of two* `021C`*s: p_1 and p_4 both lie on one* `021C`*; p_2 and p_3 both lie on two* `021C`*s.*

- *There are three subpaths of length 2:* $p_5 = (u_1, u_2, u_3), p_6 = (u_2, u_3, u_4), p_7 = (u_3, u_4, u_5)$. *Paths* p_5 *and* p_7 *lie on three* `021C`*s;* p_6 *lies on four* `021C`*s.*

- *There are two subpaths of length 3:* $p_8 = (u_1, u_2, u_4, u_4)$ *and* $p_9 = (u_2, u_3, u_4, u_5)$. *Both lie on five* `021C`*s.*

Then $B(G)$ *counts 32 total* `021C`*s, even though there are only three distinct (i.e they at most share one edge)* `021C`*s in* G.

Even though some of these counts may count triads multiple times, this is not necessarily something that needs to be eliminated, but rather a measure of importance. For example, $B(G)$ will count those triads with more edges in shortest paths or which are contained in many overlapping shortest paths more frequently than to the remaining triads in the graph. The proportion of the census will therefore tend towards triads which are more important to constructing shortest paths. Likewise, $A(G)$ will count more frequently those triads with a greater number of edges, and therefore of greater importance to edges structure of the graph.

We have introduced three new measures: Edge Triadic Census $A(G)$, Path Induced Triadic Census $I(G)$ and Triadic Edge Betweenness Census $B(G)$. Each of these census measures produce a vector of length 13. In all cases, it makes sense to use a normalised vector to enable comparisons between the metrics. We shall define these normalised vectors as: $\hat{A}(G), \hat{B}(G)$ and $\hat{I}(G)$, with the addition of the normalised Triadic Census $\hat{T}(G)$. We define what it means to be a normalised vector below (Definition 26:

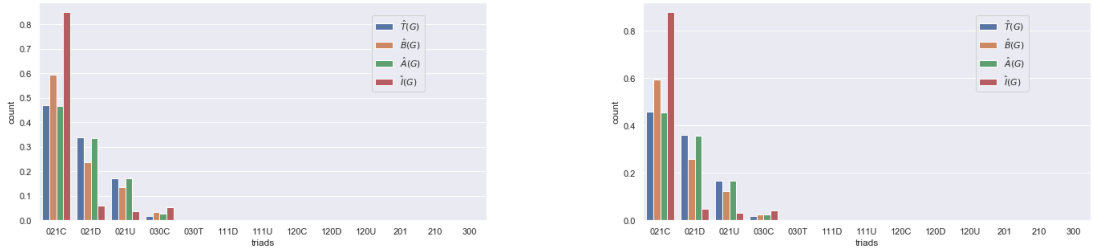
Definition 26. *Suppose* $x = (x_1, x_2, \ldots, x_n)$ *is a vector of length* n. *Then we define the normalised vector* $\hat{x}$ *as:*

$$\hat{x} = \frac{(x_1, x_2, \ldots, x_n)}{\sum_{k=1}^{n} x_k} \tag{3.17}$$

3.3 Results

In this section, we compare the relative presence of triads in shortest paths against their presence across the general network. We do this through comparing vertex based measures (i.e $\hat{T}(G)$ from Definition 10 and $\hat{I}(G)$ from Definition 22) and edge based measures (i.e $\hat{A}(G)$ from Definition 17 and $\hat{B}(G)$ from Definition 25), where $\hat{B}(G)$ and $\hat{I}(G)$ are based on shortest paths. These measures have all been normalised according to Definition 26. We are doing this because we wish to determine whether particular triad types are more prevalent, and therefore possibly more important to shortest paths. To analyse the results, we categorise triads into three sub-types: Type I, II and III triads based on the number of edges a triad may have in a shortest path. This also allows us to hypothesise which of these sub-types of triad may be higher in $\hat{B}(G)$ and $\hat{I}(G)$, and therefore potentially be more important in paths, and which of these sub-types of triad may be higher in $\hat{T}(G)$ and $\hat{A}(G)$.

We apply our four census measures on a combination of networks from Section 2.5. Due to complexity of Algorithms 1 - 5, our census measures could run to completion only on smaller networks in a reasonable time. The networks in question are: the electrical circuit networks s420 [5] and s838 [5]; the food web networks Baywet [74], Grassland [14], Little Rock Lake [42], Mangwet [74], St.Marks Seagrass [14] and Ythan [1]; the organise networks Cross Parker Consulting [63], Freemans EIESn48 1 [63], Freemans EIES n48 [63] and Cross Parker Manufacturing [63]; the regulatory networks E.coli transcription [5] and yeast transcription [5]; the neural network C.Elegans [68] and the social network Prison Inmate [5]. All of these networks have a small number of vertices, with the largest data set being C.Elegans on 297 vertices, and the smallest data set being Freemans EIES n48 1 and 2 on 34 vertices. The data sets have varying properties, demonstrating that our measures are applicable to a number of different network types, and ensuring that our results are collated from a range of sources to minimise bias from the network type. On one hand, organise networks are relatively dense and exhibit high levels of clustering, reciprocity and small diameters; indicating



(a) s420 [5] (b) s838 [5]

Figure 3.7: Census Measures of Electrical circuit networks.

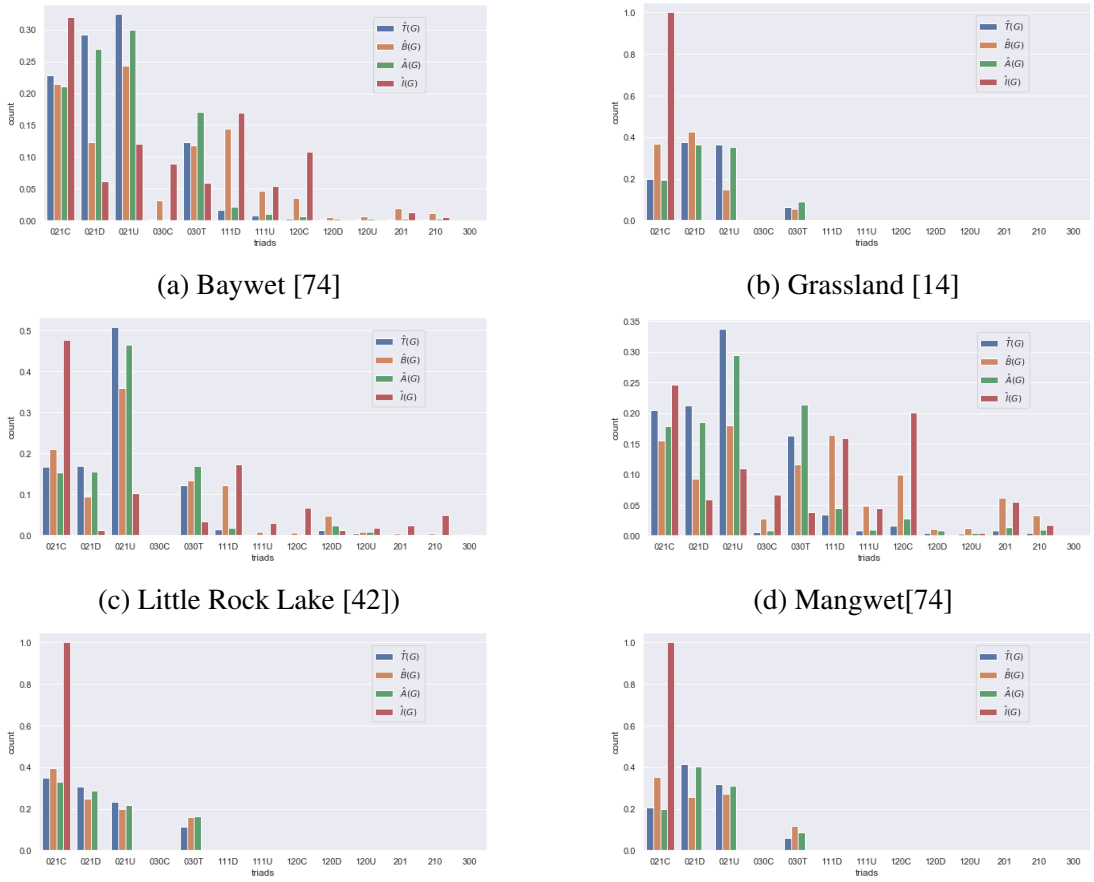
that vertices form highly connected tightly knit groups with lots of two-way interactions. Contrast this with e-circuit networks, which are relatively sparse, exhibiting very little reciprocity and clustering, and long diameters. Other classes of networks sit somewhere in between: for example, food web networks have small diameters, with medium levels of clustering and density (which could indicate grouping of vertices), yet exhibit little to no reciprocity as mutual predation in species is very unlikely.

For each network, we compute the four census measures by running each of the Algorithms 1 - 5 in turn on the input network and normalising each census measure by dividing through by the sum of the elements under Definition 26. For each network, we plot the number of triads occurring according to each census measure (represented through four coloured bars) for each class of triad. The results for this can be found in Figures 3.7 - 3.11. $\hat{T}(G)$ is represented by a blue bar, $\hat{I}(G)$ a red, $\hat{B}(G)$ an orange and $\hat{A}(G)$ a green. We compare vertex based measures $\hat{T}(G)$ (blue bar) with $\hat{I}(G)$ (red bar); and edge based measures $\hat{A}(G)$ (green bar) with $\hat{B}(G)$ (orange bar).

The relationship between the presence of a triad in shortest paths relative to the whole network seems to be dependent on two factors:

1 The maximum number of edges of a triad that can be simultaneously contained in the same shortest path.

2 The maximum number of shortest paths that overlap on one edge of a triad.



(a) Baywet [74]

(b) Grassland [14]

(c) Little Rock Lake [42])

(d) Mangwet[74]

(e) St.Marks Seagrass [14])

(f) Ythan [1]

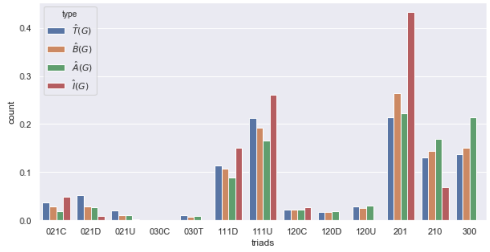
Figure 3.8: Census Measures of Food web networks.

We categorise all connected triads into following three types according to (1): the maximum number of edges simultaneously contained in the same shortest path:

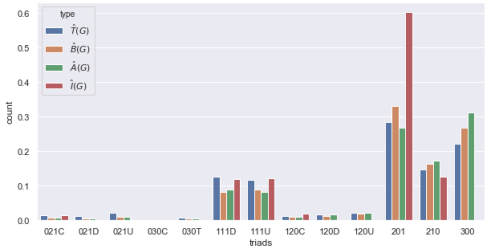
- Type I triads have at most one edge in a path.
- Type II triads have can have up to two edges simultaneously in a shortest path.
- Type III triads have two edges in a path, but at most one edge in a shortest path.

By exhaustive analysis of all connected triads, it is possible to uniquely specify each triad as having one type. This is presented in Table 3.2.

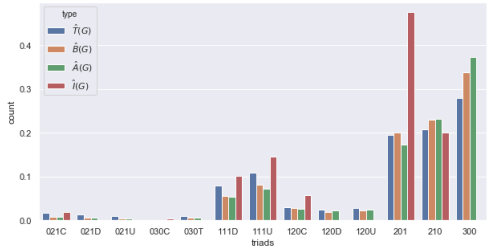
`030T`, `120D`, `120U` and `300` are all Type III triads. They all contain 2-paths; but every



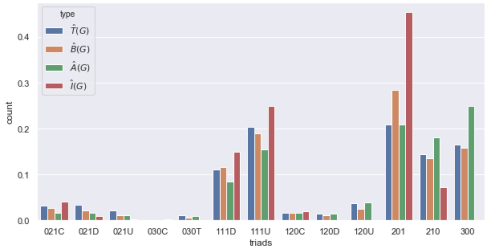
(a) Cross Parker Consulting [63]



(b) Freemans EIESn48 1 [63]

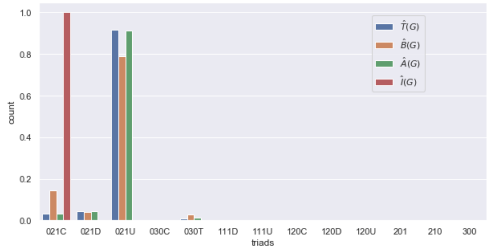


(c) Freemans EIES n48 [63]

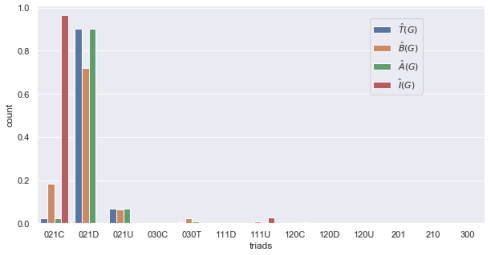


(d) Cross Parker Manufacturing [63]

Figure 3.9: Census Measures of Organise networks.



(a) E.coli transcription [5]



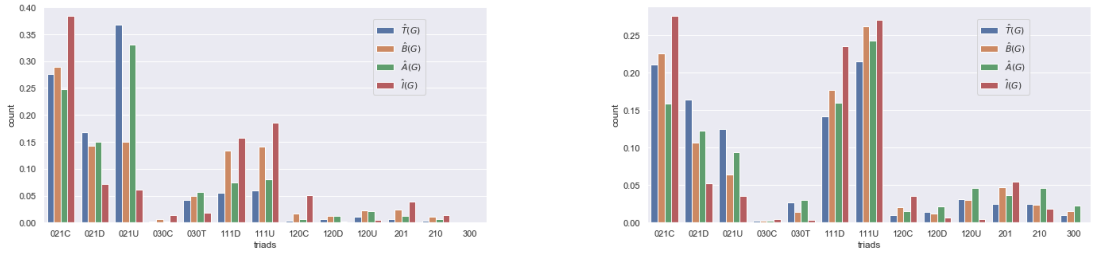
(b) Yeast transcription [5]

Figure 3.10: Census Measures of Regulatory networks.

2-path has a corresponding edge which 'shortcuts' the 2-path, and therefore the 2-path would not be in a shortest path. This is demonstrated in Figure 3.12.

Type I	Type II	Type III
021D	021C	030T
021U	111D	120D
	111U	120U
	030C	300
	201	
	120C	
	210	

Table 3.2: 13 connected triad types categorised into Type I, Type II and Type III.



(a) C.Elegans [68]) (b) Prison Inmate [5]

Figure 3.11: Census Measures of miscellaneous networks.

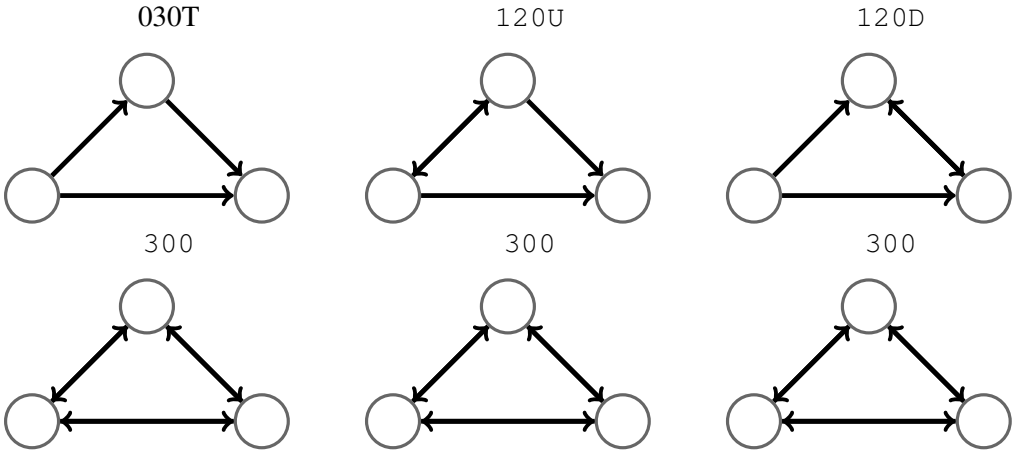


Figure 3.12: `030T`, `120D`, `120U` and `300`. The red edges represent a 2-path, with the corresponding shorcut in blue.

The categorisation of triads by type allows us to observe high-level patterns concerning the prevalence of triads. Accordingly we present Observations O1 - O3.

Observation O1: In general, Type I triads are more prevalent across the networks as compared to shortest paths.

From Figures 3.7 - 3.11 we observe:

- $\hat{T}_{021U}(G) > \hat{I}_{021U}(G)$ and $\hat{A}_{021U}(G) > \hat{B}_{021U}(G)$.
- $\hat{T}_{021D}(G) > \hat{I}_{021D}(G)$ and $\hat{A}_{021D}(G) > \hat{B}_{021D}(G)$ (except in Baywet [74]).

Observation O2: In general, Type II triads are more prevalent across shortest paths as compared to the whole networks.

From Figures 3.7 - 3.11 we observe:

- $\hat{T}_{021C}(G) < \hat{I}_{021C}(G)$ and $\hat{A}_{021C}(G) < \hat{B}_{021C}(G)$ (except in Mangwet [74]).
- $\hat{T}_{030\mathtt{C}}(G) < \hat{I}_{030\mathtt{C}}(G)$ and $\hat{A}_{030\mathtt{C}}(G) < \hat{B}_{030\mathtt{C}}(G)$.
- $\hat{T}_{111\mathtt{D}}(G) < \hat{I}_{111\mathtt{D}}(G)$ and $\hat{A}_{111\mathtt{D}}(G) < \hat{B}_{111\mathtt{D}}(G)$.
- $\hat{T}_{111\mathtt{U}}(G) < \hat{I}_{111\mathtt{U}}(G)$ and $\hat{A}_{111\mathtt{U}}(G) < \hat{B}_{111\mathtt{U}}(G)$.
- $\hat{T}_{120\mathtt{C}}(G) < \hat{I}_{120\mathtt{C}}(G)$ and $\hat{A}_{120\mathtt{C}}(G) < \hat{B}_{120\mathtt{C}}(G)$.
- $\hat{T}_{120\mathtt{D}}(G) < \hat{I}_{120\mathtt{D}}(G)$ and $\hat{A}_{120\mathtt{D}}(G) < \hat{B}_{120\mathtt{D}}(G)$.
- $\hat{T}_{120\mathtt{U}}(G) < \hat{I}_{120\mathtt{U}}(G)$ and $\hat{A}_{120\mathtt{U}}(G) < \hat{B}_{120\mathtt{U}}(G)$.
- $\hat{T}_{201}(G) < \hat{I}_{201}(G)$ and $\hat{A}_{201}(G) < \hat{B}_{201}(G)$.

To explain observations O1 and O2, we consider the number of edges of the triad that can be contained in a shortest path. Type I triads cannot have more than one edge in a path. Therefore, for every shortest path containing the Type I triad; $\hat{B}(G)$ will only count the triad once. Type II triads can have up to two edges in a shortest path. Therefore, for every shortest path containing the Type II triad; $\hat{B}(G)$ will count the triad up to two times.

An induced subgraph of each shortest path of length two on G computed in the algorithm for $\hat{I}(G)$ cannot contain a triad of Type I. Further, we know induced subgraphs will only contain triads where all three vertices are contained in a shortest path. Since Type II triads may contain two edges in a shortest path, this is more likely than in Type II triads than in Type I triads, where at most one edge is contained in a shortest path.

Hence when computing $B(G)$ and $I(G)$ there will be a greater number of Type II triads present than in $T(G)$ and $A(G)$. The same is not true for Type I triads, which can be more present in $B(G)$ and $I(G)$ than in $T(G)$ and $A(G)$ but at a rate lower than Type II triads, or potentially be even less present in $B(G)$ and $I(G)$ than in $T(G)$ and $A(G)$. Hence there will be a greater proportion of Type II triads computed by $\hat{B}(G)$ and $\hat{I}(G)$ and a low proportion of Type I triads as compared with $T(G)$ and $A(G)$.

Type III triads are more complex. We can expect them to behave similarly to Type I triads, due to only having at most one edge in a shortest path. However, there are many examples in Table 3.2 where this is not true. Hence, we make the following observation on Type III triads (O3).

Observation O3: Type III triads are generally more prevalent in paths or networks depending on (2): the maximum number of shortest paths that overlap on one edge of a triad. This depends on the structure of the network.

From Figures 3.7 - 3.11 we observe:

- $\hat{T}_{030T}(G) > \hat{I}_{030T}(G)$ and $\hat{A}_{030T}(G) > \hat{B}_{030T}(G)$ in Baywet [74], Little Rock Lake [42], Mangwet [74], St. Marks Seagrass [14], Metabolic, C. Elegans [40], Cross Parker Consulting [63], Freemans EIES n48 1 [63], Freemans EIES n48 2 [63], Cross Parker Manufacturing [63] and Prison Inmate [5].
- $\hat{T}_{030T}(G) < \hat{I}_{030T}(G)$ and $\hat{A}_{030T}(G) < \hat{B}_{030T}(G)$ in Ythan [1], E. coli transcription [5] and Yeast transcription [5].
- $\hat{T}_{120U}(G) > \hat{I}_{120U}(G)$ and $\hat{A}_{120U}(G) > \hat{B}_{120U}(G)$ in Cross Parker Consulting [63], Freemans EIES n48 1 [63], Freemans EIES n48 2 [63], Cross Parker Manufacturing [63] and Prison Inmate [5].
- $\hat{T}_{120U}(G) < \hat{I}_{120U}(G)$ and $\hat{A}_{120U}(G) < \hat{B}_{120U}(G)$ in Baywet [74], Little Rock Lake [42], Mangwet [74] and C. Elegans [40].

- $\hat{T}_{\texttt{120D}}(G) > \hat{I}_{\texttt{120D}}(G)$ and $\hat{A}_{\texttt{120D}}(G) > \hat{B}_{\texttt{120D}}(G)$ in Cross Parker Consulting [63], Freemans EIES n48 1 [63], Freemans EIES n48 2 [63], Cross Parker Manufacturing [63] and Prison Inmate [5].
- $\hat{T}_{\texttt{120D}}(G) < \hat{I}_{\texttt{120D}}(G)$ and $\hat{A}_{\texttt{120D}}(G) < \hat{B}_{\texttt{120D}}(G)$ in Baywet [74], Little Rock Lake [42], Mangwet [74] and C. Elegans [40].

We might expect Type III triads to behave similarly to Type I triads due to the property that at most one of their edges may be contained in a shortest path. Hence for every shortest path containing the Type III triad, $\hat{B}(G)$ will not count this triad twice. $\hat{I}(G)$ will only count this triad if all three of its vertices are contained in the shortest path; which is less likely in a triad that has at most one edge contained in a shortest path, and impossible in shortest paths of length two.

However, from the results we observe that Type III triads do not always behave like Type I triads. In many cases, primarily in Food web networks, Type III triads behave like Type II triads (namely they are more prevalent across shortest paths than the network).

The behaviour of Type III triads may instead be due to (2) : the maximum number of shortest paths a triad is contained within. This is due to the underlying structure of the network.

We note that all the networks where Type III triads behave like Type II triads are hierarchical. They have a high abundance of `021C`, `021D` and `021U` and crucially, a low abundance of the Type III triad of concern. Therefore, it is important to note it is possible that a single Type III triad acts as a bridge in these networks, that creates many new paths between vertices that would not otherwise have a path between them.Therefore, the number of these triads present in the network can be low, but the number of shortest paths it occurs in can be high. It is also important to note that a single Type I triad may have the same effect, but these are already very abundant in the network and therefore are not more prevalent in paths. We exemplify this in Example 3.3.1:

Example 3.3.1. *Suppose there exists some graph* $G = (V(G), E(G))$ *where* $V(G) = \{u_1, u_2, u_3, u_4,$
$u_5, u_6, u_7, u_8, u_9\}$ *as shown in Figure 3.13.* G *is hierarchical. The only triads* G *contains are* `021C`*,* `021U` *and* `021D`*, examples of which are highlighted in blue, red and green respectively.*

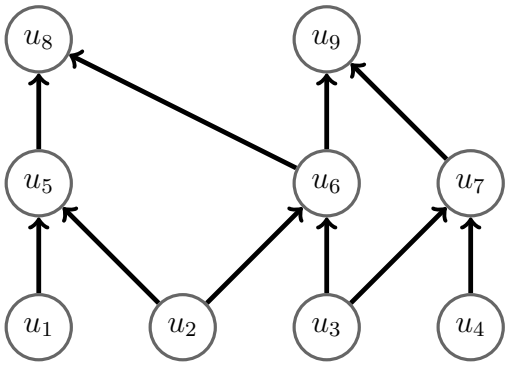


Figure 3.13: A heirarchy graph G**, with examples of** `021C`**,** `021U` **and** `021D` **highlighted in blue, red and green respectively.**

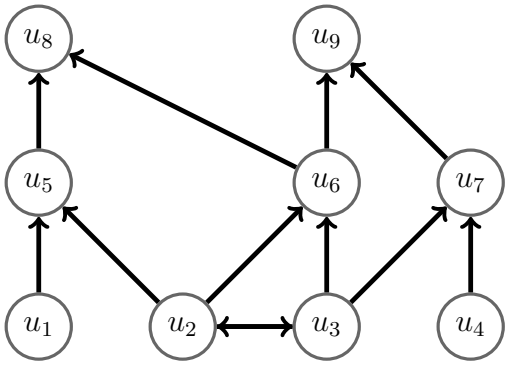


Figure 3.14: The addition of the edges (u_2, u_3) **and** (u_3, u_2) **to** G **from Figure 3.13**

Suppose we add a reciprocated edge to the graph between vertices u_2 *and* u_3 *in Figure 3.14. Adding this extra edge between* u_2 *and* u_3 *forms the* `120U` *triad:* $t = (u_2, u_3, u_6)$*. There is only one* `120U` *triad in the entire graph. However, adding this edge has created many new paths in the graph. This is because now there exists a path from* u_2 *to every neighbour of* u_3 *which is not a neighbour of* u_2*; and visa versa. The span of the new paths is highlighted in red. The new paths in question are:*

$p_1 = (u_2, u_3), p_2 = (u_2, u_3, u_7), p_3 = (u_2, u_3, u_7, u_9), p_4 = (u_3, u_2), p_5 = (u_3, u_2, u_5), p_6 = (u_3, u_2, u_5, u_8)$.

Since these paths are the only paths between the source and terminal vertex, they are also the shortest path between the two vertices in question. Therefore, although there is only one `021U` *present in the graph, it has many shortest paths incident with one edge.*

3.4 Summary

In this Chapter we have shown that a Triadic Census for a graph does not adequately represent the role of triads in shortest paths. In other words, when a census is induced by shortest paths, a different profile is exhibited. To demonstrate this, we have constructed two new census measures that use Triadic Census in different ways: $\hat{B}(G)$ and $\hat{I}(G)$. $\hat{B}(G)$ counts triad based on edges contained in shortest paths, whilst $\hat{I}(G)$ is vertex-based and counts triads based on presence in induced substructures of shortest paths. We have compared these measures with their respective edge or vertex based graph census measures: $\hat{A}(G)$ and $\hat{T}(G)$.

The importance of a triad to shortest paths is determined by two factors: the maximum number of edges of a triad that can be simultaneously contained in the same shortest path and the maximum number of shortest paths that overlap on one edge of a triad.

Furthermore, we have shown that triads can be usefully categorised according to how many of their edges can be contained in a single shortest path. Triads that can only contain up to one edge in a shortest path are more prevalent across the census for the graph than the census for shortest paths. On the other hand, triads that may contain up to two edges in a shortest path are more prevalent across shortest paths than the rest of the graph.

Triads that may contain two edges in a path, but only one edge in a shortest path can act either way; their importance to shortest paths is instead determined by the maximum

number of shortest paths that overlap on one edge of the triad, which is determined by the structure of the graph. These triads are more important to shortest paths in graphs with a higher volume of `021C`, `021D` and `021U` but a low volume of the triad in question. Otherwise, these triads are more prevalent in the graph than across shortest paths.

Overall, these results show that connectivity (i.e., shortest paths) in graphs leave particular signatures through their incidence with certain triads over others. This arises from the way in which edges in triads can be used to provide connectivity, as indicated through the categorisation of triads introduced in this chapter. This motivates further examination of the role of edges in triads, concerning their role in supporting graph wide connectivity.

This chapter addresses the first research question:

RQ1: The role of triads in paths: if connectivity in a graph is dependent on the existence of paths, then what is the relationship between triads and paths?
We do this through adapting triadic census to look at which triads occur more frequently in paths, resulting in the first contribution:
C1 *New triad census measures to establish the proportion of triads that occur more frequently along shortest paths, as compared with the entire graph. This addresses research question* **RQ1**.

Chapter 4

The Centrality of Edges Based on their Role in Induced Triads

In this chapter we use connected triads to characterise the local importance of an edge, based on its role in providing connectivity within a particular triad. As demonstrated in the last chapter, particular triads play more significant roles in supporting network connectivity through shortest paths. Here we address this by further differentiating the roles that edges within triads play in terms of supporting connectivity. We introduce a new form of edge centrality that captures the role that an edge plays in providing connectivity relative to a particular triad in a directed network. Critically, we note that relative to facilitating a path to the third node, an edge can be characterised in one of two alternative states, which we call *overt* and *covert*.

We introduce these concepts and demonstrate them through weighting edges of a network according to how many triads such an edge occurs in as overt or covert. This represents a new centrality measure for networks. We examine 34 publicly available networks representing a range of different scenarios and plot the distribution of overt and covert centrality, using this to observe the inherent similarities and differences between networks. For a range of synthesised networks, we also consider the differences in structure between minimally overt/covert paths as compared to shortest paths.

4.1 Overview

As presented in the literature review (Chapter Two), analysing complex networks through induced substructures (a.k.a graphlets) has gained much popularity in recent years. This approach assumes that induced substructures are the basic 'building blocks' for the network. This form of network analysis has mainly supported inter-network comparison based on assessing the relative volume of induced substructures (i.e those graphlets which occur significantly more in a network than in an ensemble of comparable random networks are called motifs). Much less attention has been paid to using graphlets to better understand the connectivity within a network.

In particular, the presence of particular substructures remains important and our results from Chapter Three indicate that certain categories of induced triad can be important to sustaining connectivity, based on edges supporting shortest paths. Recall we are interested in triads because they are the smallest non-trivial graphlet after dyads and vertices, yet still partition into relatively few isomorphism classes as compared with tetrads or higher order substructures. The categorisation we introduced in Chapter Three came from the positioning of edges within triads, and the viability of shortest paths being able to take multiple edges from a triad.

Taking this a step further, in this chapter we consider the role that edges within an induced substructure can play locally, in terms of edges supporting paths *within* a graphlet. This is a very local consideration of connectivity, and presents an additional perspective through which network connectivity can be characterised. Furthermore, this complements the existing use of graphlet analysis for networks, leading to a new centrality measure aligned to graphlet based analysis.

4.1.1 Edge Centrality and Graphlets

Centrality is a general concept for assigning importance to an edge or node for some particular reason. In this chapter we introduce a new centrality metric for edges that

addresses edge importance based on the connectivity roles that an edge plays for *multiple* triads. Within one triad in isolation, we have a very simple scenario, that can be built upon. As triads have just three nodes, connectivity from a node can be characterised by:

1. A node's direct reach (i.e., can the node communicate directly with other nodes);
2. A node's potential for indirect communication with other nodes (i.e., a path of length two to another node).

These aspects govern how local communication can take place in triads and the role that individual edges play, which can be seen by imaging potential for communication between nodes in a triad. In this scenario, an edge may facilitate direct communication only between a pair of nodes i, j in the triad (i.e., there is an edge (i, j) but no edge (j, k)) or an edge (i, j) may facilitate all communication to all nodes in the network (i.e., there is also a edge (j, k) in the triad where k is the third node). This means that an edge can have different significance based on the presence of other edges in the triad.

As a further example, consider Figure 4.1. Edge (i, j) holds a different role in Triad One than in Triad Two, because in Triad Two, once the edge (i, j) is traversed, it is still possible to traverse the edge (j, k) to reach node k. We call this the ability to 'flood' the triad. If we were sending a message from node i and we were interested in limiting dissemination, it would be desirable to have network structure in Triad Two rather than Triad One. The reverse would be true if our intent was contagion. We also note that an edge can play both these roles simultaneously for different triads when an edge is contained in multiple triads.

In this chapter we formally define this ability of an edge to support flooding a triad in the manner above as *overt*, otherwise we classify an edge as *covert*. Using edges in this way is convenient for a number of reasons – firstly it is a simple binary definition that can be used to assess the local connectivity in triads. Secondly it can be used to interpret and compare different triads, should graphlet analysis be the basis for comparison between networks. Thirdly the definition allows for overlap between triads to be assessed, noting

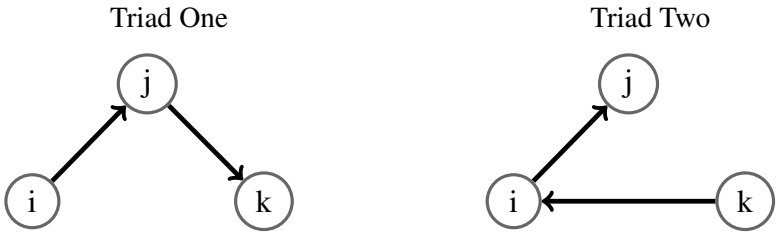


Figure 4.1: Different roles edges can play in connected triads. In Triad One, the edge (i, j) can support flooding the triad, but it does not in Triad Two.

also that an edge can play different roles (covert and overt) across the triads in which it participates. Finally we note that the concept can be imposed to assess relations between nodes of particular interest to a scenario. Thus the extent that an edge is overt or covert in the triads that it participates represents a basis for a centrality measure, that expresses the importance of edges relative to induced triads. We proceed by formalising these concepts.

4.2 Defining the Role of Edges

In this section, we formalise the concept of overt and covert edges. We assume a graph $G = (V(G), E(G))$ is represented by the set of nodes $V(G)$ and the set of edges $E(G)$. We assume that edges are directed, with an edge from node u to v if and only if $(u, v) \in E(G)$. For an induced triad t, we refer to its vertices as $V(t) \subseteq V(G)$ and its edges as $E(t) \subseteq E(G)$.

Definition 27. *Let t be an induced triad in $G = (V(G), E(G))$ with vertex set $V(t) = (i, j, k)$ where there is a directed edge from i to j (i.e., $(i, j) \in E(t)$). Edge (i, j) is overt with respect to t if and only if $(j, k) \in E(t)$, otherwise edge (i, j) is covert.*

Note that in a triad representing communication involving nodes i, j and k, when (i, j) is an overt edge, dissemination of content from i to j may reach k indirectly from i, via j. Conversely, when (i, j) is covert, i is assured that dissemination to j will not reach k. Therefore local awareness of the status of edges gives nodes some agency in either

the containment of information along a path involving that edge (i.e., the edge (i, j) is covert), or the potential for content to be more widely disseminated across a network due to the use of that edge (i.e., the edge (i, j) is overt). Example 4.2.1 demonstrates this.

Example 4.2.1. *For example, consider the triad* t *in Figure 4.2 where* $V(t) = \{i, j, k\}$ *and* $E(t) = \{(i, j), (j, k)\}$. *Then* (i, j) *is overt because* (j, k) *is contained in* $E(t)$ *but* (j, k) *is covert as no edge* (k, i) *exists in* $E(t)$.

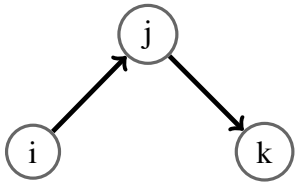


Figure 4.2: Triad t where (i, j) is overt and (j, k) covert. If t represents communication, then (i, j) enables message spread within the triad, whilst (j, k) does not..

Because overt and covert are defined upon triad structure, they can be applied to networks where message spread or contagion is not appropriate. For example, in food web networks an edge represents predator/prey trophic links (i.e if there is an edge from i to j, then j predates i). In this case, an overt edge represents a trophic link between prey and a primary predator: there is a secondary predator present in the triad who predates the primary predator. Since a reciprocated link would represent mutual predation (a very unlikely event) then a covert edge generally indicates a trophic link between prey (or primary predator) and a predator (primary or secondary) who, in terms of the triad, has no further predators. Therefore, overt and covert may enable us to establish which vertex is who in a food web. Alternatively, in airport networks an edge between two vertices i and j represents a flight from i to j. Therefore, in terms of a triad, an overt edge would represent a connecting flight whilst a covert edge a direct flight. In general, an overt edge is a relationship between two vertices which itself enables a further, adjacent relationship.

Figure 4.3 shows all possible connected triads with overt and covert edges indicated.

Note that there is high variability on the presence of overt and covert edges, between triads. This gives a new way to view all triads, noting the role that there edges play.

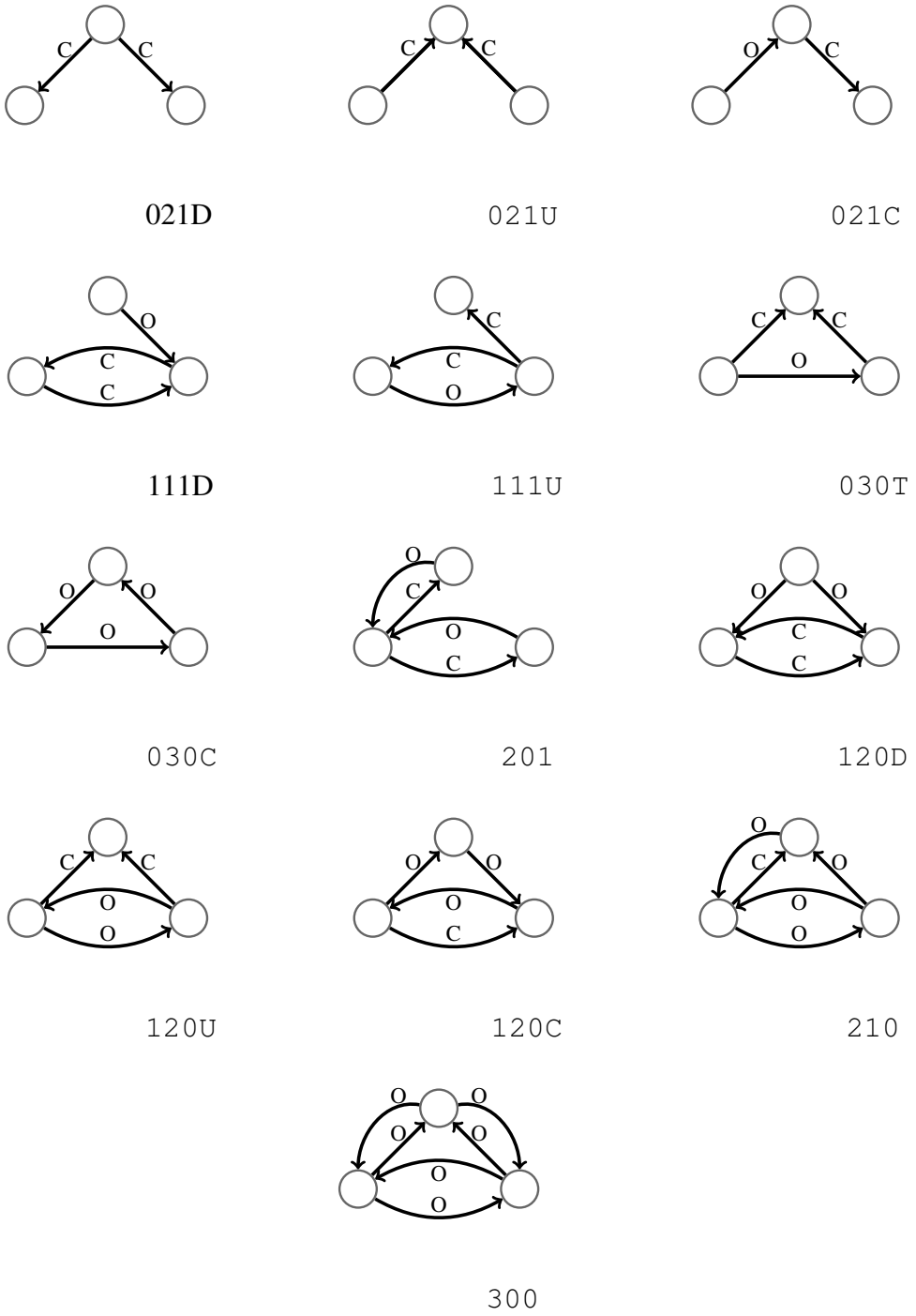


Figure 4.3: All connected triad types with overt and covert edges indicated by O and C respectively.

4.3 Relationship with Transitivity

In Chapter Two, we highlighted the clustering coefficient (Section 2.2) for networks [49] looking into definitions for transitive triads as defined by Wasserman and Faust [75] (Section 2.2). Transitive triads and overt edges in triads have some aspects in common. Transitive triads focuses on the existence of a third edge when the two other edges are present between three nodes. This is not dissimilar to our definition of an overt edge, which involves another edge in a triad being present. However in this section we point out why our classification for edges in triads as overt is related but distinct from transitive triads.

Recall the definition of a transitive triad [75] from Section 2.2:

Definition 28. *A triad (i, j, k) is a transitive triad if whenever there is a directed edge from i to j and from j to k, then there is a directed edge from i to k [75]. If no edge (i, j) or (j, k) exists then a triad is a vacuously transitive triad. Otherwise, it is an intransitive triad.*

Consider Example 4.4.

Example 4.3.1. *Suppose there exists three triads T_1, T_2 and T_3 on vertex set: $\{i, j, k\}$ and edge sets $E(T_1) = \{(i, j), (j, k)(i, k)\}$, $E(T_2) = \{(i, j), (j, k)\}$ and $E(T_3) = \{(i, j), (i, k)\}$ as shown in Figure 2.3. Then T_1 is a transitive triad, whereas T_2 is intransitive. T_3 is vacuously transitive as there is an edge from i to j and i to k, yet no edge from j to k.*

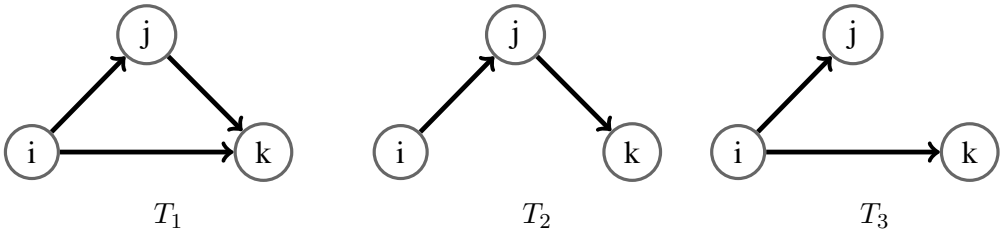


Figure 4.4: Example of three triads T_1, T_2 and T_3 where T_1 is transitive, T_2 is intransitve and T_3 is vacuously transitive.

Transitive triads are defined upon a structural relationship, though they have strong semantic interpretations dependent on the network type. For example, transitive triads are important in social networks because they highlight the relationship 'The friend of my friend is also my friend '. Similarly in food web networks, a transitive triad could indicate the relationship 'The predator of my predator is also my predator ', or in airport networks such triads could indicate the same end destination for a direct and connecting flight.

We are interested in showing that transitive triads and overt/covert edges are not two different interpretations of the same underlying structure. In Proposition 1 and Example 4.3.2 we show that the existence of an overt edge in a triad is necessary but not sufficient for the triad to be a transitive triad.

Proposition 1. *Suppose the triplet $t = (i, j, k)$ is a transitive triad. Then t necessarily contains an overt edge.*

Proof. If t is a transitive triad, then the edges $(i, j), (j, k), (i, k)$ are contained in t, as shown in Figure 4.5.

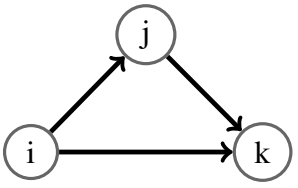


Figure 4.5: Transitive triad t, where t necessarily contains the overt edge (i, j).

Since (j, k) is contained in t, then (i, j) is overt. □

Note that in Figure 4.5 both (i, k) and (j, k) are covert since the edges (k, j) and (k, i) (which would make each edge (i, k) and (j, k), respectively, covert) are not contained in t. However, these edges could be contained within t without prevent t being a transitive triad, hence covert edges are not necessary for t to be a transitive triad.

The converse of Proposition 1 is not true: if a triad contains an overt edge, the triad is not necessarily a transitive triad. This is demonstrated by Example 4.3.2.

Example 4.3.2. *Suppose the triad $t = (i, j, k)$ contains the overt edge (i, j). Then (j, k) is also contained in t. Then t can be a transitive triad, an intransitive triad or vacuously transitive triad. Figure 4.6 shows one example where t is an intransitive triad.*

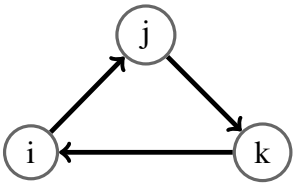


Figure 4.6: Triad t where the edge (i, j) is overt yet t is an intransitive triad.

Proposition 1 and Example 4.3.2 highlight a crucial difference between the clustering coefficient and overt edges, that it is necessary but not sufficient that a transitive triplet contain an overt edge.

4.4 Profiling Data Sets using Overt and Covert Centrality

In Section 4.2 we defined overt and covert as a categorisation for an edge based on its role within a triad. However, an edge can be present in multiple triads simultaneously. Further, an edge may play different roles across different triads. Consider Example 4.4.1:

Example 4.4.1. *Consider the simple graph G involving two induced triads: $t_1 = (x, u, v)$ and $t_2 = (u, v, w)$. Edge (u, v) is involved in both triads t_1 and t_2. (u, v) is covert in t_1 and (u, v) is overt in t_2, as shown in Figure 4.7.*

Centrality measures assigning importance to an edge or vertex for some particular reason. In this section we introduce Propositions 2, 3 and 4 that enable enumeration of the number of triads in which an edge is overt or covert, which we call the *overt centrality of an edge* and the *covert centrality of an edge*. The overt and covert edge

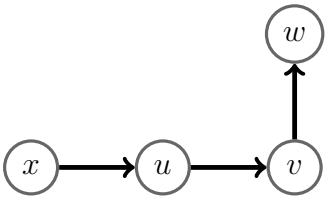


Figure 4.7: The simple graph G containing overlapping triads in which the same edge acts as both overt and covert.

counting approaches we introduce involve quick computation using the neighbourhood of nodes belonging to the edge of interest. This local approach means that in contrast to some alternative edge centrality metrics, computing overt and covert centrality is highly scalable. To understand the presence and implications of overt and covert centrality, we apply these metrics as edge weights to real world data sets. To exemplify this we determine the frequency distribution for overt/covert centrality across 34 test networks from different real world scenarios (Figures 4.9 to 4.41).

Assigning overt and covert centrality to edges has various applications based on what the edges in a network represent. Our motivating study is the dissemination of a message through sending a message via an edge from vertices i to j. In this example, the overt centrality of an edge represents the number of unintended recipients that a message can reach in one step. Therefore, the edges in a network which have a high overt centrality are important because they enable message spread within the network. Edges with a high covert centrality may be important because they supress spread. In alternative scenarios, overt and covert centrality have different semantics. The overt centrality of an edge is a measure of how important the relationship it represents is in terms of the number of adjacent relationships it enables. For example, in an airport network, the overt centrality of an edge represents the number of times a flight from i to j acts as a connecting flight to a destination one step away. Edges are assigned importance based on their ability to provide further passage to a greater number of destinations once the initial flight is taken. In a food web network, the overt centrality represents the number of times a prey's predator is itself the prey for another predator. Therefore, an edge may be interpreted as important because it enables further predator/prey links. Overt

and covert centrality assign edge importance based on the direction and existence of adjacent edges: in application they assign importance based on what is happening one step away. Therefore, it is natural to question what happens in terms of edge importance multiple steps away? We discuss this further in Chapter Five by applying overt and covert centrality to path problems in networks.

4.4.1 Computing Overt and Covert Centrality

Definition 29. *The overt centrality of an edge (u, v) is the number of induced triads in which (u, v) is overt. Similarly, the covert centrality of an edge (u, v) is the number of induced triads in which (u, v) is covert.*

As shown by example in Figure 4.7, an edge may be present in multiple triads and may take different roles in each (i.e., overt and covert). This means that edges can simultaneously hold non-zero overt and covert centrality.

Therefore computing overt and covert centrality for all edges is necessary. The following propositions present counting arguments to characterise this in general. We use d_v^- and d_v^+ to respectively denote the in-degree and out-degree of a node v. We let $N(v)$ denote the undirected neighbourhood of v, and $N(v)^-$ and $N(v)^+$ respectively denote the in and out-neighbourhood of v.

Proposition 2. *Let $(u, v)_p$ denote the number of induced connected triads in which $(u, v) \in E(G)$ is* present. *Then:*

$$(u, v)_p = |N(u) \cup N(v) - \{u, v\}| \tag{4.1}$$

Proof. Let $A = N(u) \cup N(v) - \{u, v\}$. Suppose $(u, v)_p > |A|$. Then there exists some $q \in V(G)$ such that (u, v, w) is a connected triad but $w \notin A$. Then w is not in the neighbourhood of u or v. So either $w \notin \{u, v\}$ or $w \in \{u, v\}$. In both cases (u, v, w) is not a connected triad which contradicts the assumptions for w. Hence $(u, v)_p \leq |A|$.

Alternatively suppose $(u,v)_p < |A|$. Then there exists some $w \in A$ such that (u,v,w) is not a connected triad. Since $w \in A$ then $w \neq u$ and $w \neq v$ and $w \in N(u) \cup N(v)$. Then this forces (u,v,w) to be a connected triad, contradicting the assumptions for w. Hence $(u,v)_p \geq |A|$.

Since $(u,v)_p \leq |A|$ and $(u,v)_p \geq |A|$ then $(u,v)_p = |A|$. □

Proposition 3. *Let $(u,v)_o$ denote the number of triads in which $(u,v) \in E(G)$ is an* overt *edge. Then:*

$$(u,v)_o = |N^+(v) - \{u\}| \tag{4.2}$$

Proof. Let $A = \{w|(v,q) \in E(G) \text{ and } w \neq u, w \neq v\}$. Suppose $(u,v) \in E(G)$ where $(v,u) \notin E(G)$ Then $|A| = d_v^+$.

Suppose $(u,v)_{\text{o}} > d_v^+$. Then $\exists\, w \in V(G)$ such that $(v,w) \in E(G)$ but $w \notin A$. Then $w = v$ or $w = u$. If $w = v$ then (u,v,w) is not a triad hence (u,v) is not overt. Hence $w = u$, but then $(v,u) \in E(G)$, contradicting the assumption that $(v,u) \notin E(G)$. Therefore, $(u,v)_{\text{o}} \leqslant d_v^+$.

Alternatively, suppose $(u,v)_{\text{o}} < d_v^+$. Then $\exists\, w \in A$ such that $(v,w) \notin E(G)$. This is impossible by definition of A. Hence $(u,v)_{\text{o}} \geqslant d_v^+$.

Since $(u,v)_{\text{o}} \leqslant d_v^+$ and $(u,v)_{\text{o}} \geqslant d_v^+$ then $(u,v)_{\text{o}} = d_v^+$. The proof for $(u,v)_{\text{o}} = d_v^+ - 1$ if $(v,u) \in E(G)$ follows similarly. □

The function OVERTCENTRALITY(G,u,v) in Algorithm 6 is used to compute the number of triads in which the edge (u,v) is overt, based on the previous counting arguments.

Proposition 4. *Let $(u,v)_c$ denote the number of connected triads in which $(u,v) \in E(G)$ is a* covert *edge. Then:*

$$(u,v)_c = |N(u)^+ \cup N(u)^- \cup N(v)^- - N(v)^+ - \{u,v\}| \tag{4.3}$$

Algorithm 6 OVERTCENTRALITY(G, u, v)

Input: Graph G**,** $(u, v) \in E(G)$
Output: $(u, v)_o$
1: **if** $(v, u) \in E(G)$ **then**
2: $\quad (u, v)_o = d_v^+ - 1$
3: **else**
4: $\quad (u, v)_o = d_v^+$
5: **return** $(u, v)_o$

Proof. Let $A = N(u)^+ \cup N(u)^- \cup N(v)^- - N(v)^+ - \{u, v\}$. Suppose that $(u, v)_{\mathrm{c}} \neq |A|$.

Firstly we assume that $(u, v)_{\mathrm{c}} > |A|$. Then there exists some $w \in V(G)$ such that (u, v, w) is a connected triad and (u, v) is a covert edge, but $w \notin A$. Hence $w \in N(v)^+$ or $w \in \{u, v\}$. If $w \in \{u, v\}$ then $w = u$ or $w = v$. Then (u, v, w) is not a connected triad, contradicting our definition of w. Hence $w \in N(v)^+$. This implies $(v, w) \in E(G)$ so (u, v) is an overt edge in (u, v, w), contradicting our definition of w again. Then no such w exists, hence $(u, v)_{\mathrm{c}} \leq |A|$.

Alternatively suppose $(u, v)_{\mathrm{c}} < |A|$. Then there exists some $w \in A$ such that (u, v) is an overt edge in the connected triad (u, v, w) or (u, v, w) not a connected triad. Since $w \in A$ then $w \in N(u)$ or $N(v)^-$ and $w \neq u$ and $w \neq v$ so (u, v, w) must be a connected triad. Then (u, v) must be overt in the connected triad (u, v, w). Then $(v, w) \in E(G)$ but since $w \in A$ then $w \notin N(v)^+$ so this contradicts our definition of w. Hence no such w exists so $(u, v)_{\mathrm{c}} \geq |A|$. Since $(u, v)_{\mathrm{c}} \leq |A|$ and $(u, v)_{\mathrm{c}} \geq |A|$ then $(u, v)_{\mathrm{c}} = |A|$. □

The function COVERTCENTRALITY(G, u, v) in Algorithm 7 is used to compute the number of triads in which the edge (u, v) is covert, based on the above counting argument.

Algorithm 7 COVERTCENTRALITY(u, v)

Input: Graph G**,** $(u, v) \in E(G)$**.**
Output: $(u, v)_c$
1: $(u, v)_c = |N(u)^+ \cup N(u)^- \cup N(v)^- - N(v)^+ - \{u, v\}|$
2: **return** $(u, v)_c$

Note that $(u, v)_p = (u, v)_{\text{o}} + (u, v)_{\text{c}}$. In other words, the number of induced triads in which an edge participates is the sum of its overt centrality and covert centrality.

An edge (u, v) having high overt centrality indicates that in a large number of triads, edge (u, v) contributes to a path from u to the third node w. Conversely, an edge (u, v) having high covert centrality indicates that in a large number of triads, edge (u, v) is not involved in a path from u to the third node w.

To compute overt and covert centrality of a whole graph, we apply function NETWORK-CENTRALITY(G) in Algorithm 8, using Algorithms 6 and 7 from section 4.4.1.

Algorithm 8 NETWORKCENTRALITY(G)

Input: Graph G
Output: Weighted graph.
1: **for** $(u, v) \in E(G)$: **do**
2: **if** $u \neq v$ **then**
3: $(u, v)_o =$ OVERTCENTRALITY(u, v) ▷ Add edge weights
4: $(u, v)_c =$ COVERTCENTRALITY(u, v)
5: **return** G

4.4.2 Results

To explore overt and covert centrality, we apply the techniques from Section 4.4.1 using the 34 real-world data sets from Section 2.5. We find the overt centrality for each edge in the network and plot the frequency of occurrence of each centrality value. We repeat this for covert and then for every network. The frequency distributions of the overt and covert centrality are presented in Figures 4.9 to 4.42. The results are verified because Algorithms 6 and 7 are constructed using Propositions 3 and 4, which we conclusively prove to be true for every case in Section 4.4.1.

Interestingly we see that in many cases, the resulting frequency distributions are effective in associating networks originating from same context or domain. For example, the airport flights data sets [42, 67] have long tail distributions with a relatively slow drop off as compared to the peer-to-peer Internet data sets [47]. Other classes of network, such as

the Organise networks [63] exhibit distinctive increasing profiles in frequency of overt centrality. Some classes of network also exhibit greater weighting for one particular type of centrality, such as the electrical circuit scenarios [5] where covert edges show greater prevalence. In general, we are able to classify the networks into three categories based on the shape of their overt and covert centrality profiles. There are those networks, such as Prison Inmate [5], where the overt centrality and covert centrality profiles are long tailed (i.e there are many edges with a proportionately very low overt/covert centrality, and a rapidly decreasing number of edges with proportionately higher overt/covert centralities). Within this classification, there are those networks where the difference between the number of edges with low overt/covert centrality and the number of edges with higher overt/covert centrality is less extreme (i.e the 'drop off 'in the long-tailed distribution is less pronounced), such as US Airports [42] or Political Blogs [2], or those where the drop off is more pronounced, such as Prison Inmate [5] or p2p-gnutella04 [47]. The second classification of network are those where the overt centrality profile is long-tailed, yet the covert centrality profiles are not. This categorisation includes food web networks such as Mangwet [74] , or neural networks such as Rattus Norvegicus [58]. Finally, there are networks whose overt and covert centrality profiles both form a bell curve, such as the food web networks Ythan [1] or St.Mark's Seagrass [14], or any of the organise networks (with the exception of Eva [41]).

To understand what drives these classifications, we observe how the latent network features may affect overt or covert centrality of an edge. In Chapter Two, Table 2.4, we offer a number of measures to give information about each of the data sets, such as network density or average betweenness centrality.

There are three network features in Table 2.4 which may affect the possible induced triads which can occur in a network: namely density, clustering coefficient and reciprocity. For each of these measures, we observe Figure 4.3 and count the total number of overt and covert edges amongst the triads that exhibit features relevant to the measure. The restrictions on the possible induced triads occurring may affect the likelihood an

edge behaves as overt or covert within a triad, and therefore its overt or covert centrality. However: it is important to note this isn't the whole picture. Overt and covert centrality are also affected by the way the triads in a network overlap, thus an edge may have a higher overt or covert centrality than another simply because it is contained in more triads. Thus, if measures such as clustering coefficient are skewed by an area of high clustering in the network, this could mean many closed triplets overlap in one area and therefore contribute to an area of edges with high overt and covert centrality simply due to the abundance of overlapping triads in this area. Nevertheless, it is useful to consider the triads in isolation as they partially explain the proportion of overt to covert centralities in a network.

Clustering coefficient relates to the number of closed triplets (i.e triplets where there is an edge from each vertex in the triad to every other vertex) in the undirected equivalent of a network (see Section 2.2). Amongst the triads in Figure 4.3, we observe that `021D`, `021U`, `021C`, `111D`, `111U` and `201` are open triplets, whilst `030T`, `030C`, `120D`, `120U`, `120C`, `210` and `300` are closed triplets. Amongst the seven closed triplets there are a total of eight covert edges and 21 overt edges, whilst amongst the six open triplets there are a total of three overt edges and nine covert edges. Thus we expect that closed triplets are more likely to induce overt than covert edges, whilst open triplets the opposite. Therefore, we predict a network which exhibits a greater volume of clustering will also exhibit a greater proportion of overt edges to covert edges; and visa versa.

An edge (u, v) is reciprocated if there also exists the edge (v, u). Amongst the 13 triads in Figure 4.3, eight exhibit at least one reciprocated edge, they are: `111D`, `111U`, `201`, `120D`, `120U`, `120C`, `210` and `300`. Amongst these, there are a total of 21 overt edges and 12 covert edges. Five triads contain no reciprocated edges, namely: `021D`, `021U`, `021C`, `030T` and `030C`. Amongst these there are a total of five overt edges (three of which are contained in the triad `030C`) and seven covert edges. Thus, we predict a network which exhibits a greater volume of reciprocity will also

exhibit a greater proportion of overt edges to covert edges; and visa versa.

Finally, density is the proportion of edges in a network which exist. This may affect the number of edges present in triads (for example, in a network of density 1, then the only possible triads are type `300`). Amongst the triads on two edges (`021D`, `021U`, `021C`) there are five total covert edges and one overt edge. Amongst triads on three edges (`111D`, `111U`, `030T`, `030C`) there are six total covert and overt edges, though three of the overt edges are derived from `030C`. There are nine overt and seven covert edges across the triads on four edges (`201`, `120D`, `120U`, `120C`), four overt and one covert edge in `210` (the only triad on five edges), and all six edges in `300` are overt. Thus, we predict a network which exhibits a greater density will also exhibit a greater proportion of overt edges to covert edges; and visa versa.

We have partitioned the set of thirteen triads according to isolated features they exhibit, but we can combine features to partition the set further. For example, amongst closed triplets, those which exhibit reciprocity (i.e `120D`, `120U`, `120C`, `210` and `300`) contain a total of 17 overt edges and six covert edges. Incidentally, these triads are the most dense amongst the thirteen. For example, `300` is simultaneously the most dense triad and contains the greatest number of reciprocated edges amongst the 13 triads.

Aside from reciprocity, density and clustering coefficient, which we have discussed as features which may affect the possible triads induced by a network, there are other network features from Table 2.4 that relate to the possible number of triads which overlap on one edge. If more triads overlap on one edge, then this edge will have a greater total overt and covert centrality because it is present in more triads. For example, degree centrality counts the number of edges incident with a vertex. All edges incident with a single vertex are adjacent to one another, and therefore form triads. Thus, edges incident with a vertex with a higher degree centrality will be contained in a greater number of overlapping triads, and therefore will have a greater total overt and covert centrality. Similarly, betweenness centrality assesses the number of shortest paths which overlap on an edge. Every pair of adjacent edges of a path form a triad, thus if more

paths overlap on an edge then it is likely a greater volume of triads overlap on this edge, and therefore the edge may have a greater total overt and covert centrality.

We predict that the higher the average reciprocity, clustering coefficient and density of a network, the less long-tailed the overt and covert centrality profiles become, where the covert centrality profile more rapidly becomes bell-shaped (rather than long-tailed) as these values increase than the overt centrality profile. We focus on reciprocity, clustering coefficient and density because, as observed in Table 2.4, the average mean degree centrality and betweenness centrality are particularly low amongst all networks.. Certainly, from Figures 4.9 to 4.42 our predictions appear true. For example, consider the Organise networks (Figure 4.29 and 4.30). From Section 2.5 we know that Organise networks (with the exception of Eva [41]) have the highest density amongst all the data sets. They also exhibit relatively high levels of clustering and high reciprocity. Overt centrality profiles tend to be more bell-shaped than covert centrality profiles, with a greater proportion of edges with a higher overt centrality. This could mean that there is such a high abundance of overt edges due to the density, reciprocity and clustering exhibited in these networks, that in fact there are not many covert edges present. Contrast this with the Internet data sets (Figure 4.19 and Figure 4.20): where there is a very high proportion of edges with a low (or zero) overt centrality. Amongst the data sets, from Section 2.5 we know that Internet networks have the lowest densities by a significant margin, as well as very little clustering and no reciprocity. There are still many edges with a low overt centrality, telling us that there are many edges contained in a low number of triads in which they behave as overt. This corresponds with other sparse networks, such as the organise network Eva [41]. Further, since Eva [41] is an organise network yet its profiles do not fit the trend of other organise networks and are instead more similar to the profiles exhibited by internet networks, we believe that overt and covert centrality profiles are restricted by network structure , and not the particular domain or context.

Since Internet and Organise networks are so different across all features, it is difficult

to unpick which feature is influencing the overt and covert centrality profiles the most, though it is likely it is a combination of factors. Thus, we look to compare networks where most features are of similar value, but differ in one feature.

US airports [42] exhibit relatively low density but a surprisingly high volume of reciprocity, with relatively high clustering. Therefore, they are a good candidate to compare with organise networks. US Airports [42] and Cross Parker Consulting [63] exhibit the similar values for clustering and reciprocity, yet US Airports [42] has a low density (0.011) whilst Cross Parker Consulting [63] has a much higher density (0.28). Certainly, overt and covert centrality profiles for US Airports [42] (Figure 4.9) are long-tailed, whilst Cross Parker consulting exhibit bell-shaped profiles (Figure 4.29). Therefore, density may play a significant role in the shape of the overt and covert centrality profiles. In contrast, when comparing the neural network Mouse Retina [58] with the food web network Mangwet [74] (which have similar levels of clustering and reciprocity but different densities) we observe the centrality profiles look similar. In fact, the denser network Mangwet [74] (with a density of 0.16) exhibits a more pronounced peak in covert centrality than in the sparser network Mouse Retina [58] (with a density of 0.079).

Our data sets do not offer two networks where density and reciprocity are similar yet clustering is very different. Instead, to understand both reciprocity and clustering coefficient we can compare networks where density is similar but both reciprocity and clustering are different. Food web networks are useful because they have similar densities to many other networks amongst our data sets, yet contain very little reciprocity due to the rarity of mutual predation. They do also exhibit some clustering, much higher than clustering values found for Internet or Regulatory networks. In contrast, we know Airport networks exhibit very high levels of reciprocity, though they do also exhibit higher levels of clustering than food web networks. Accordingly, we compare the airport network US Airports [42] with the food web network Grassland [14] due to their similar densities (0.011 and 0.018 respectively). Whilst both overt centrality

profiles for US Airports [42] and Grassland [14] are long-tailed, the covert centrality profile for Grassland [14] is more bell shaped than US Airports [42], which also has a long-tailed covert centrality distribution. This could be because US Airports has such high levels of reciprocity the types of triad induced in airport networks do not contain many covert edges, and indeed, the overt centrality profile for US Airports [42] exhibits a less extreme drop-off than the overt centrality profile for Grassland [14].

However, there are other networks that do not behave as we have predicted. For example, consider the Social network Prison Inmate [5]. Prison Inmate exhibits similar density and clustering to Food web networks (particularly Ythan [1]), with a much higher level of reciprocity (0.44 in Prison Inmate CITE, 0 in Ytahan CITE). Similarly to comparisons of Grassland with US Airports [42], whilst Ythan [1] has a long-tailed overt centrality with relatively slow drop off and a bell shaped covert centrality profile, Prison Inmate [5] has a long-tailed profile for both overt and covert centrality with a fast drop off, despite exhibiting greater reciprocity. Indeed, the overt and covert centrality profiles for Prison Inmate CITE oappear similar to Internet networks, which we have discussed as having very low values in density, reciprocity and clustering coefficient. Though surprising, other social networks such as Bitcoin Alpha [47] and UCIrvine [63] also have similar profiles. This could signify that, contrary to our predictions, perhaps clustering and reciprocity are not so important to determining the overt and covert centralities of edges. Perhaps these results are caused by some other latent feature present in social networks, such as a hub-and-spoke structure, which are not featured in food web networks, which are hierarchical.

A network is ‘hub-and-spoke’if it contains many vertices with relatively low degrees (the ‘spokes’) centred around a vertex with a relatively high degree (the ‘hubs’). Spoke vertices may attach to each other, but they are more likely to attach to the hub vertex (i.e they exhibit ‘preferential attachment ’). Further, if there are several hub vertices, the spoke vertices tend to cluster around one particular hub vertex, not attaching to other hub vertices. Many complex networks are hub-and-spoke, characterised by long-tailed

or scale free degree distributions. It is possible that if a network exhibits hub-and-spoke structure this may affect the overt and covert centrality of edges and therefore the centrality profiles due to the number of triads that can possibly overlap on an edge. Consider Example 4.4.2.

Example 4.4.2. *Consider the very simple hub-and-spoke graph G, as shown in Figure 4.8. In G, each u_i and v_i vertex has an out-degree of one, whilst u and v both have an in-degree of three.*

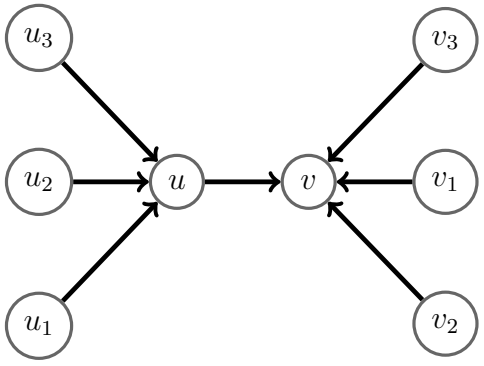


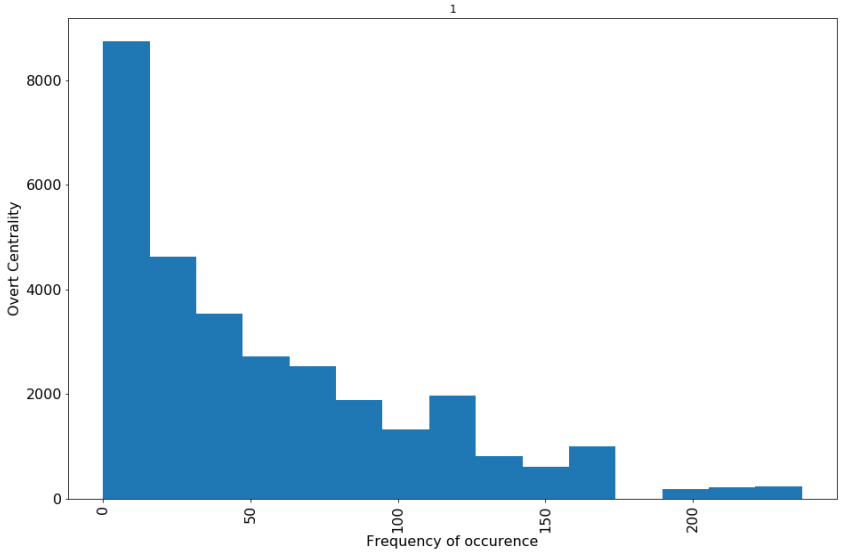
Figure 4.8:

Each edge (u_i, u) forms a triad with the edge (u, v), and a triad with each edge (u_j, u) where $j \neq i$. Similarly, each edge (v_i, v) forms a triad with the edge (u, v), and a triad with each edge (v_j, v) where $j \neq i$. In other words, they can only form triads with other edges adjacent to the same hub node, and therefore the number of triads containing the edge is entirely dependent on the degree of the hub vertex. In contrast, (u, v) forms triads with each edge (u_i, u) and triads with each edge (v, v_i). Therefore the number of triads the edge between two hub vertices is contained in is dependent on the degree of both hub vertices. Since an edge can only act as overt or covert in triads it is contained in, then an edge contained in more triads has a higher combined overt and covert centrality than edges contained in fewer triads, and therefore edges connecting hub nodes have a higher combined overt and covert centrality than edges connecting a hub node with a spoke node.

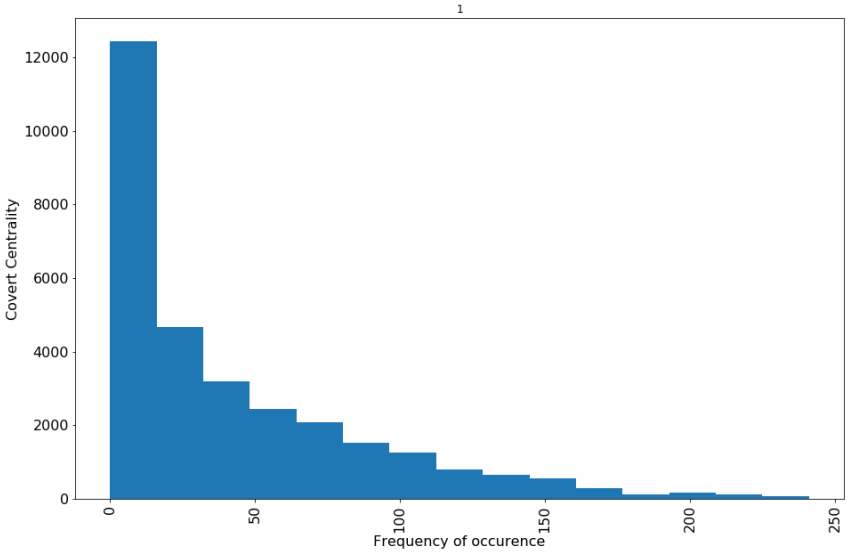
In Example 4.4.2, we show that in a network that is hub-and-spoke, edges between hub

nodes may have a higher total overt and covert centrality (dependent on the degree of both hub nodes) than edges between spoke and hub nodes (dependent on the degree of the hub node). Thus, in terms of overt and covert centrality profiles, this would correspond to some edges having a very high centrality, and many edges having a much lower centrality: which could result in long tailed centrality profiles. This could explain why social networks exhibit long tailed overt and covert centrality profiles despite having higher levels of reciprocity and clustering than other comparable networks.

These results demonstrate that overt centrality and covert centrality distinguish different networks by the role that their edges play within triads. This offers the potential to characterise networks in a new way, which is aligned to triadic census, as commonly used for characterising complex networks. Note that this is complementary to the commonly used technique of motif analysis, however in our case, comparison is based on the role edges play within induced triads, rather than the types of induced triads that are over or under represented.

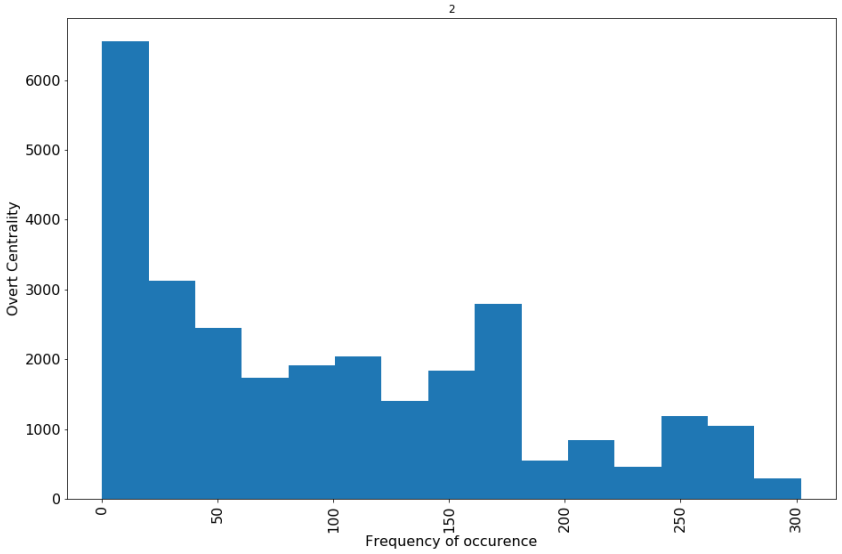


(a) Overt Centrality Frequency Distribution

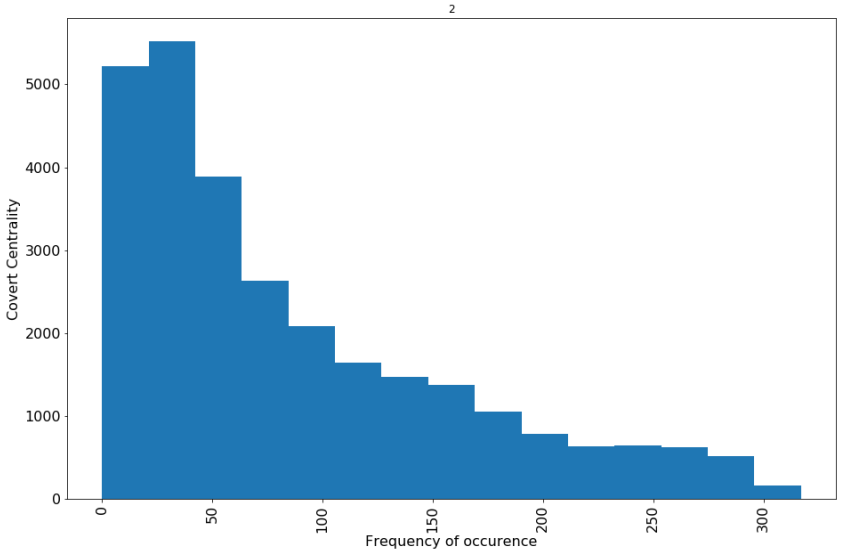


(b) Covert Centrality Frequency Distribution

Figure 4.9: US Airports [42]

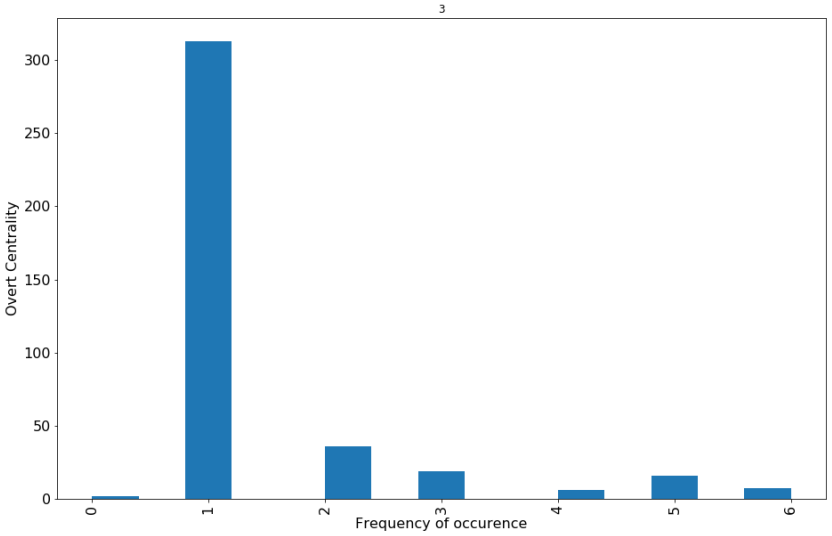


(a) Overt Centrality Frequency Distribution

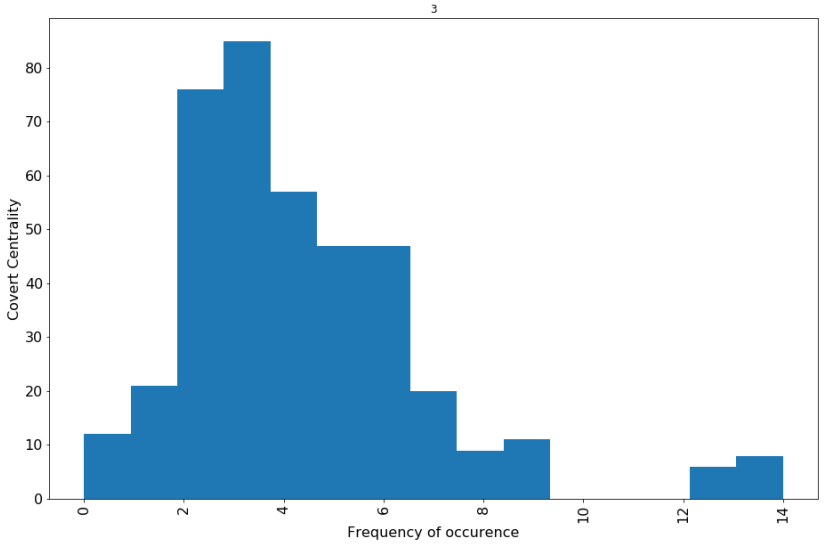


(b) Covert Centrality Frequency Distribution

Figure 4.10: Open Flights [67]

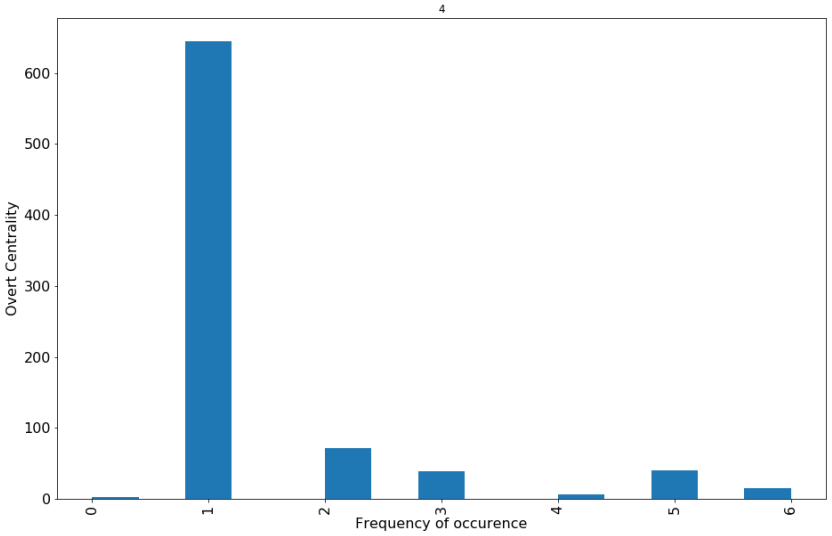


(a) Overt Centrality Frequency Distribution Comparison

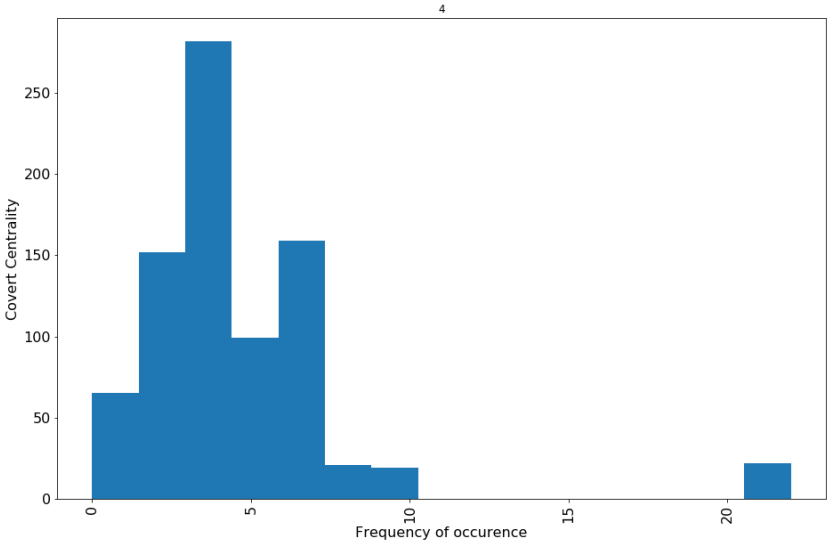


(b) Covert Centrality Frequency Distribution

Figure 4.11: s420 [5]

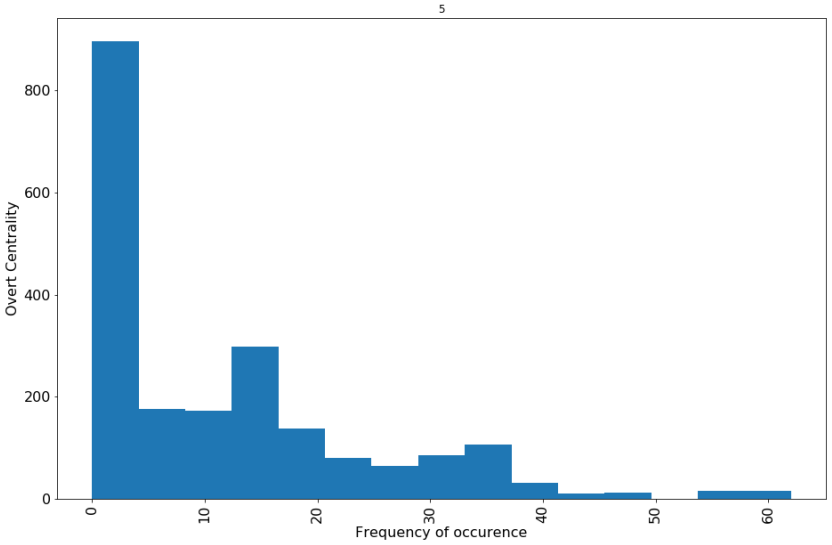


(a) Overt Centrality Frequency Distribution Comparison

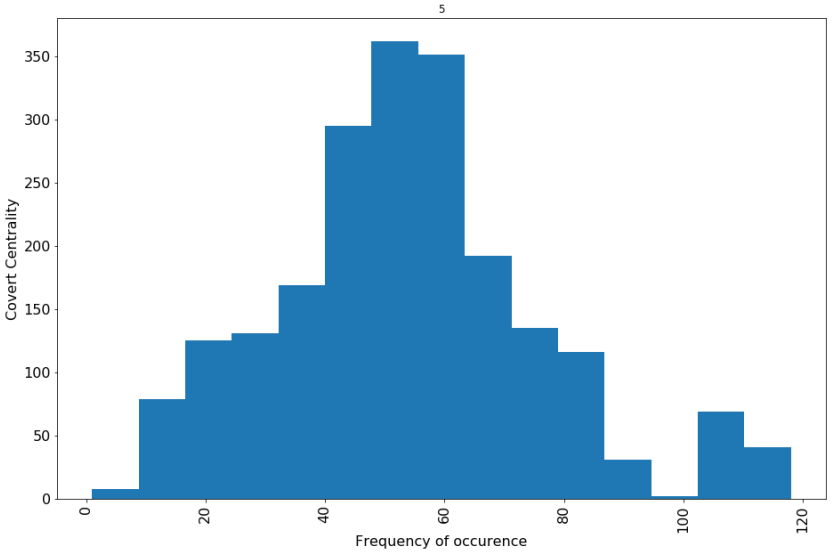


(b) Covert Centrality Frequency Distribution

Figure 4.12: s838 [5]

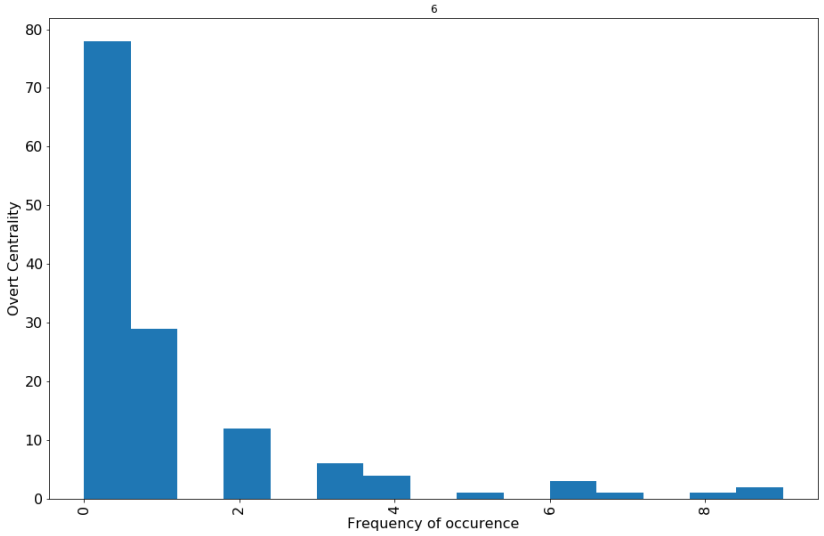


(a) Overt Centrality Frequency Distribution

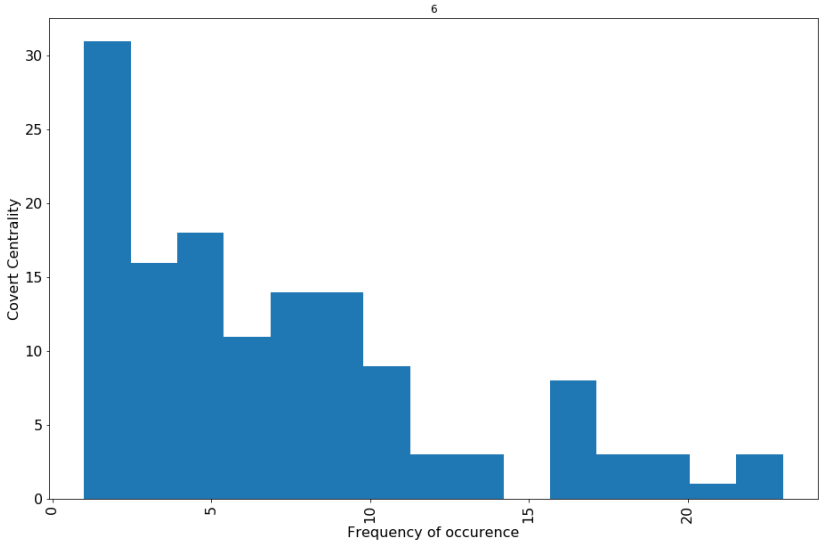


(b) Covert Centrality Frequency Distribution

Figure 4.13: Mangwet [74]

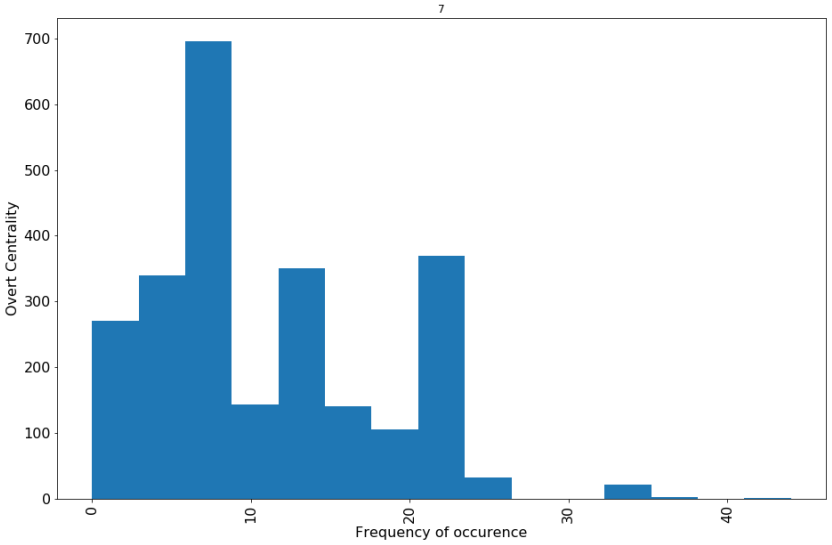


(a) Overt Centrality Frequency Distribution Comparison

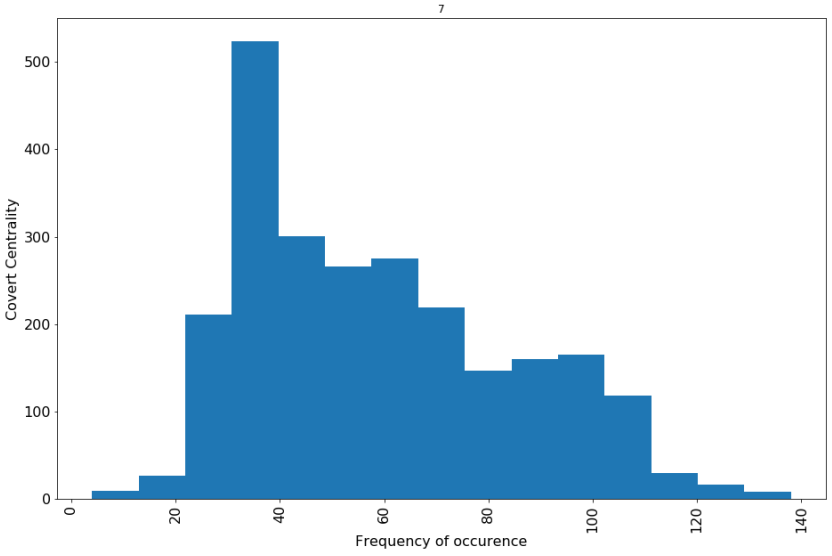


(b) Covert Centrality Frequency Distribution

Figure 4.14: Baywet [74]

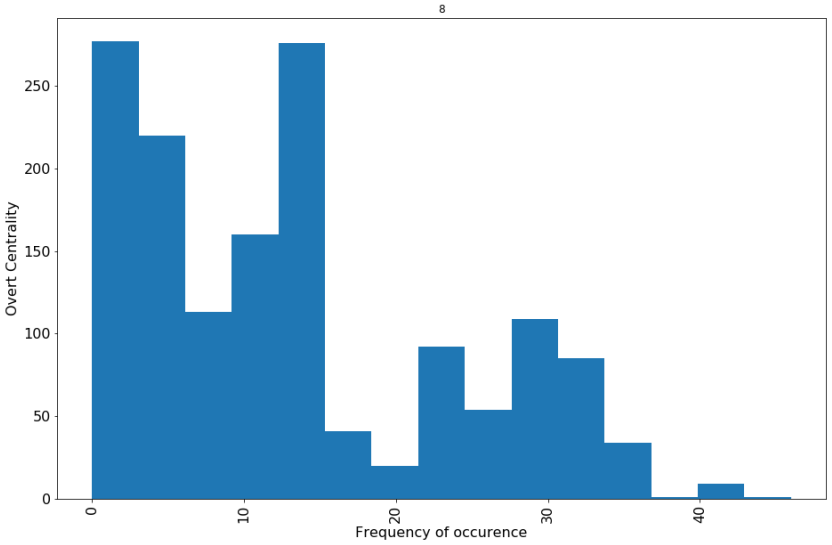


(a) Overt Centrality Frequency Distribution Comparison

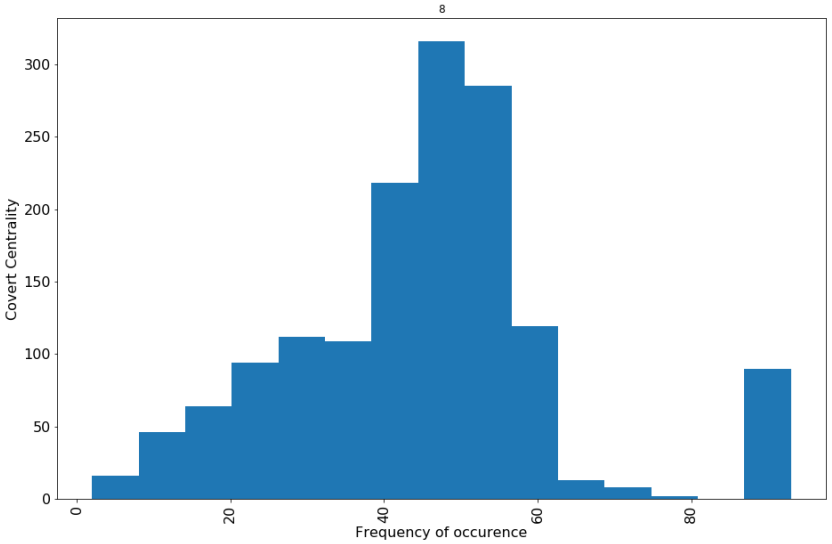


(b) Covert Centrality Frequency Distribution

Figure 4.15: Little Rock Lake [42]

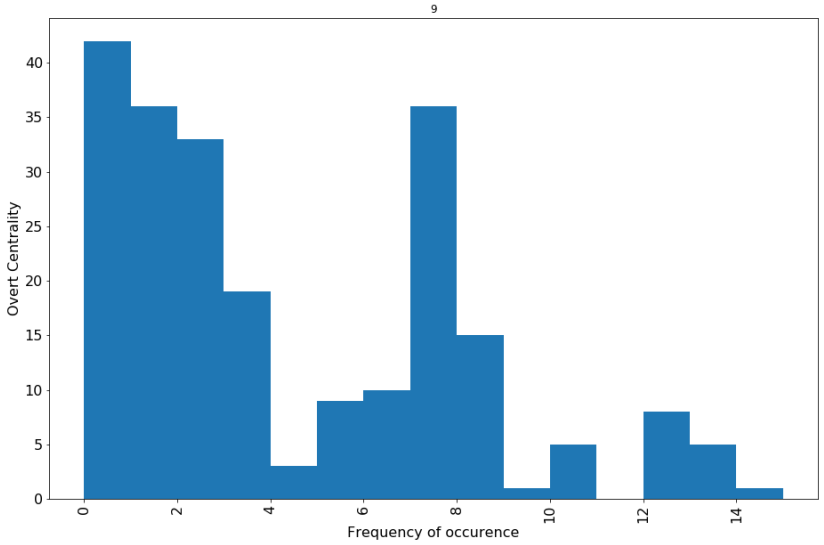


(a) Overt Centrality Frequency Distribution Comparison

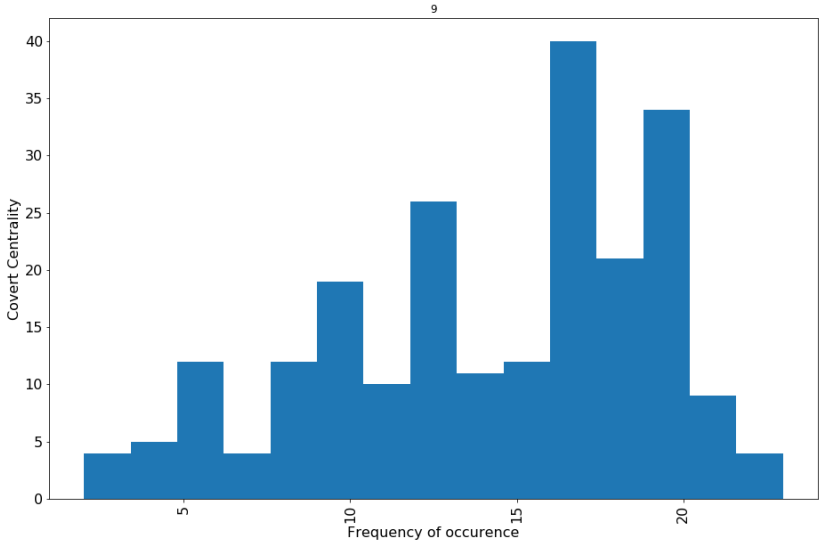


(b) Covert Centrality Frequency Distribution

Figure 4.16: Ythan [1]

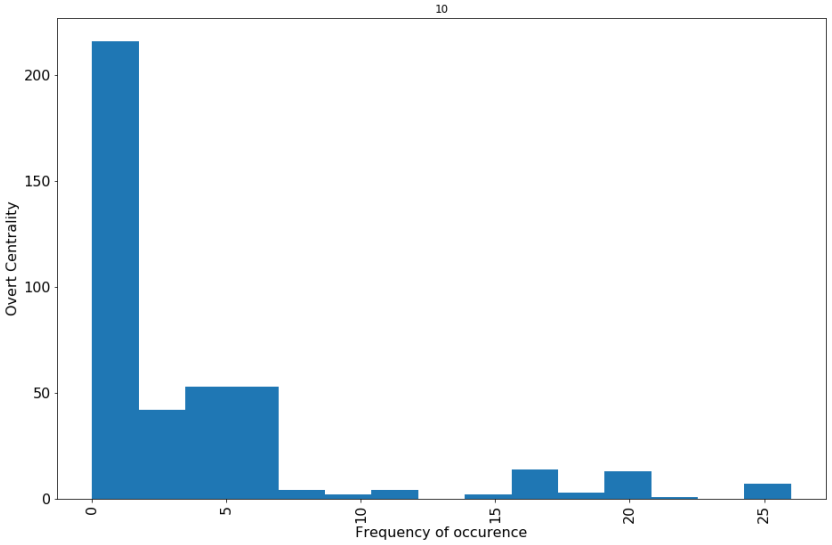


(a) Overt Centrality Frequency Distribution Comparison

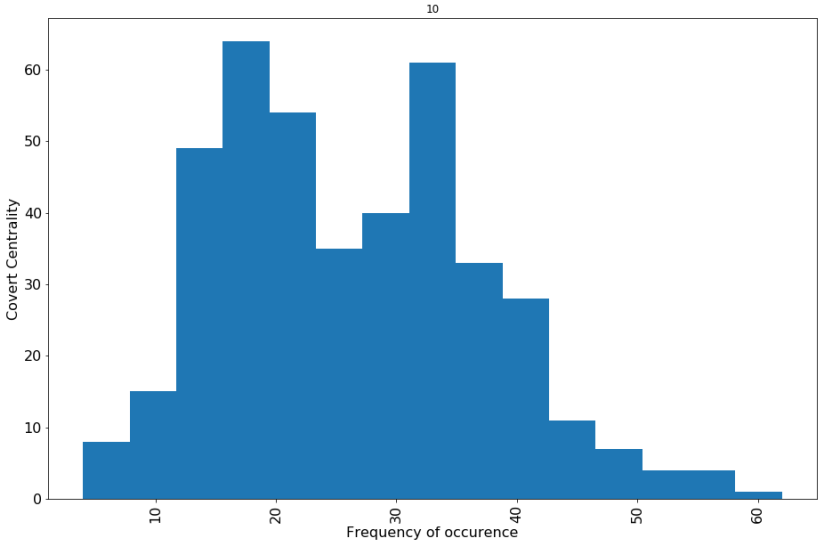


(b) Covert Centrality Frequency Distribution

Figure 4.17: St. Marks Seagrass [14]

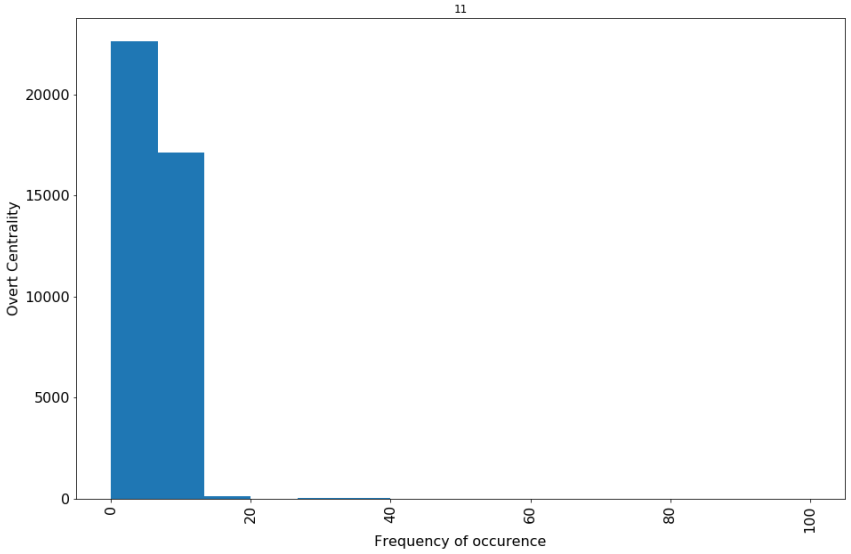


(a) Overt Centrality Frequency Distribution Comparison

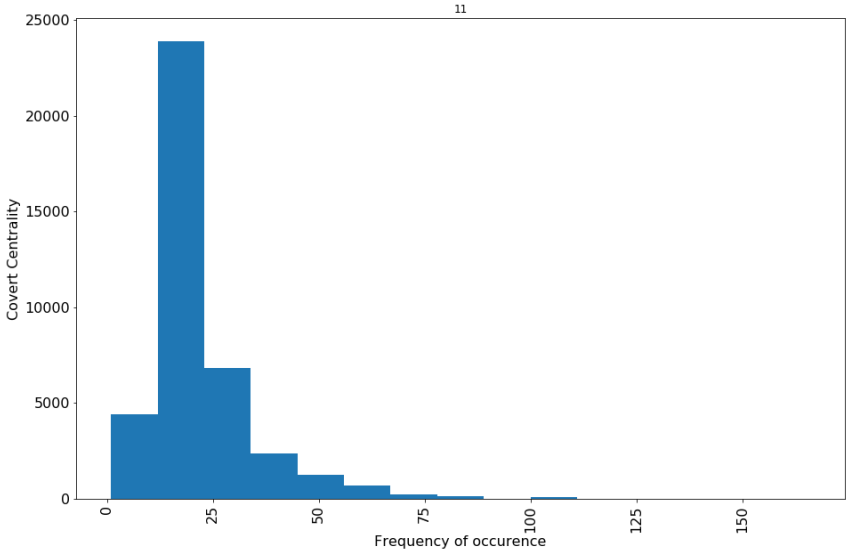


(b) Covert Centrality Frequency Distribution

Figure 4.18: Grassland [14]

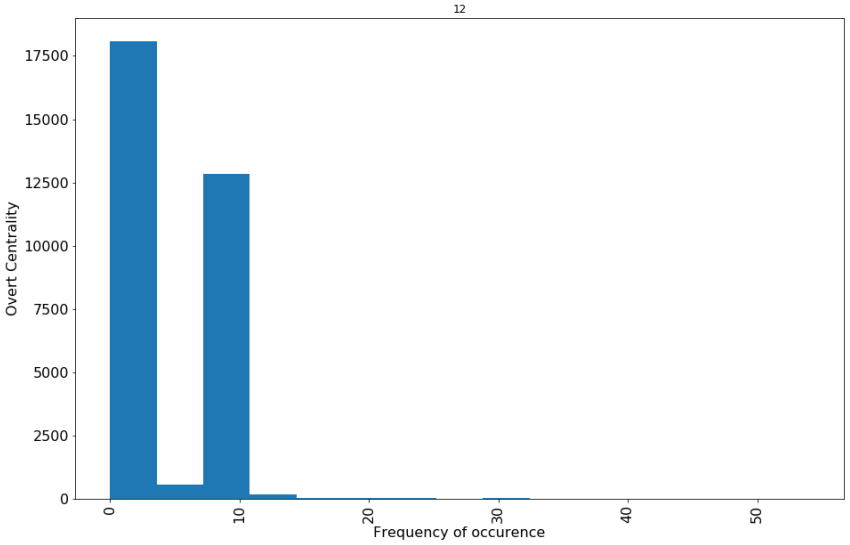


(a) Overt Centrality Frequency Distribution

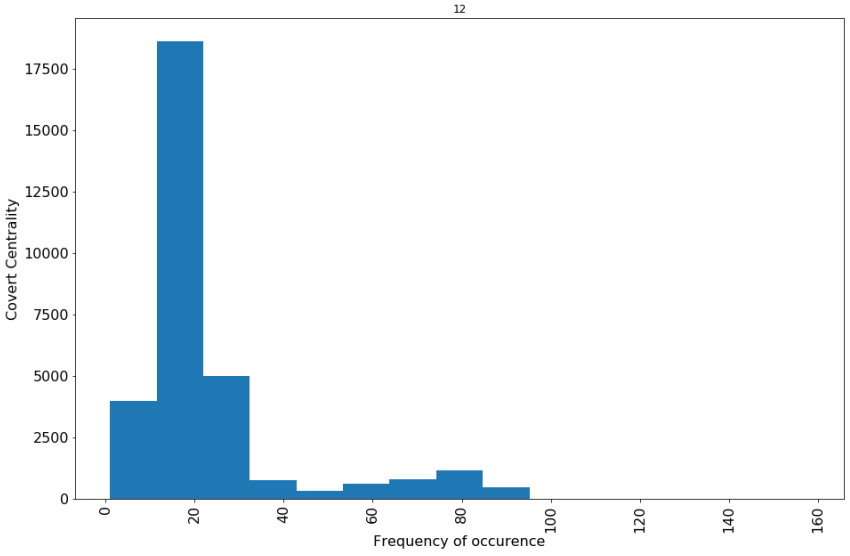


(b) Covert Centrality Frequency Distribution

Figure 4.19: p2p-gnutella04 [47]

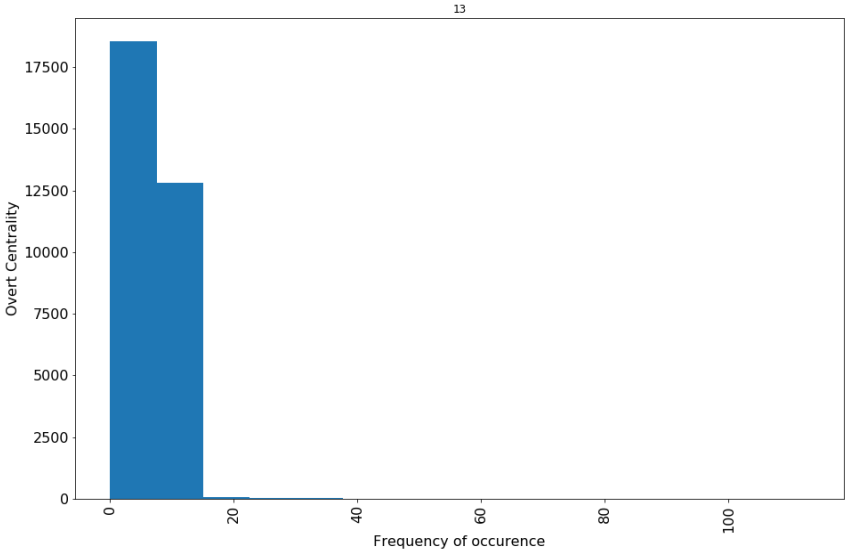


(a) Overt Centrality Frequency Distribution

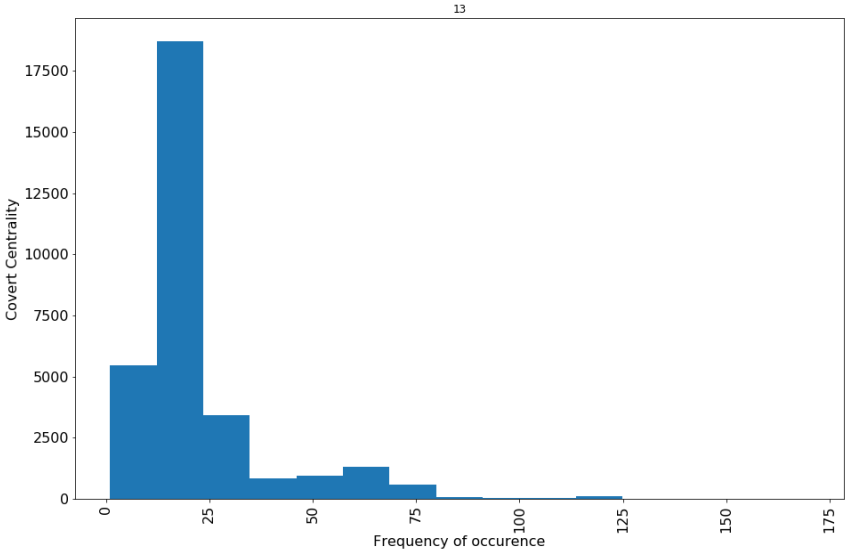


(b) Covert Centrality Frequency Distribution

Figure 4.20: p2p-gnutella05 [47]

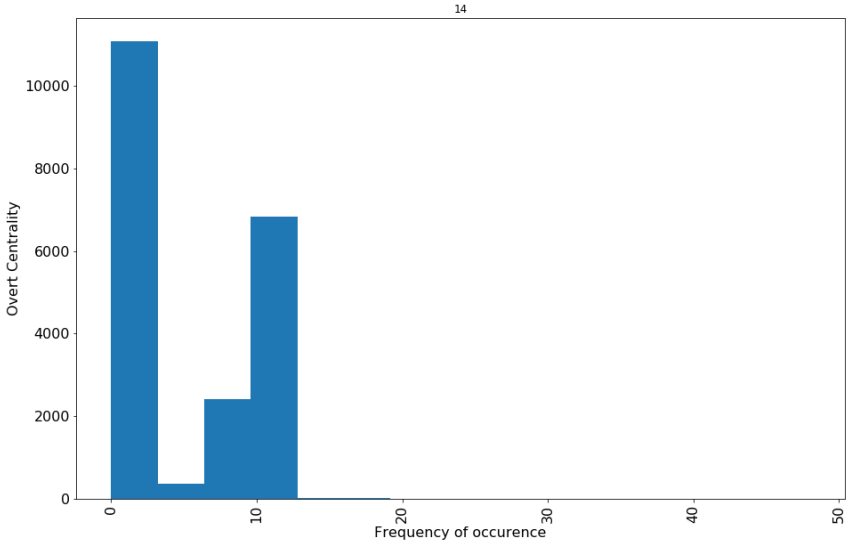


(a) Overt Centrality Frequency Distribution Comparison

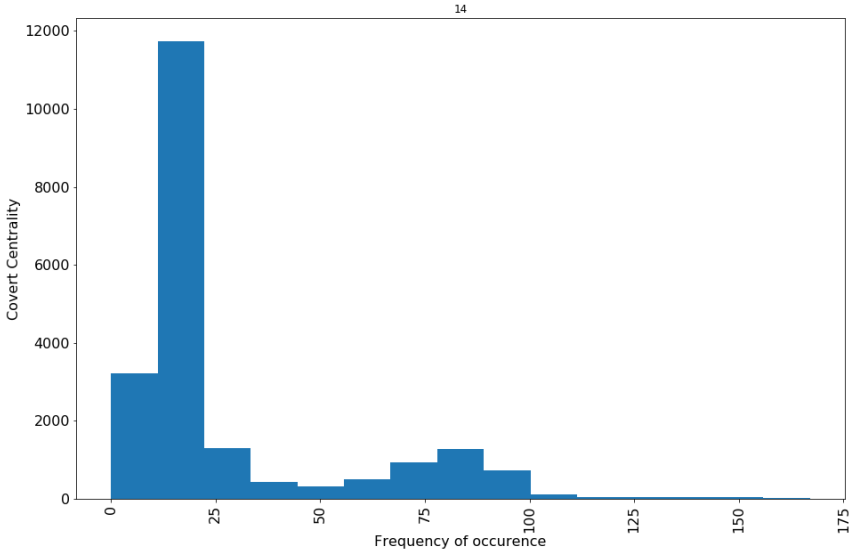


(b) Covert Centrality Frequency Distribution

Figure 4.21: p2p-gnutella06 [47]

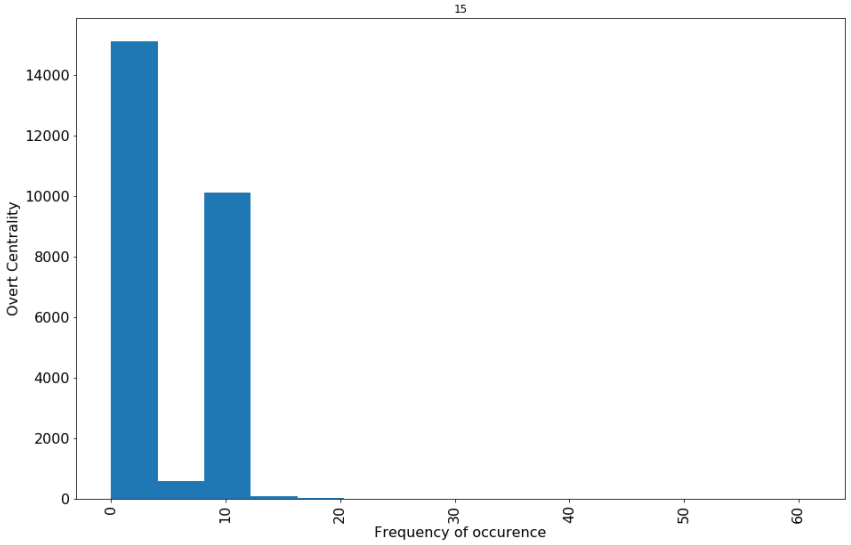


(a) Overt Centrality Frequency Distribution Comparison

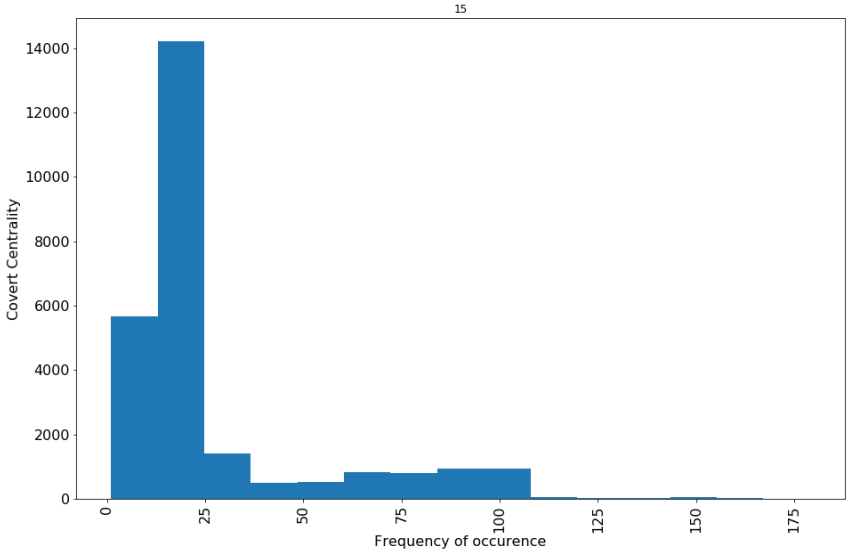


(b) Covert Centrality Frequency Distribution

Figure 4.22: p2p-gnutella08 [47]

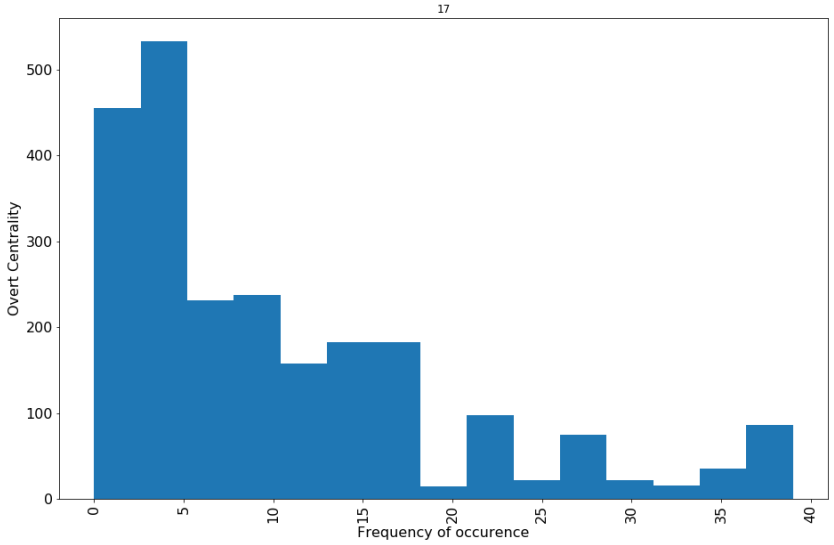


(a) Overt Centrality Frequency Distribution

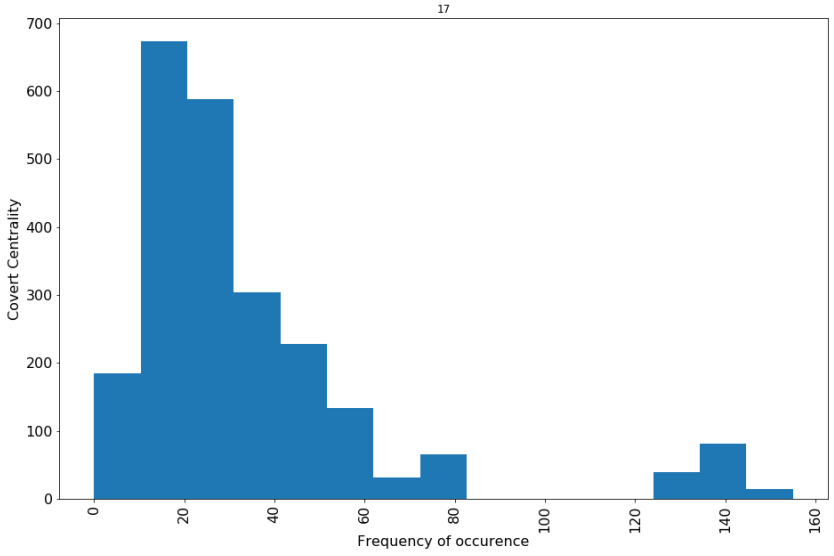


(b) Covert Centrality Frequency Distribution

Figure 4.23: p2p-gnutella09 [47]

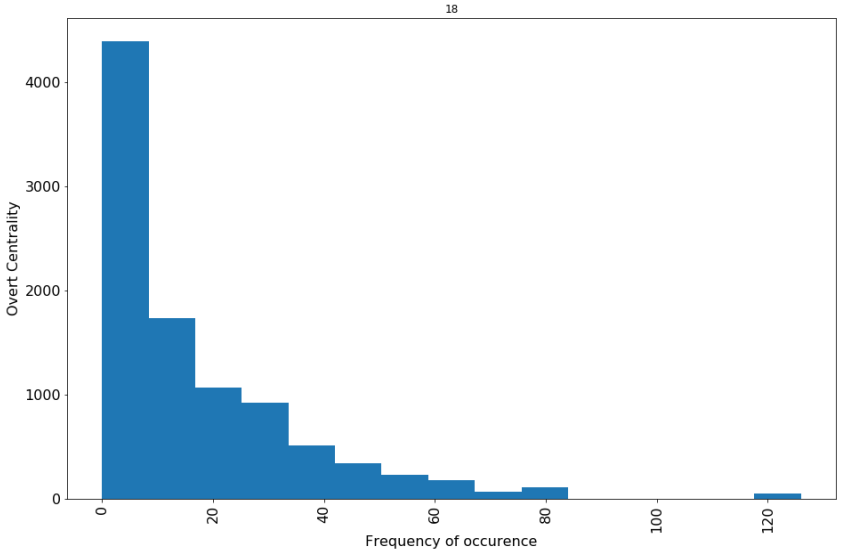


(a) Overt Centrality Frequency Distribution

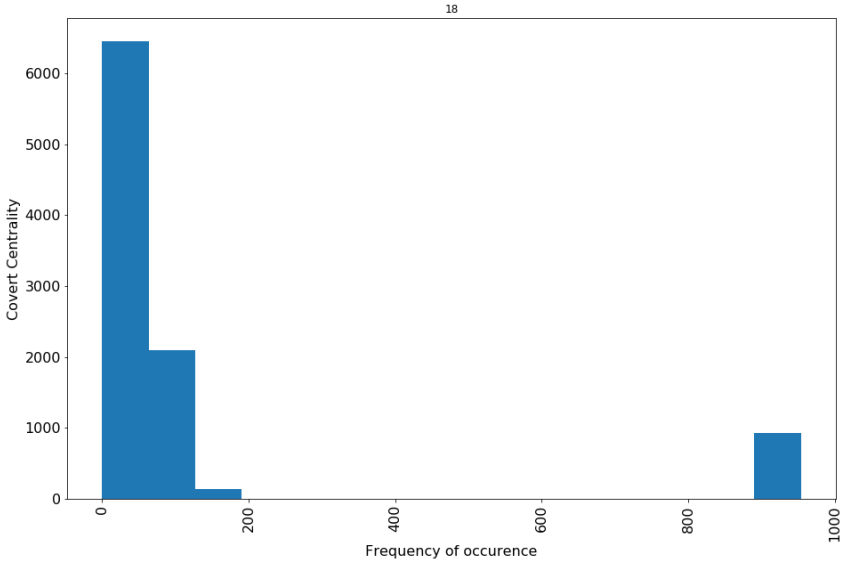


(b) Covert Centrality Frequency Distribution

Figure 4.24: C. Elegans [40]

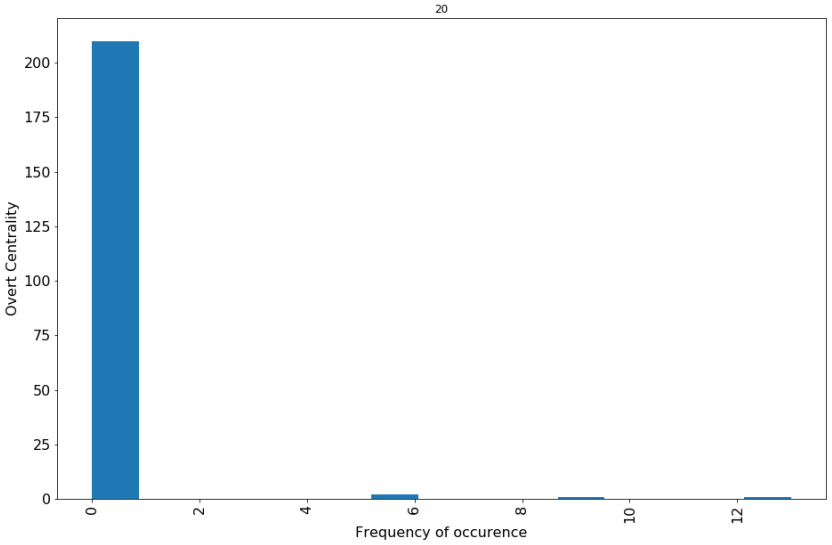


(a) Overt Centrality Frequency Distribution Comparison

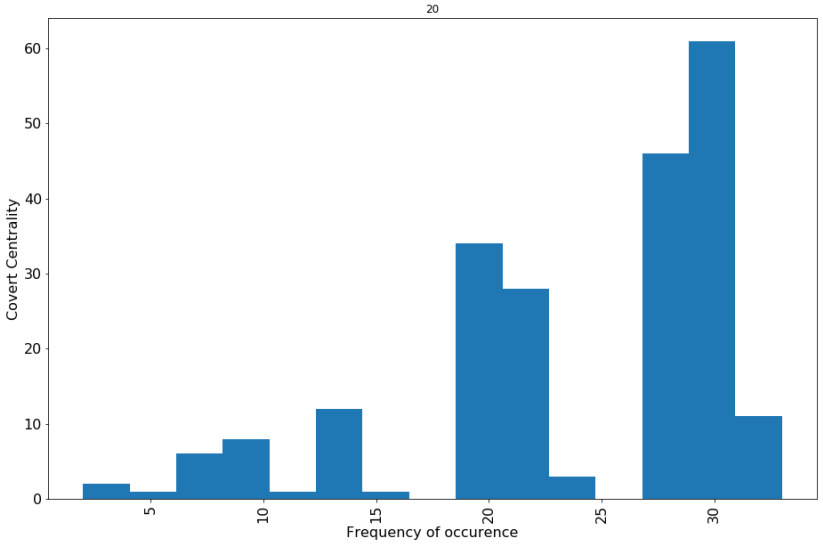


(b) Covert Centrality Frequency Distribution

Figure 4.25: Drosophila Medilla 1 [58]

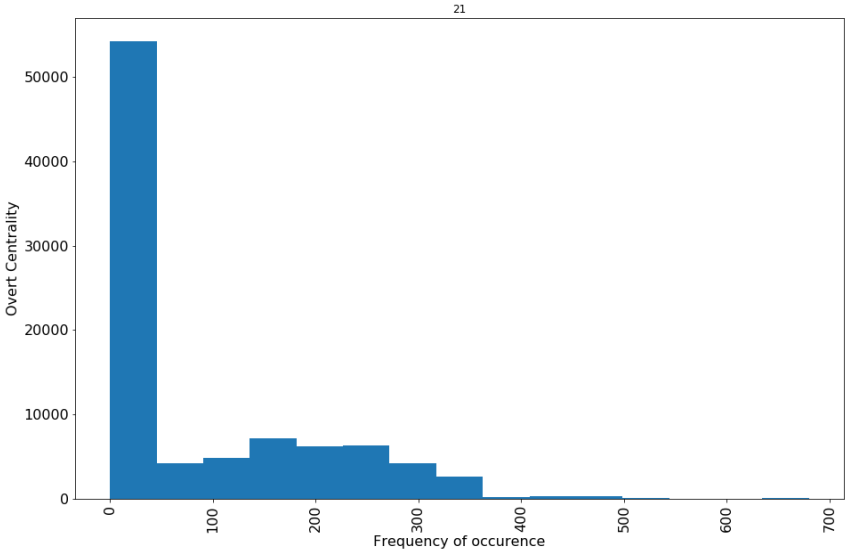


(a) Overt Centrality Frequency Distribution

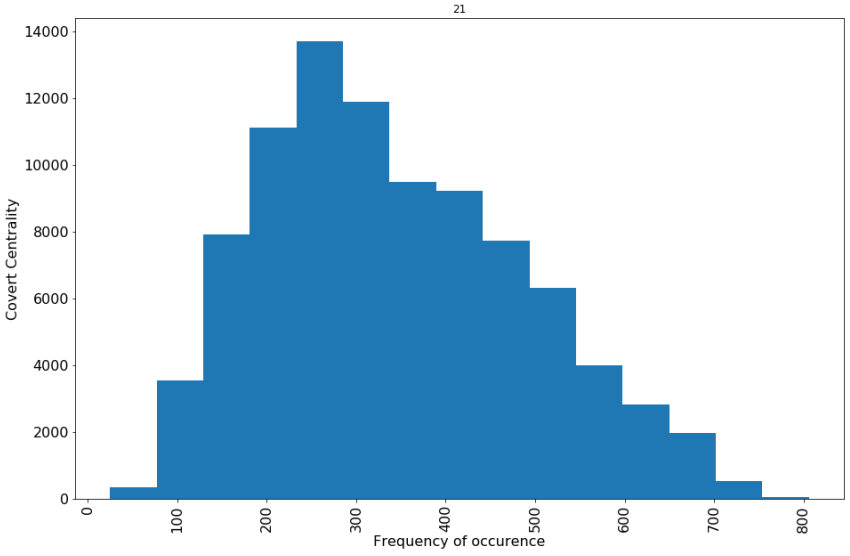


(b) Covert Centrality Frequency Distribution

Figure 4.26: Mouse Visual Cortex 2 [58]

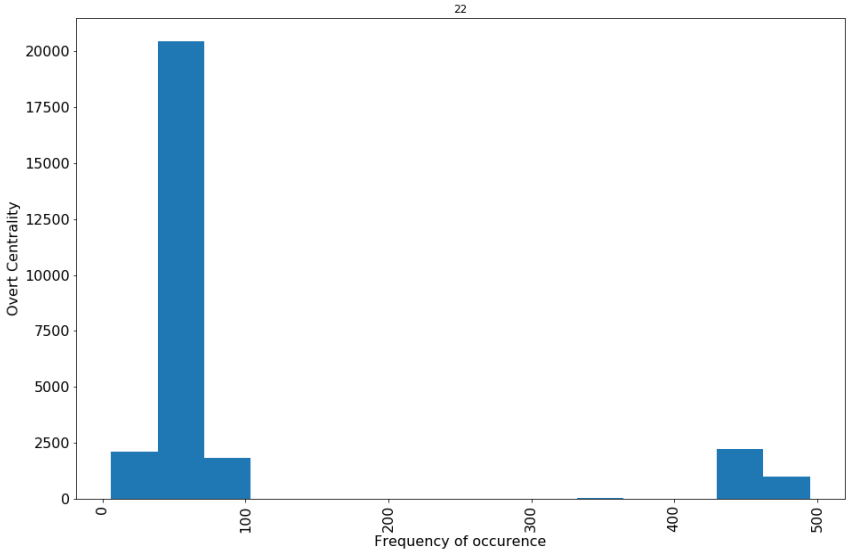


(a) Overt Centrality Frequency Distribution Comparison

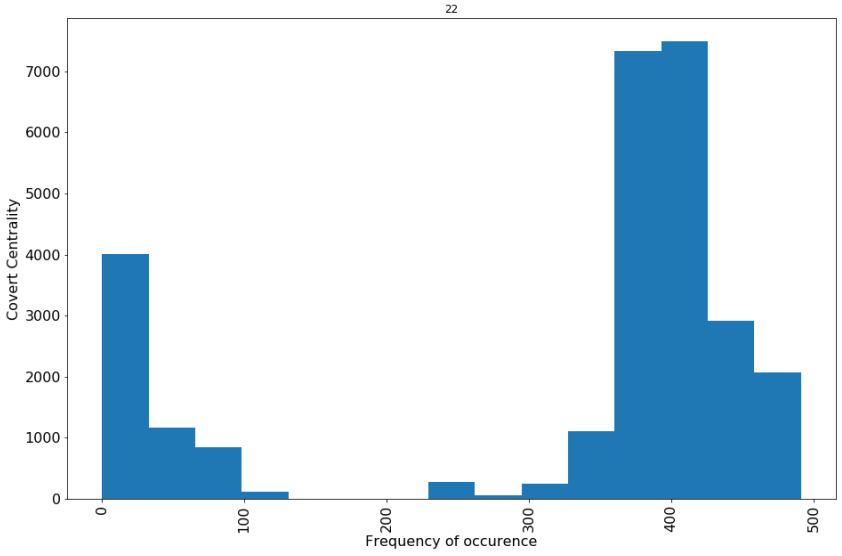


(b) Covert Centrality Frequency Distribution

Figure 4.27: Mouse Retina 1 [58]

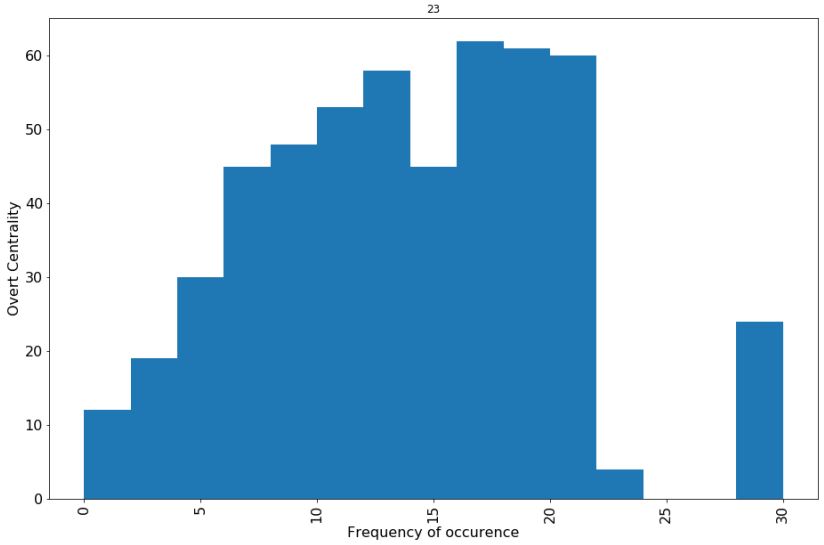


(a) Overt Centrality Frequency Distribution Comparison

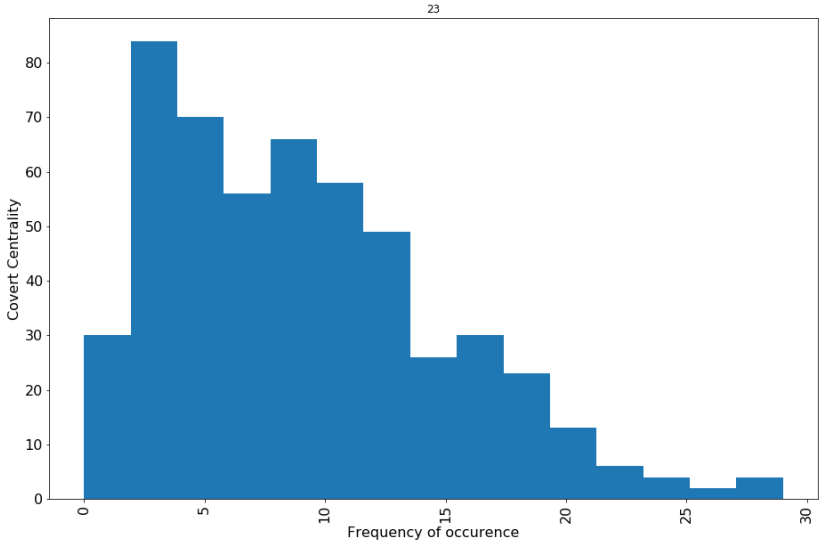


(b) Covert Centrality Frequency Distribution

Figure 4.28: Rattus Norvegicus [58]

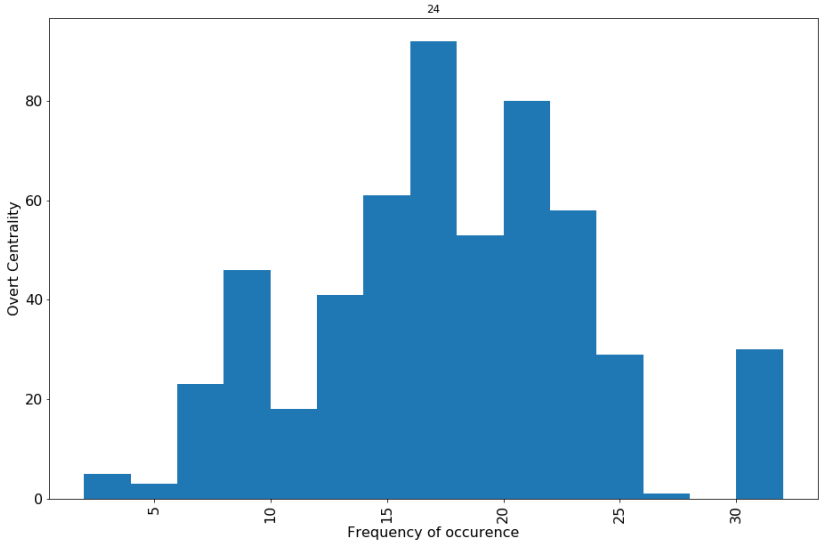


(a) Overt Centrality Frequency Distribution Comparison

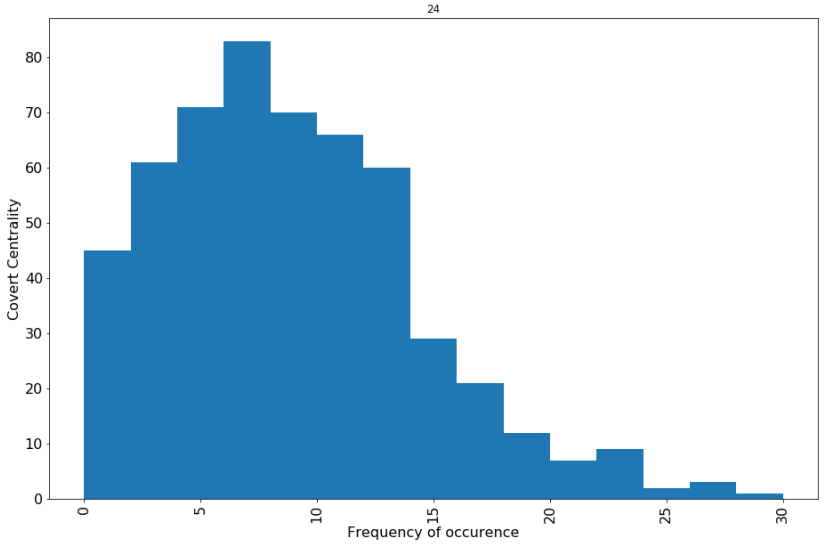


(b) Covert Centrality Frequency Distribution

Figure 4.29: Cross Parker Consulting [63]

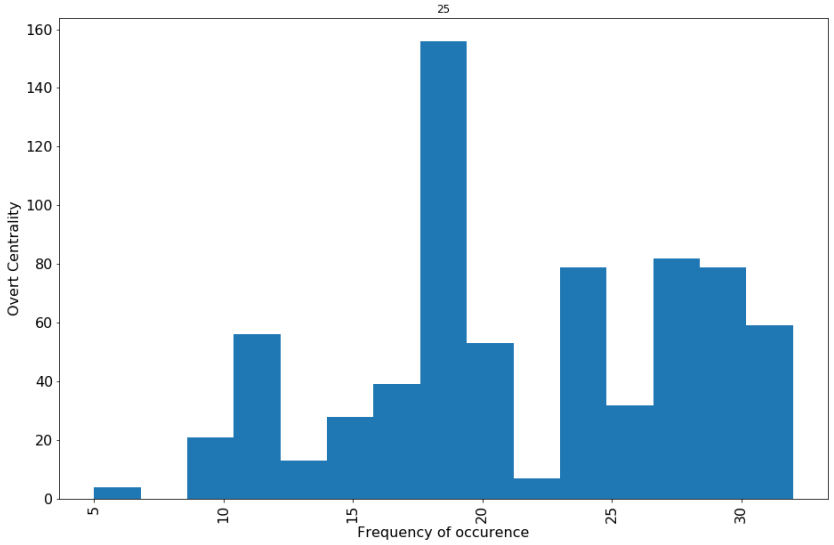


(a) Overt Centrality Frequency Distribution Comparison

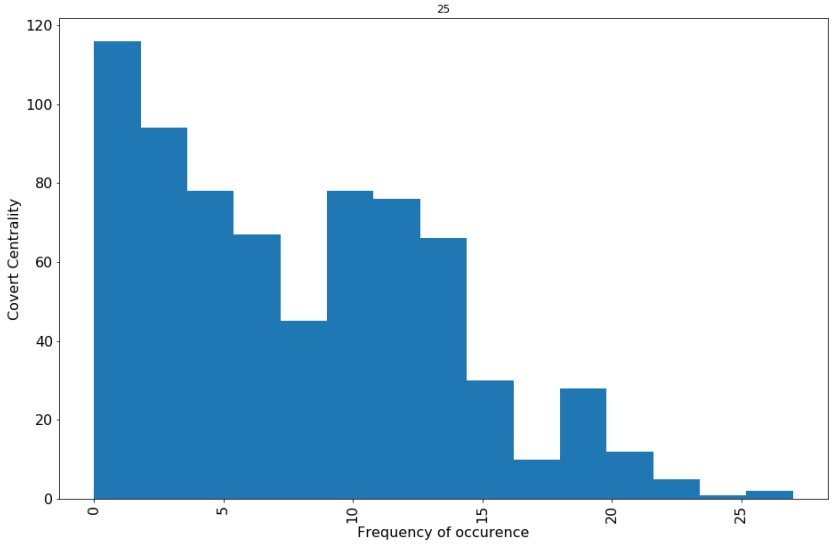


(b) Covert Centrality Frequency Distribution

Figure 4.30: Freemans EIES n48 1 [63]

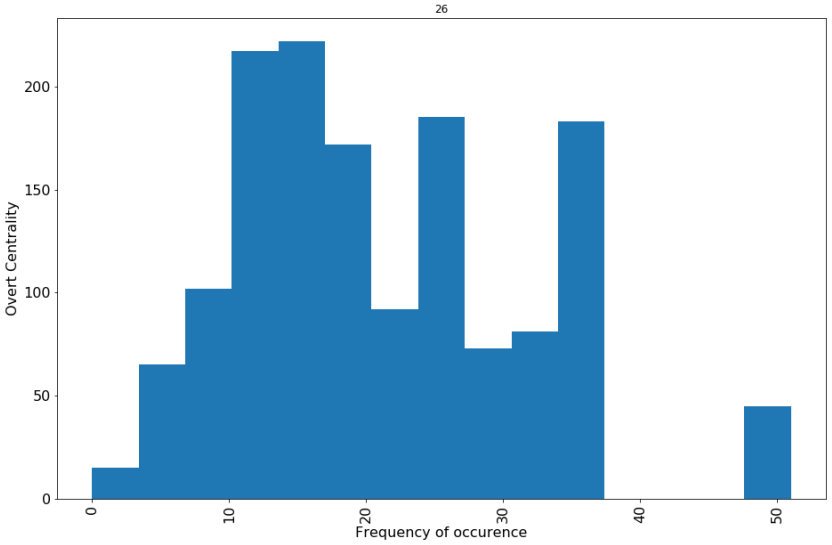


(a) Overt Centrality Frequency Distribution Comparison

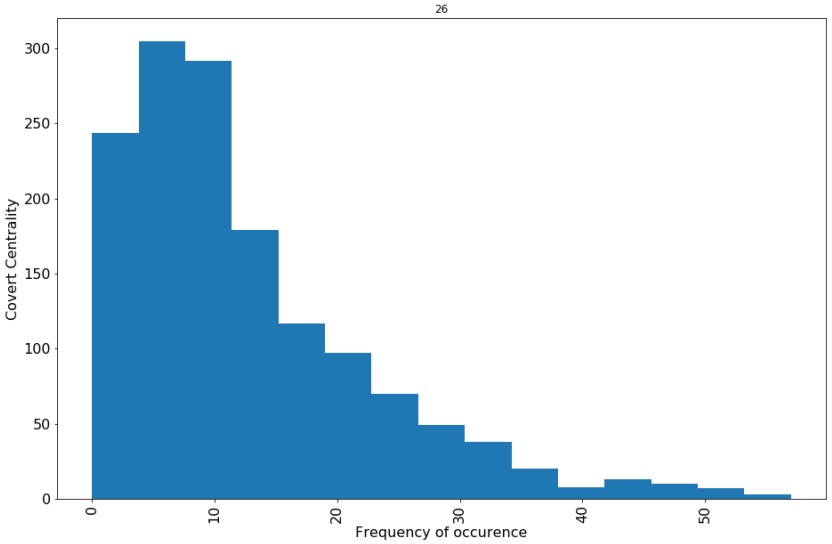


(b) Covert Centrality Frequency Distribution

Figure 4.31: Freemans EIES n48 2 [63]

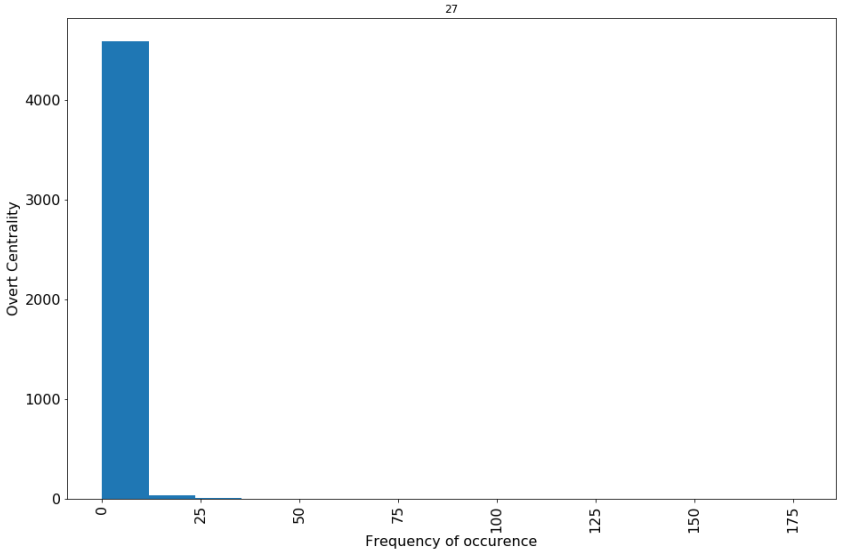


(a) Overt Centrality Frequency Distribution Comparison

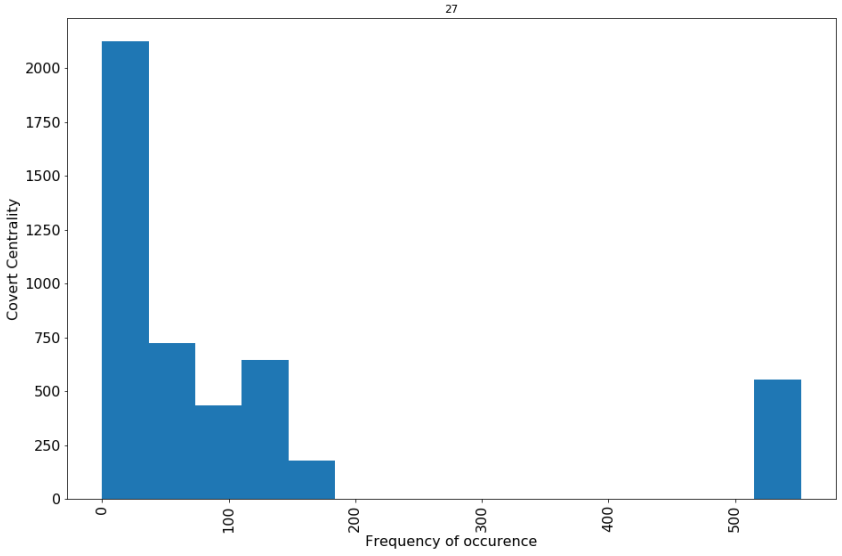


(b) Eva [41] Covert Centrality Frequency Distribution

Figure 4.32: Cross Parker Manufacturing [63]

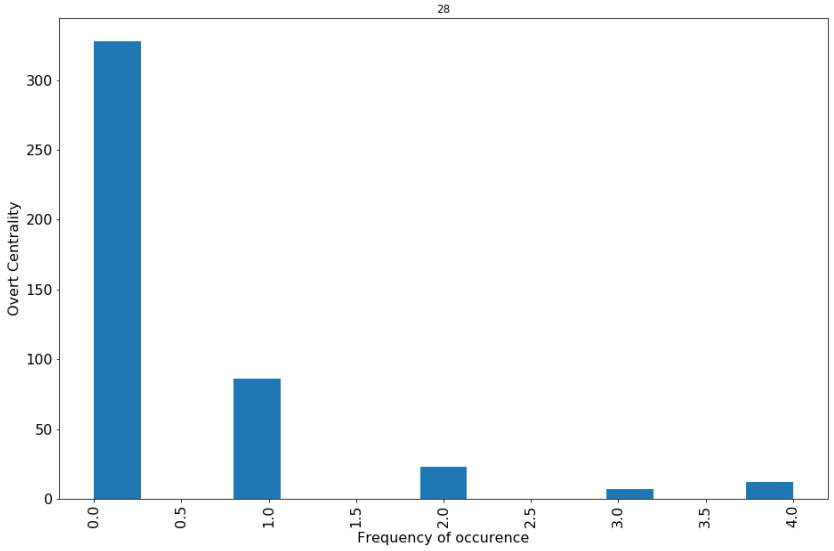


(a) Susceptibility Index Comparison

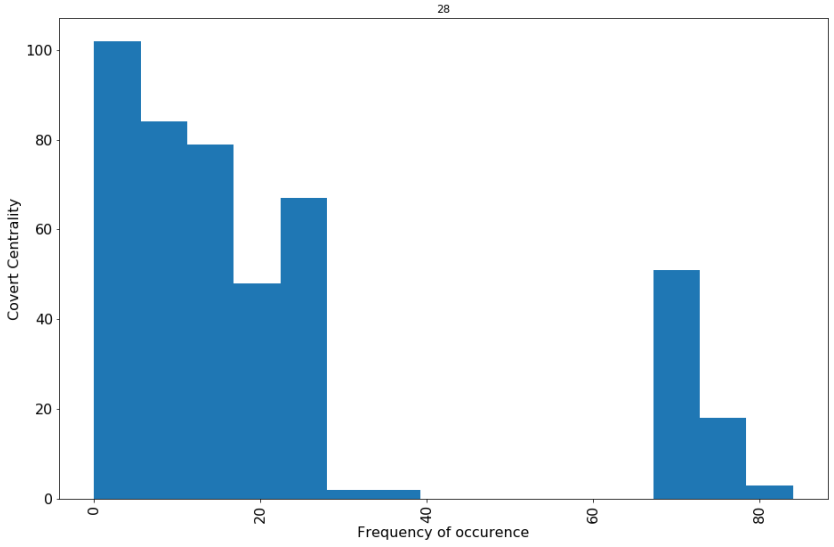


(b) Covert Centrality Frequency Distribution

Figure 4.33: Eva [41]

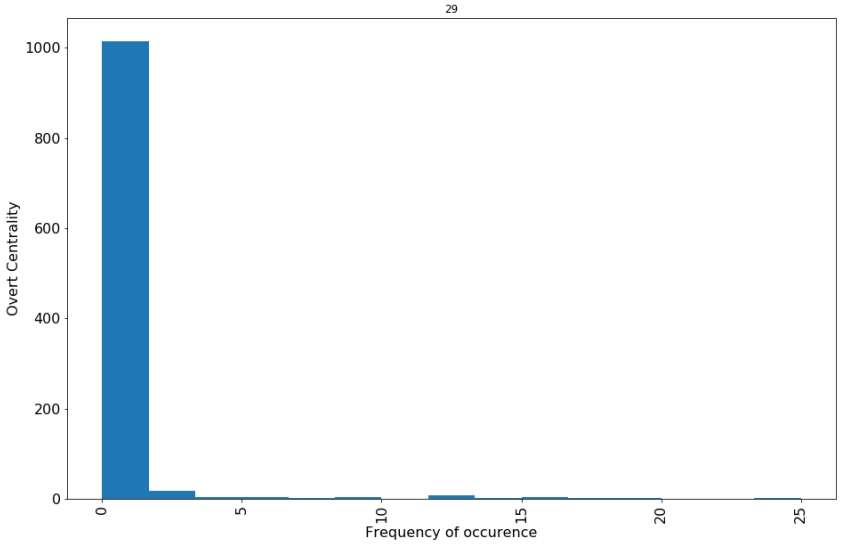


(a) Overt Centrality Frequency Distribution Comparison

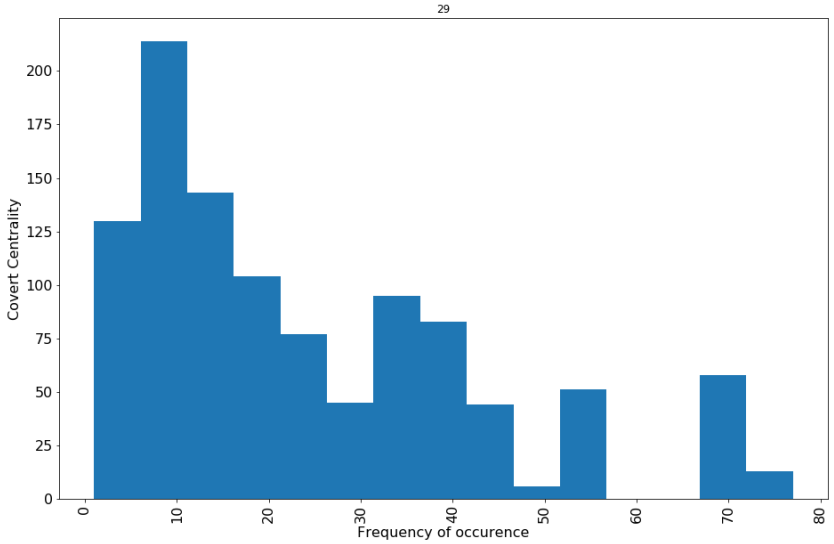


(b) Covert Centrality Frequency Distribution

Figure 4.34: Bitcoin Alpha [47]

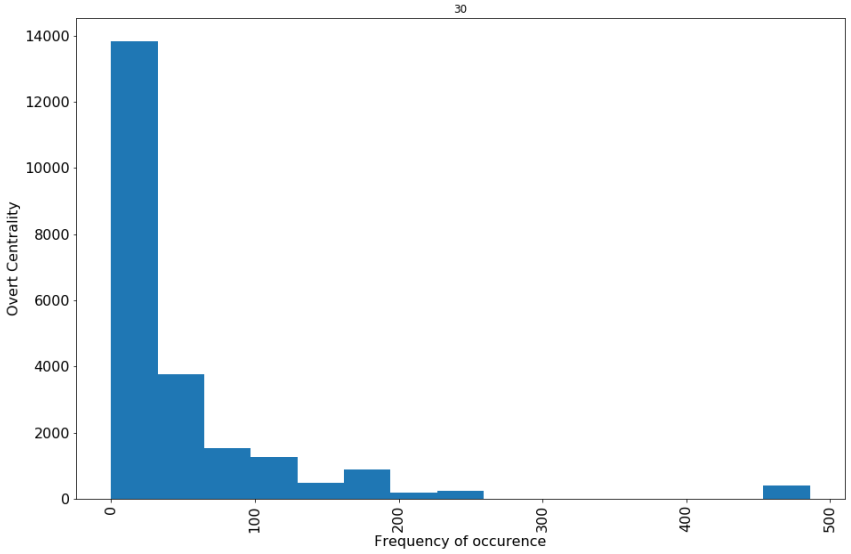


(a) Overt Centrality Frequency Distribution Comparison

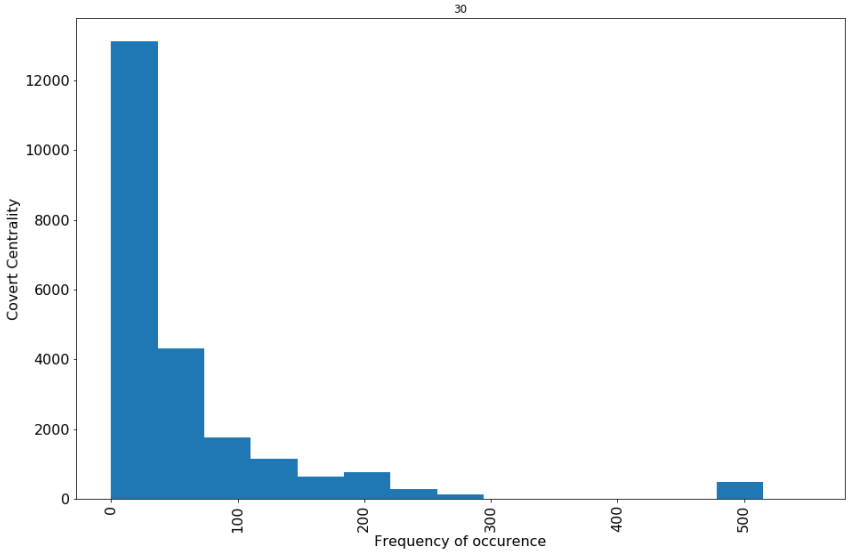


(b) Covert Centrality Frequency Distribution

Figure 4.35: Bitcoin OTC [47]

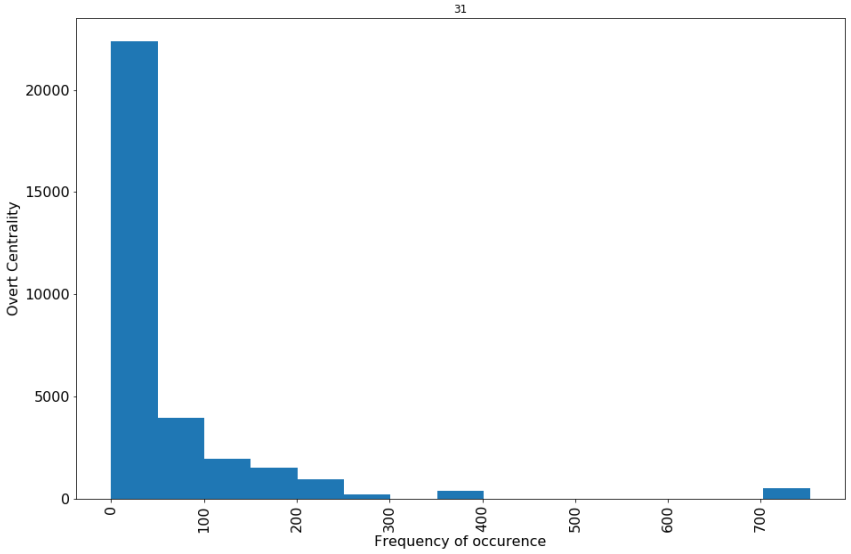


(a) Overt Centrality Frequency Distribution Comparison

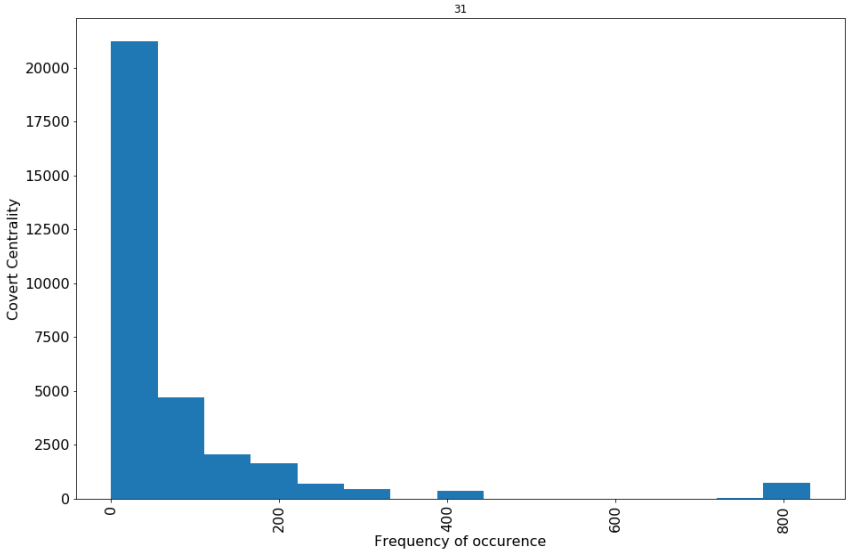


(b) Covert Centrality Frequency Distribution

Figure 4.36: Email EU Core [47]

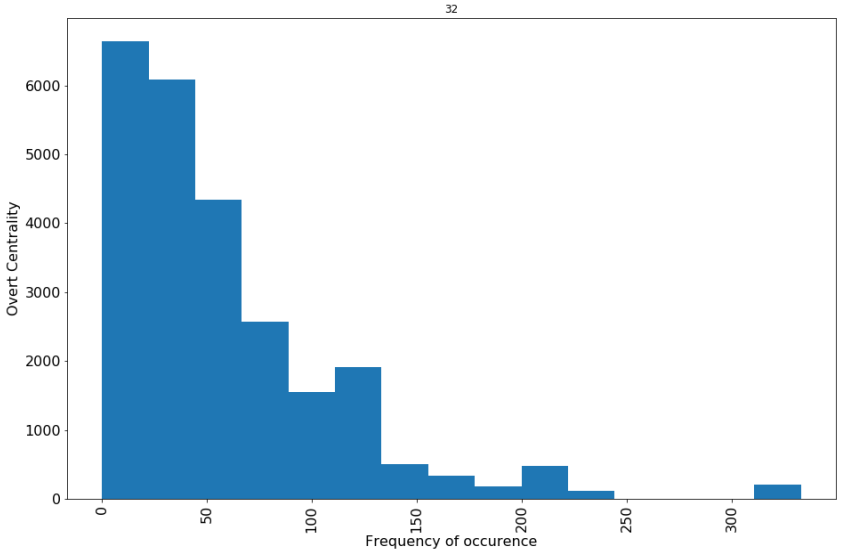


(a) Overt Centrality Frequency Distribution Comparison

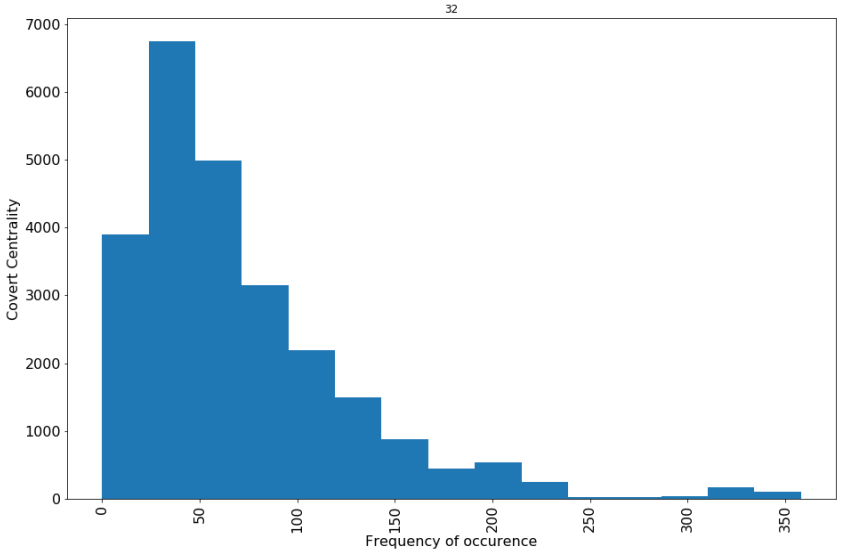


(b) Covert Centrality Frequency Distribution

Figure 4.37: Prison Inmate [5]

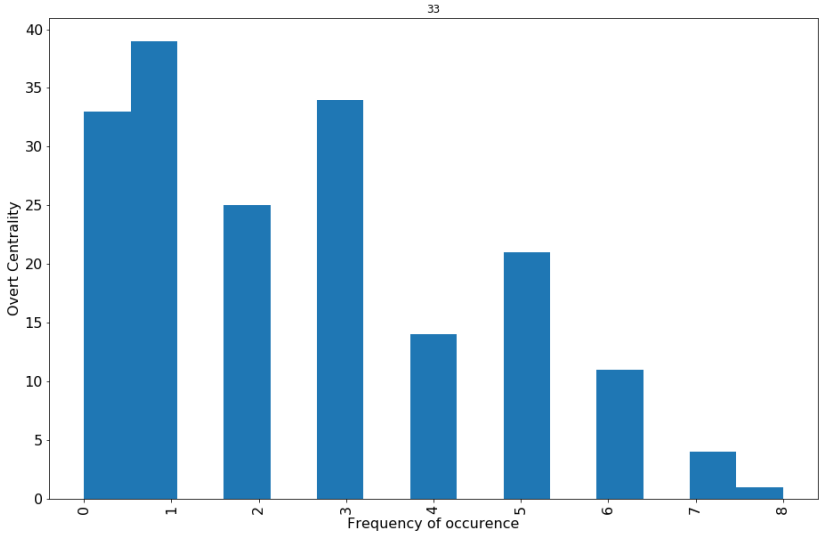


(a) Overt Centrality Frequency Distribution Comparison

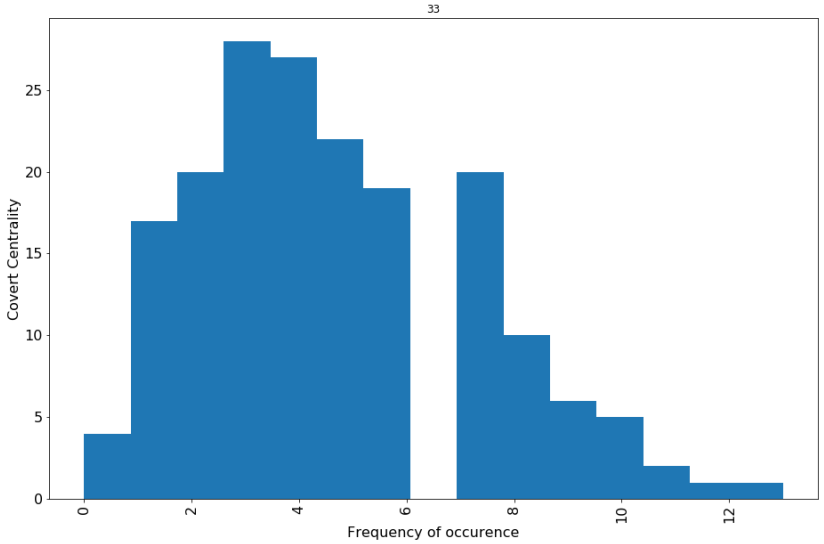


(b) Covert Centrality Frequency Distribution

Figure 4.38: UCIrvine [63]

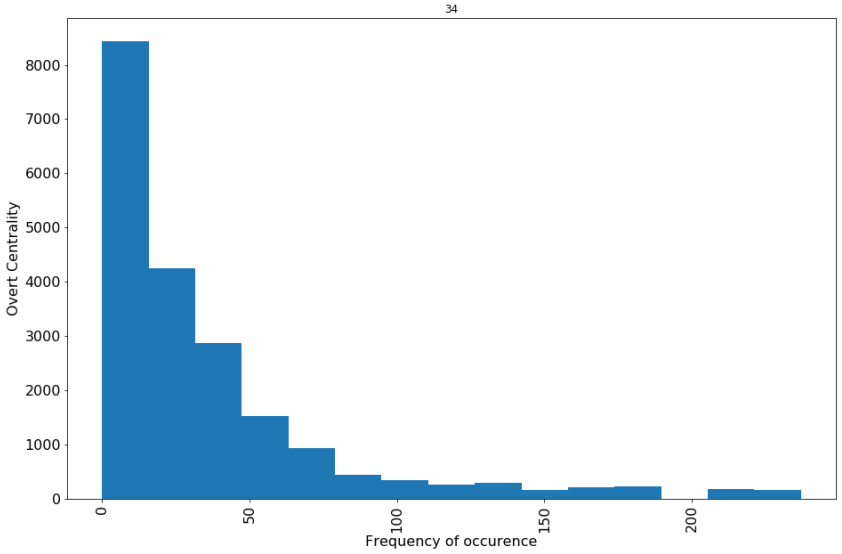


(a) Overt Centrality Frequency Distribution Comparison

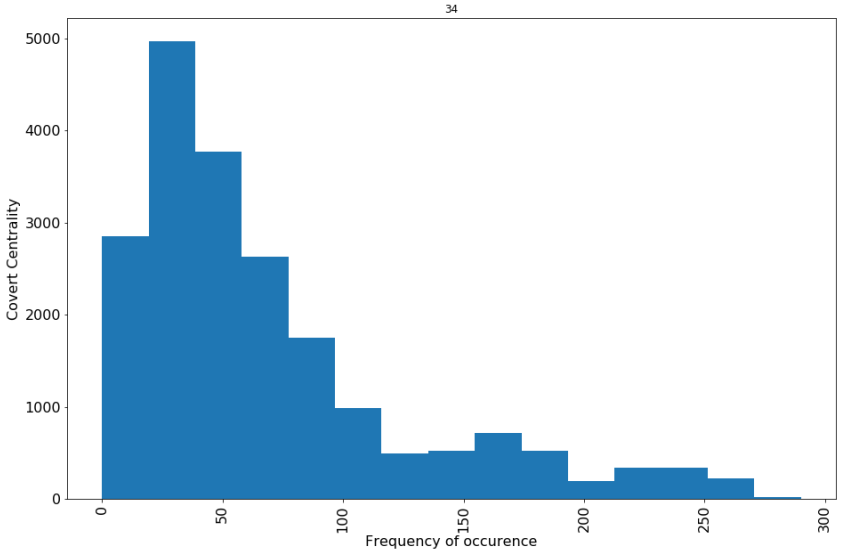


(b) Covert Centrality Frequency Distribution

Figure 4.39: WikiVote [47]

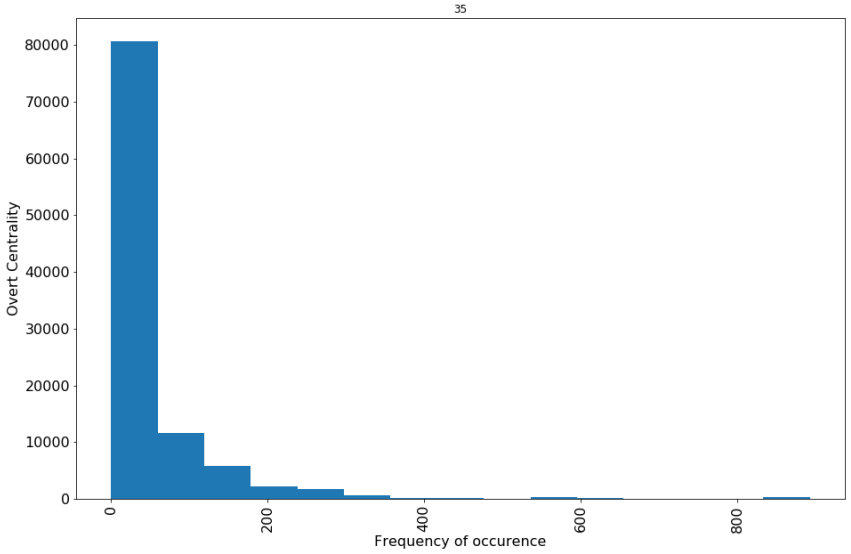


(a) Overt Centrality Frequency Distribution Comparison

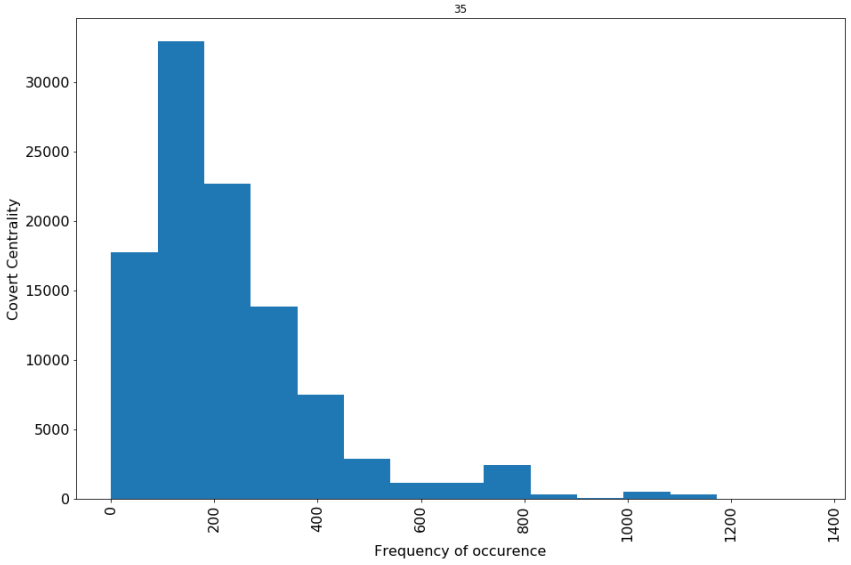


(b) Covert Centrality Frequency Distribution

Figure 4.40: E. coli transcription [5]

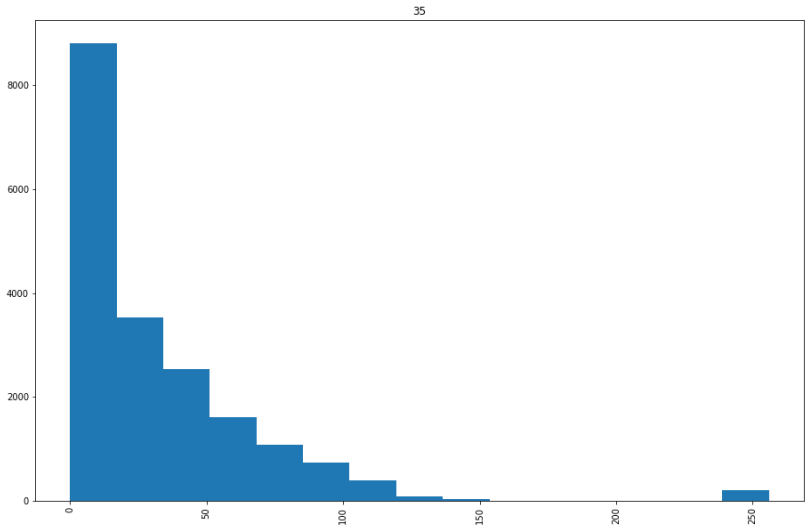


(a) Overt Centrality Frequency Distribution Comparison

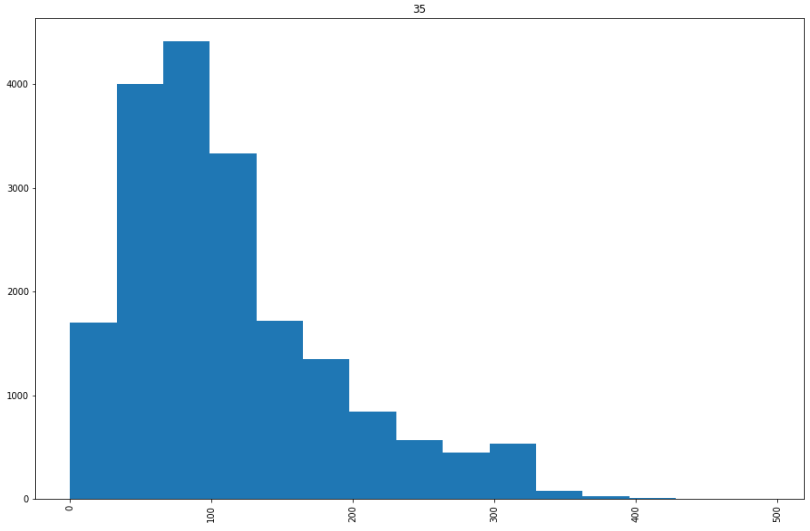


(b) Covert Centrality Frequency Distribution

Figure 4.41: Yeast transcription [5]



(a) Overt Centrality Frequency Distribution Comparison



(b) Covert Centrality Frequency Distribution

Figure 4.42: Political Blogs[2]

4.5 Comparing Overt and Covert Centrality with Existing Metrics

In this section, we compare our overt and covert centrality metrics with some key existing metrics that are commonly used in Social network analysis. This involves assessing the correlation between metrics across the 34 datasets from Section 2.5 used throughout the thesis. This allows us to assess the relationship between other related concepts.

We apply metrics as follows. We find the mean overt and covert centrality of edges, by averaging over the total centrality (i.e., overt plus covert centrality) of edges. We normalise this mean count by dividing through by the maximum possible overt/covert centrality of an edge (the number of edges in the network minus 2). The additional social network analysis metrics that we consider are defined in Table 2.4 in Section 2.5.

Concerning correlations, we calculate the Spearman rank correlation (since the data does not follow a normal between metrics using all 34 data sets to provide a sample for each comparison, as seen in Figure 4.43. A positive correlation indicates a monotonic increasing relationship, whilst a negative correlation indicates a monotonic decreasing relationship. The closer to 1 (or -1) the value, the stronger the relationship. In particular, a correlation between 0.4 and 0.6 we have classed as moderate, whilst between 0.6 and 0.8 strong and above 0.8 as very strong. We display this in a correlation matrix as shown in Figure 4.43. In Figure 4.44 we colour ingsignficiant correlation results in green, to show which of the correlations in Figure 4.43 are significant.

Results

Figure 4.43 indicates that all significant relationships between overt/covert centrality with existing metrics from Table 2.4 are at least moderate, though many are strong or very strong. In particular, we observe very strong correlation between mean cov-

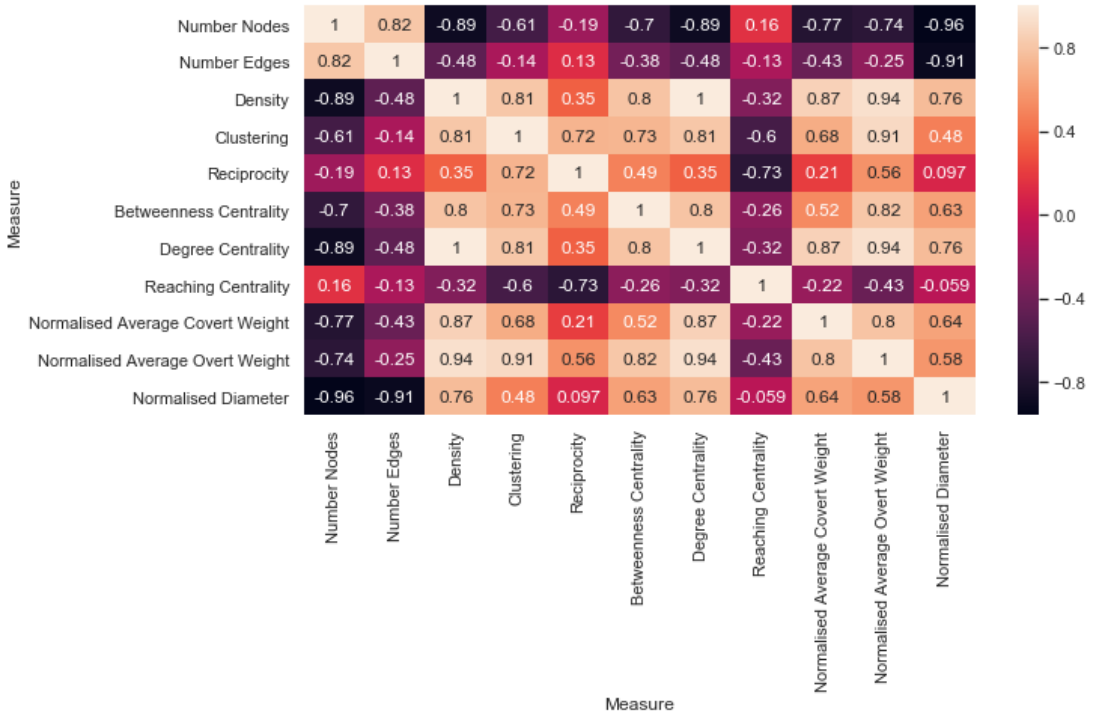


Figure 4.43: The correlation matrix comparing the spearman correlation of each metric across all 34 data sets.

ert weight of a network and the network density (0.87) and degree centrality (0.87). Mean overt weight of the network follows similarly, with the addition of very strong relationships with betweeness centrality (0.82) and global clustering (0.91). This indicates that mean overt weight, which is easy to locally calculate, is a useful proxy for these particular global centrality measures. Mean overt and covert weight of networks correlate very strongly with each other (0.8). Interestingly, there was no significant correlation between normalised average covert weight and reciprocity, and global reaching centrality. For overt centrality, there was no significant correlation with number of edges. We discussed in Section 4.4.2 the link between reciprocity, density and global clustering with overt and covert centrality. From this discussion we would expect a strong correlation between density, reciprocity and clustering with overt centrality, and a progressively weakening relationship with covert centrality. The reason we expect a weakening relationship with covert centrality and not a negative correlation is because increasing density, reciprocity and clustering in a network with low density, reciprocity

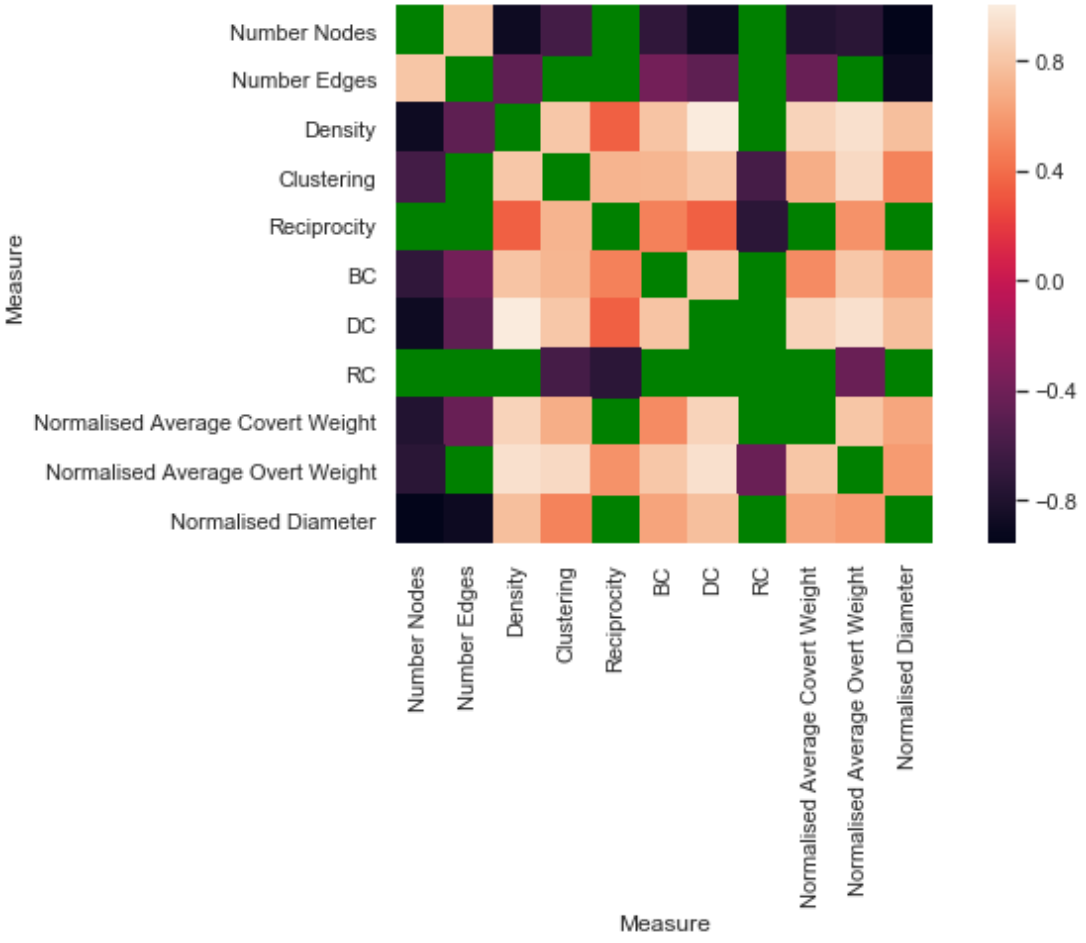


Figure 4.44: The correlation matrix comparing the spearman correlation of each metric across all 34 data sets, with insignificant correlation results indicated by green cells (p >0.05).

and clustering increases number of possibilities for covert edges at first, until a point is reached where density and reciprocity are high enough to force triads to progressively contain fewer and fewer covert edges. Figure 4.43 seems to indicate that indeed, density is very important to average overt centrality but to a lesser extent covert centrality. However, Figure 4.43 also shows that reciprocity does not have particularly strong correlations with overt or covert centrality (a correlation of 0.21 with covert centrality and 0.56 with overt centrality). Indeed, we saw in Section 4.4.2 that Social networks, which exhibit higher reciprocity than comparable networks, form long-tailed overt and covert centrality distributions. We concluded in Section 4.4.2 that solely looking to

density, reciprocity and clustering may not paint a full picture for overt and covert centrality, instead overt and covert centrality may rely on network structure and the number of triads that overlap on a single edge. Figures 4.43 strengthen this argument, showing that reciprocity may not be so important in controlling overt/covert centrality. Structural measures such as degree centrality or betweenness centrality may be a better representation of how many triads overlap on an edge, and therefore have a greater effect than density, reciprocity or clustering on the overt/covert centrality of edges, though we have not explored this in Section 4.4.2 because these measures were very low amongst all data sets (this may be due to averaging: there may be few areas of high betweenness/degree centrality but many areas with low degree/betweenness centrality). We now see in Figure 4.43 that betweenness centrality has a moderate correlation with covert edge weight and a very strong correlation with overt edge centrality (0.52 and 0.82 respectively), whilst degree centrality has a very strong correlation with both overt and covert edge centrality (0.87 and 0.94 respectively). This makes density, clustering and degree centrality the biggest factors affecting overt and covert centrality. Interestingly, the number of shortest paths flowing through an edge has more effect on overt centrality than covert centrality: meaning overt edges may be more important to paths.

The existence of significant, directed correlations motivates further investigation through direct comparison of the mean overt/covert centrality with each metric for each data set, the results for which are found in Figure 4.45-4.52. We observe in Figures 4.45-4.52 that results are different depending on comparison with mean overt or covert centrality. For example, when comparing reaching centrality with mean covert weight the Social networks category of data points cluster together. When comparing reaching centrality with mean overt weight, the data points are more dispersed along the x axis, pulling apart this cluster of nodes. The same effect can also be seen when comparing diameter with the average overt and covert weight in Social networks.

These results show that overt and covert centrality measures have a role to play in differentiating the alternative network classes. Without overt and covert centrality,

the data points in Figures 4.45-4.52 would project onto the y-axis, with little or no differentiation between some classes of network in many cases. One such example is the Clustering measure, where average overt centrality is able to distinguish alternative classes of network with similar clustering values (i.e., separate in the x-axis). This is particularly the case for low clustering levels (e.g., 0.3 or lower).

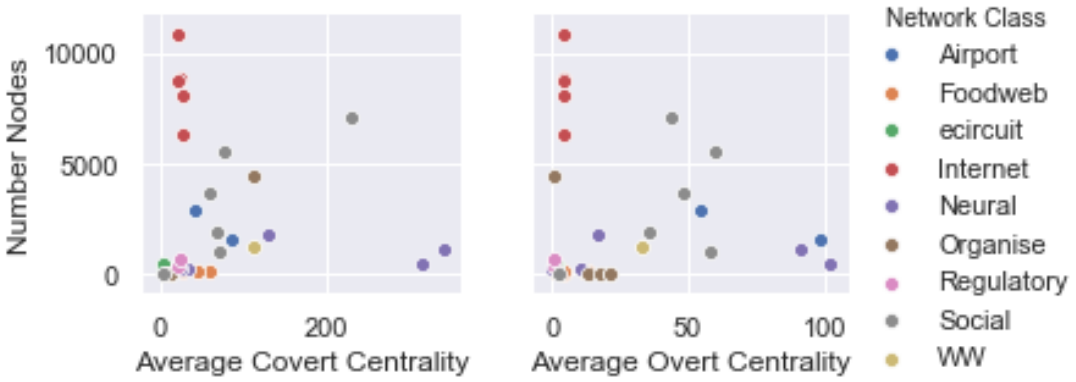


Figure 4.45: Scatter plots showing the relationship between number of nodes in a network and mean covert and overt edge centrality.

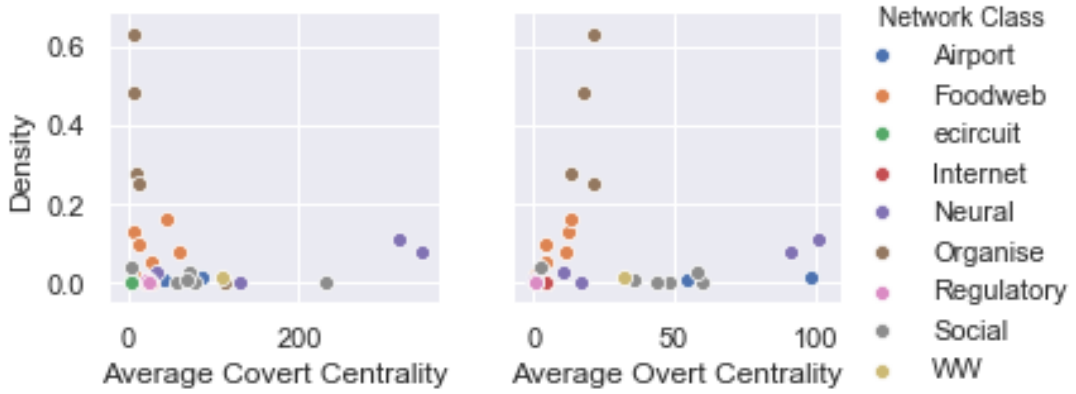


Figure 4.46: Scatter plots showing the relationship between density of a network and mean covert and overt edge centrality.

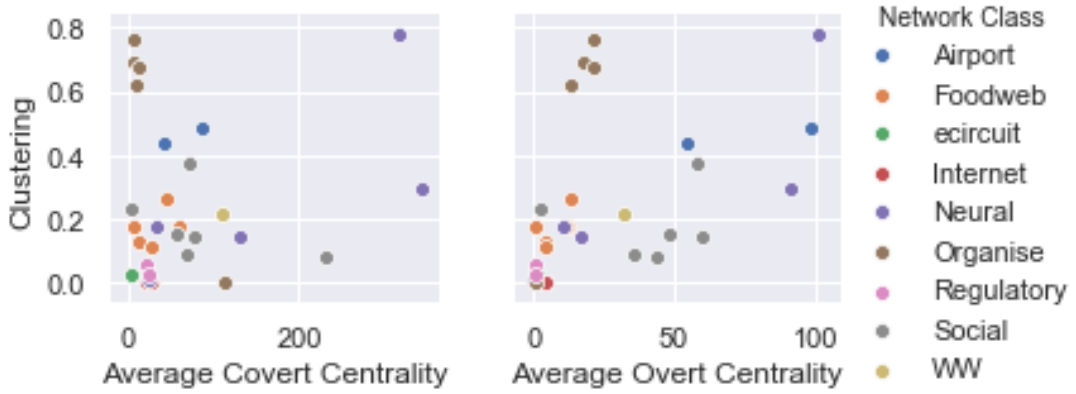


Figure 4.47: Scatter plots showing the relationship between clustering of a network and mean covert and overt edge centrality.

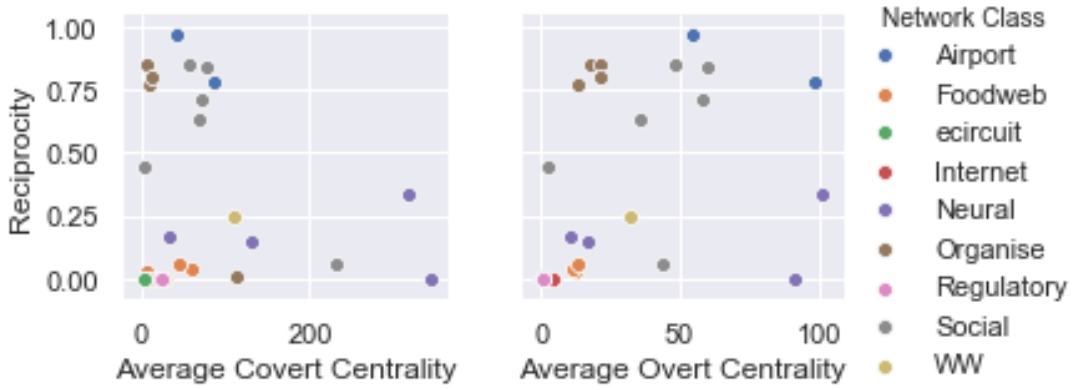


Figure 4.48: Scatter plots showing the relationship between reciprocity in a network and mean covert and overt edge centrality.

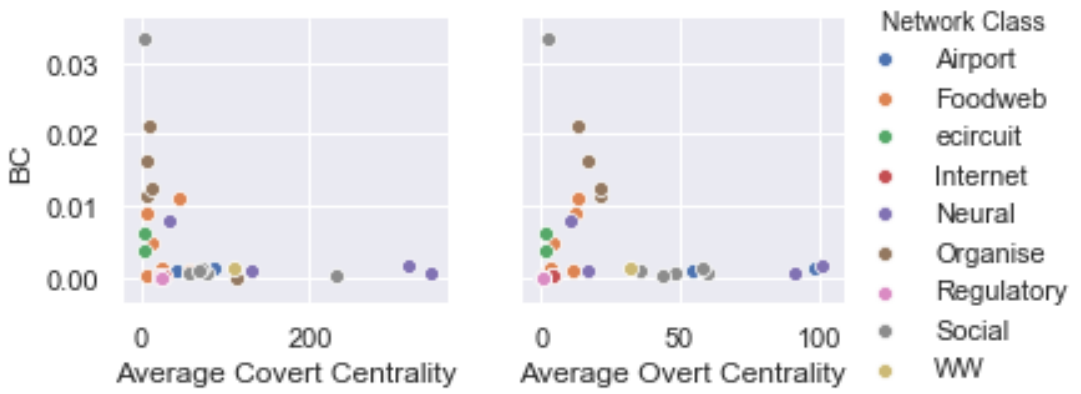


Figure 4.49: Scatter plots showing the relationship between reciprocity in a network and mean covert and overt edge centrality.

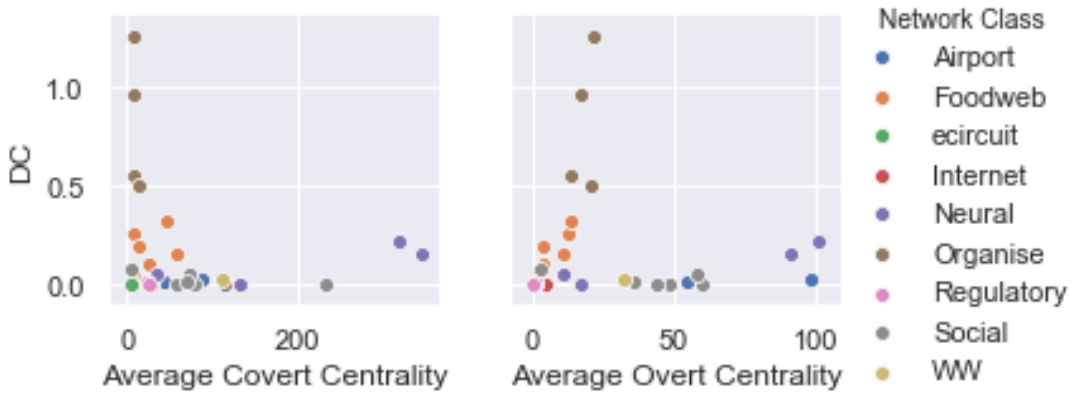


Figure 4.50: Scatter plots showing the relationship between average degree centrality and mean covert and overt edge centrality.

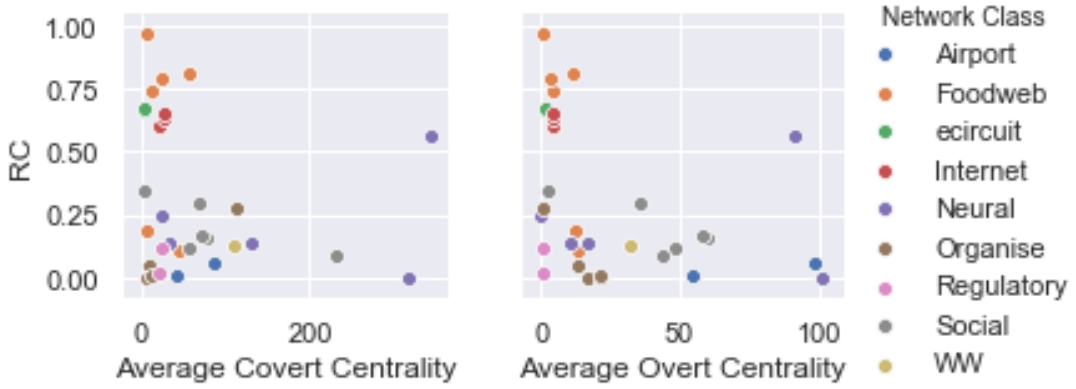


Figure 4.51: Scatter plots showing the relationship between average reaching centrality and mean covert and overt edge centrality.

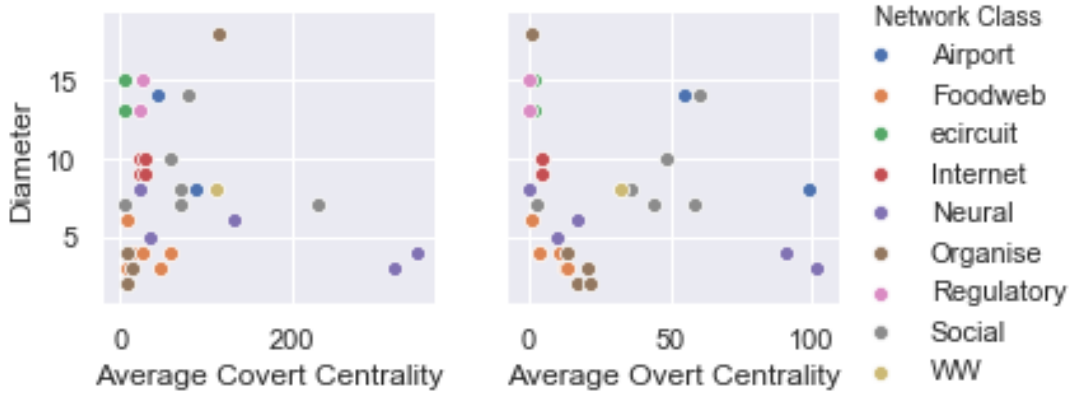


Figure 4.52: Scatter plots showing the relationship between average diameter and mean covert and overt edge centrality.

4.6 Summary

We have introduced a novel and fundamental approach to characterising the centrality of edges in networks based on their role in supporting connectivity within induced triads. This constitutes a binary classification of edges with respect to a particular induced triad within which they participate. This concept is important because triads are the smallest possible (non-trivial) induced substructure in a network. Therefore our definition of

overt and covert centrality with respect to edges captures the lowest level at which non-trivial network connectivity can be considered, beyond the concept of an edge itself. Using counting arguments based on in and out degrees within a triad, we have derived formula to determine the both overt and covert edge centrality. This means that overt and covert centrality can be determined without recourse to lengthy search - the concept is locally defined and therefore scalable.

To demonstrate these concepts on 'real-world' scenarios, we have applied it to data derived from 34 public data sets. From profiling the overt and covert centrality distributions across these networks, we have shown that the frequency of overt and covert edges is useful in presenting further insights both within and between different classes of network. These concern the relative dependency that networks have on the prevalence of edges with key roles concerning connectivity. On the one hand, overt edges play a role in potential containment of content transferred through an edge, since the edge cannot support dissemination to the third node in the triad. Alternatively, overt edges align with the potential for onward local dissemination, since by definition an overt edge supports a path to the third node in the triad. This provides an alternative lens through which global network connectivity can be considered at a local level.

We have additionally compared overt and covert centrality with a number of conventional social network analysis metrics, including global reach centrality, mean degree centrality, betweenness centrality, density, clustering, reciprocity and diameter. Potential significant correlations have been considered with a view to understanding relationships with these metrics. By examining the relationship between these metrics and overt/covert centrality across 34 networks, we have found that overt/covert centrality can provide additional insights. These relate to distinguishing between classes of network, which the global social network metrics would fail to achieve in isolation, for numerous cases.

In summary, we observe that overt and covert centrality are important concepts for characterising a network's structural potential for dissemination and contagion, based

on consideration of a network's underlying building blocks. Overt and covert centrality of an edge may depend upon the density or reciprocity of a network (therefore affecting the type of triads induced by the network, and thereby the likelihood an edge is overt or covert) or the number of triads overlapping on an edge (affected by the structure of the network, thereby giving edges which overlap on more triads a higher overt/covert centrality).

This chapter addresses the second research question, namely:

RQ2: The role of edges in triad connectivity: if connectivity within the triad is dependent on the arrangement of its edges, then how do we identify which edges are most fundamental to enabling connectivity within the triad?

We do this through classifying edges as overt or covert, resulting in the second contribution: **C2:** *A new classification for edges based on their role in supporting connectivity within triads.*

We also consider multiple triads through creating overt and covert centrality measures for edges, resulting in the third contribution: **C3:** *A new local centrality metric for edges based on our new edge classification in* **C2**.

Chapter 5

The Role of Overt and Covert Edge Centrality in Paths

Paths are the fundamental entity in graphs that enable connectivity, using intermediate vertices to relay communication or sustain a relationship, depending on the context. This introduces the potential for intermediate vertices to relay knowledge from the path to other vertices. Generally speaking, two extreme scenarios can be envisaged - on the one hand a path may connect two vertices while minimising risks of dissemination beyond those vertices in the connecting path. Alternatively widespread dissemination from vertices on the connecting path might be desirable. The choice of edges in a path fundamentally affects this.

In this chapter we examine how the distribution of edges in paths affects this problem, using overt and covert edge centrality. These measures are highly valuable because they locally characterise the potential for containment along, or dissemination from, a path. For example, an edge with high covert centrality and low overt centrality is potentially useful if spreading knowledge beyond a selected path is undesirable. The measures of overt and covert provide this characterisation by directly considering the role of edges within induced triads that participate in a path.

In considering this problem, it leads to possible trade offs due to different potential objectives. For example, shortest paths in a graph can be defined by the total number of edges along a path, but if containment is important, then minimising total overt

centrality of edges on a path may be more important, leading to a longer path, in terms of number of edges. The metrics of overt centrality and covert centrality allow such trade-offs to be assessed, which have previously not been undertaken. We examine this in the context of different test graphs, to understand the extent that there is freedom and compromise in trade-offs for path selection. Because this issue is highly sensitive to structure and density of edges, we synthesise a number of different classes of network and examine the differences.

5.1 Overview

To characterise paths we consider the hypothetical situation where messages are being transmitted along paths from a source to a single destination. Specifically, suppose in a graph G we were interested in sending a message between two vertices through a series of intermediaries. To do this, we send a messages along edges from the source vertex to intermediary vertices until the target vertex is reached. In other words, we need to construct a path between two vertices.

Definition 30. *Let $p(u, v) = (u_1 = u, u_2, \ldots u_n = v)$ denote a path from u to v, where $u_i \in V(G)$ for $i = 1, \ldots, n$ and $(u_i, u_{i+1}) \in E(G)$ for $i = 1, \ldots, n-1$. The length of the path $p(u, v)$ is given by the number of edges, and denoted by $l(p) = |p(u, v)| - 1 = n - 1$.*

In many networks, particularly those with high connectivity, there are often many paths between pair of vertices. Therefore, we can define additional criteria to choose the best path to suit the objectives of the scenario. For example, sending a message on a path with the minimum number of edges would provide the quickest way to transmit the message. A path of minimum edge length is known as a*shortest path*, formally defined as follows.

Definition 31. *Let $G = (V(G), E(G))$ be a graph. Then suppose every edge $(x, y) \in E(G)$ has a weight, $w_{x,y}$. Then a shortest path from u to v, for $u, v \in V(G)$ is a path*

$p(u, v) = (u = u_1, u_2, \ldots, u_n = v)$ *such that*

$$\sum_{(u_i, u_{i+1}) \in P} w_{u_i, u_{i+1}} \tag{5.1}$$

is minimised. When $w_{(x,y)}$ *is equal for every edge, then* $\sum_{(u_i,u_{i+1})\in P} w_{u_i,u_{i+1}}$ *finds a path of minimum length.*

Throughout the rest of the thesis, we will refer to a shortest path (or minimum length path) as a path between a pair of vertices such that the number of edges is minimised (i.e, from Definition 31 a shortest path where the weight of each edge is equal). We refer to least weight, or minimum overt/covert paths as, from Definition 31, a shortest path between a pair of vertices where the weight of its edges are equal to their overt/covert centrality .

In the hypothetical case that a message we want to send contains sensitive information, and beyond using technology to encode information (e.g., in a human social network) we may wish to consider the local graph positioning of individuals that are in the path as these individuals have the opportunity to relay to others. Thus in such a scenario, when sending this message across a path we may want to minimise the number of vertices the message can reach outside of the path. This may be applicable in military scenarios where, supposing information is secretive and passed through a chain of intermediaries, overt edges may not be trustworthy as they enable further spread of secretive information to third parties outside the chain of intermediaries: the message has been ‘leaked ’.

For modelling purposes we assume that any edge directed out of a vertex and into another has potential to spread this message to a third party. In other words, any edge that is overt (see Definition 27 in Chapter 4) has the potential to spread a sensitive message. Therefore, if our aim is to minimise potential spread, then in this scenario we want to find a path where the total overt centrality of edges is minimal. We explain this further in Example 5.1.1.

Example 5.1.1. *Suppose there exist two vertices u_1 and u_5 in graph G with two paths between them, $p_1 = (u_1, u_2, u_3, u_4, u_5)$ and $p_2 = (u_1, u_6, u_7, u_5)$, as shown in Figure 5.1. Suppose there also exist the edges (u_6, u_9), (u_6, u_{10}) and (u_6, u_{11}). Suppose we want to send a message down either path from u_1 to u_5 down the edges in the graph and want to ensure that potential spread of the of the message is minimised. Then, p_2 is shorter than p_1 in terms of edge length, yet the number of vertices the message could potentially spread to (highlighted through the traversing the red edges) in p_2 is greater than the number of vertices the message could reach by traversing p_1 (highlighted through the traversing the blue edges). Therefore, choosing p_1 over p_2 would promote containment of the message, at the expense of needing an extra edge in the path.*

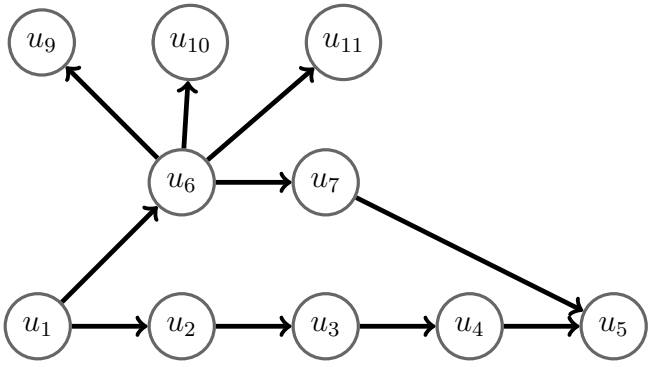


Figure 5.1: The graph G, with paths $p_1 = (u_1, u_2, u_3, u_4, u_5)$ and $p_2 = (u_1, u_6, u_7, u_5)$. The potential spread of a message sent down p_1 is highlighted in blue, whereas the potential spread of a message sent down p_2 is highlighted in red.

Following this example, it is important to explore the relationship between shortest paths and paths of minimum total overt centrality. We also investigate minimum total covert centrality, as this could represent an alternative scenario: where maximising spread of knowledge from a path is desirable. This may be applicable to other scenarios, such as infection control, where we can model how to achieve maximum spread and take measures to mitigate this.

Paths of minimum overt/covert centrality may represent different scenarios in networks where edges do not represent communication. As discussed in Chapter Four, identifying

edges with a high overt centrality identifies which relationships enable the greater number of further relationships (or visa versa for covert). These 'further relationships ' themselves enable further relationships, which enable further relationships, etc. In other words, these naturally forming paths. Therefore, considering paths which control for minimum/maximum overt/covert centrality may recognise those paths made up of edges which which enable the greater number of further relationships; acting as a backbone for the entire network. In the language of food web networks, where an edge highlights a trophic relation between prey and predator, paths of maximally overt paths represent individual food chains that enable the greatest number of trophic relations outside of the path, and therefore may be vital to enabling the survival of the greater species present in the food web. In the language of airport networks, identifying maximally overt paths may identify core flight paths between two airports which enable the greatest number of possible further connecting destinations. In short, identifying which paths are maximally overt may identify which paths are crucial to maintaining network structure.

5.2 Methodology

In this section we seek to construct four trade-off metrics that allow us to consider the relationship between path length (based on number of edges) when seeking to control for the overt and covert centrality characteristics of a path. We introduce *overt and covert path weights* to assess the total overt/covert centrality of the edges in a path in Section 5.2.1. We then use these path weights to construct sets of minimal paths: those with minimal weight, and those with minimal length in Section 5.2.2. We construct our trade-off measures: the *overt-length/covert-length trade-off* (Section 5.2.3) and *length-overt/length-covert trade-off* (Section 5.2.4). We then transform these trade-off measures into a global metric, the *average improvement in edges*, which allows us to observe the overall effect on the graph when choosing paths which contain edges that are less overt/covert over a path of minimum length.

5.2.1 Overt and Covert Path Weights

Let $G = (V(G), E(G)$ be a graph throughout. We define the overt weight of a path as the sum of overt centralities of all the edges in the path as follows.

Definition 32. *The* overt weight*, $w_o(p)$, of a path $p = p(u, v) = (u_1 = u, u_2, \ldots u_n = v)$ is given by the sum of the overt centralities of the edges contained in p. Therefore:*

$$w_o(p) = \sum_{k=1}^{n-1} (u_k, u_{k+1})_o \tag{5.2}$$

The function to compute $w_o(p)$ can be found in Algorithm 9. This utilises the function OVERTCENTRALITY from Algorithm 6 in Chapter 4 to determine the overt centrality of an edge.

Algorithm 9 OVERTPATHWEIGHT(p): Computes the total overt weight $w_o(p)$ of a path p.

Input: Path $p = (u_1 = u, u_2, \ldots, u_n = v)$ between vertices u and v in a graph G

Output: The overt weight of the path, $w_o(p)$

1: $w_o(p) = 0$
2: **for** $(u_k, u_{k+1}) \in E(p)$ **do**
3: $\quad w_o(p) = w_o(p) + (u_k, u_{k+1})_o$
4: **return** $w_o(p)$

Similarly to overt weight of a path, we make the following definition for covert weight:

Definition 33. *The* covert weight *of a path $p = p(u, v) = (u_1 = u, u_2, \ldots u_n = v)$ is given by:*

$$w_c(p) = \sum_{k=1}^{n-1} (u_k, u_{k+1})_c \tag{5.3}$$

The function to compute $w_c(p)$ can be found in Algorithm 10. This utilises the function COVERTCENTRALITY from Algorithm 7 in Chapter 4 to determine the covert centrality of an edge.

Algorithm 10 COVERTPATHWEIGHT(p): Computes the total covert weight $w_c(p)$ of a path p.

Input: Path $p = (u_1 = u, u_2, \ldots, u_n = v)$ between vertices u and v in a graph G

Output: The covert weight of the path, $w_c(p)$

1: $w_c(p) = 0$
2: **for** $(u_k, u_{k+1}) \in E(p)$ **do**
3: $\quad w_c(p) = w_c(p) + (u_k, u_{k+1})_c$
4: **return** $w_c(p)$

5.2.2 Choosing Minimal Paths

We are interested in understanding the relationship between key metrics when considering paths in the context of overt/covert centrality – specifically the minimum number of edges, and the best overt or covert weight that can be achieved. These metrics clearly trade-off against each other. To examine this, we apply Dijkstra's [17] shortest path algorithm with path length as the first objective and overt (or covert) weight as a secondary objective, and then vice versa. In our notation, we use the convention of subscripts to denote the first objective, and superscripts to denote the second.

Clearly multiple alternative shortest paths may exist (i.e., different paths having the same shortest possible length). We define the set of all these shortest paths between a pair of vertices in Definition 34.

Definition 34. *Let $P(u, v)$ denote the set of all paths between u and v. Let $\delta_s(u, v)$ denote the length of a shortest path from u to v (in terms of hops), i.e.:*

$$\delta_s(u, v) = \min_{p \in P(u,v)} l(p)$$

Then $P_s(u,v) \subseteq P(u,v)$ denotes the set of all paths of least length between u and v, where:

$$P_s(u,v) = \{p \in P(u,v) | l(p) = \delta_s(u,v)\} \tag{5.4}$$

We can now search amongst the shortest paths to find the subset which have the minimum overt weight:

Definition 35. *Given a set of paths P, let $\Delta_s(P)$, $\Delta_o(P)$ and $\Delta_o(P)$ denote the minimum of their lengths, overt weights and covert weights respectively as:*

$$\Delta_s(P) = \min\{l(p) | p \in P\} \tag{5.5}$$

$$\Delta_o(P) = \min\{w_o(p) | p \in P\} \tag{5.6}$$

$$\Delta_c(P) = \min\{w_c(p) | p \in P\} \tag{5.7}$$

For a given vertex pair u and v, we define the subset of shortest paths that have minimum overt weight as $P_s^o(u,v) \subseteq P_s(u,v)$ where:

$$P_s^o(u,v) = \{p \in P_s(u,v) | w_o(p) = \Delta_o(P_s(u,v))\} \tag{5.8}$$

Similarly, we let $P_s^c(u,v)$ denote the subset consisting of the least covert paths of those between u and v that have shortest length, where:

$$P_s^c(u,v) = \{p \in P_s(u,v) | w_c(p) = \Delta_c(P_s(u,v))\} \tag{5.9}$$

Thus $P_s^o(u,v)$ and $P_s^c(u,v)$ respectively contain the least overt and covert paths among those with shortest length between u and v.

As well as considering the weight of the shortest path between vertices, we can alternatively consider the *least weight overt/covert path* between a pair of vertices. Note that the least weight overt/covert path may contain more edges than a shortest path (but never fewer), and may have a lower overt/covert weight than a shortest path. To explore this further, consider Definition 36.

Definition 36. *Let $\delta_o(u, v)$ denote the weight of a least overt path from u to v:*

$$\delta_o(u, v) = \min_{p \in P(u,v)} w_o(p).$$

We define $P_o(u, v)$ as the set of all possible paths of least overt weight between u and v, where:

$$P_o(u, v) = \{p \in P(u, v) | w_o(p) = \delta_o(u, v)\} \tag{5.10}$$

Similarly, we define $P_c(u, v)$ as the set of all possible paths of least covert weight between u and v:

$$P_c(u, v) = \{p \in P(u, v) | w_c(p) = \delta_c(u, v)\} \tag{5.11}$$

For a given vertex pair u and v, define the subset of minimum overt paths with fewest edges as $P_o^s(u, v) \subseteq P_o(u, v)$ where:

$$P_o^s(u, v) = \{p \in P_o(u, v) | l(p) = \Delta_s(P_o(u, v))\} \tag{5.12}$$

Similarly we define $P_c^s(G) \subseteq P_c(u, v)$ as the subset of these least covert paths that have the fewest edges, i.e:

$$P_c^s(u, v) = \{p \in P_c(u, v) | l(p) = \Delta_s(P_c(u, v))\} \tag{5.13}$$

5.2.3 Overt-Length and Covert-Length Trade-off

The difference between the length of a shortest path as compared to the minimum length of a least weight overt (or covert) path is of interest as it indicates the extent to which shortest paths through the graph are aligned to edges that can potentially forward content to the remaining members of an induced triad. We define the *overt-length trade-off* for a pair of vertices u and v in Definition 37:

Definition 37. *The overt-length trade-off for* u *and* $v \in V(G)$ *is defined by* $t_s^o(u, v)$*, where:*

$$t_s^o(u, v) = \Delta_s(P_o^s(u, v)) - \delta_s(u, v) \tag{5.14}$$

Therefore $t_s^o(u, v)$ represents the number of additional edges that results when prioritising overt edges. The covert-length trade-off is defined similarly and denoted by $t_s^c(u, v)$:

Definition 38. *The covert-length trade-off for* u *and* $v \in V(G)$ *is defined by* $t_s^c(u, v)$*, where:*

$$t_s^c(u, v) = \Delta_s(P_c^s(u, v)) - \delta_s(u, v) \tag{5.15}$$

Algorithm 11 highlights the function OVERTLENGTHTRADEOFF, which computes the overt-length trade-off $t_s^o(u, v)$. The algorithm for the function COVERTLENGTHTRADEOFF to compute the covert length trade-off $T_s^c(u, v)$ for u and v follow similarly.

Algorithm 11 OVERTLENGTHTRADEOFF(G, u, v): Computes $t_s^o(u, v)$.

Input: Graph G, $u, v \in V(G)$

Output: The overt length trade-off between u and v: $t_s^o(u, v)$.

1: $P_s(u, v) =$DIJKSTRA_ALL_PAIRS(G, u, v)

2: **for** $(x, y) \in E(G)$ **do** ▷ Calculate dummy weights

3: $\quad w_{(x,y)} = (x, y)_o(|V(G)| - 1) + 1$

4: $P_o^s(u, v) =$DIJKSTRA_ALL_PAIRS$(u, v, weights = w)$

5: $t_s^o(u, v) = l(p_2) - l(p_1)$ for any $p_1 \in P_s(u, v), p_2 \in P_o^s(u, v)$

6: **return** $t_s^o(u, v)$

In Algorithm 11, when computing the overt-length trade-off on vertices u and v, we compare a path of least length between u and v (i.e a path in $P_s(u, v)$) with a least overt path with the fewest edges (i.e a path in $P_o^s(u, v)$). To find a path in $P_o^s(u, v)$ we are optimising for two variables: finding least overt path with respect to minimising length. In order to do this we introduce dummy variables to weight edges when running Dijkstra's Algorithm.

Note that to optimise a a with respect to b in paired weight (a, b), edges must be weighted as: $aU_b + b$ where U_b denotes an upper bound on b. Similarly to optimise b with respect to a, edges must be weighted as $bU_a + a$ where U_a is an upper bound on a.

Thus, to find paths in $P_o^s(u, v)$ we must give edges the dummy weight $aU_b + b$ where a denotes the overt weight of an edge and b denotes the length of an edge. Then $a = w_o(x, y)$ and $b = 1$. An upper bound on path length is set as $U_b = |V(G)| - 1$. Hence $aU_b + b = (w_o(x, y)(|V(G)| - 1)) + 1$.

We can consider the overt-length trade off across all possible paths in a graph. We define the global overt-length trade-off in Definition 39.

Definition 39. *Let $C(G)$ be the set of all pairs of vertices of G connected by a path, i.e:*

$$C(G) = \{(u, v)|u, v \in V(G), \exists p(u, v)\} \tag{5.16}$$

Then we define the global average overt-length trade-off as:

$$T_s^o(G) = \frac{\sum_{(u,v)\in C(G)} t_s^o(u,v)}{|C(G)|} \tag{5.17}$$

The global average covert-length trade-off in path length is defined similarly and is denoted by $T_s^c(G)$.

Definition 40. *We define the global average covert-length trade-off as:*

$$T_s^c(G) = \frac{\sum_{(u,v)\in C(G)} t_s^c(u,v)}{|C(G)|} \tag{5.18}$$

We are particularly interested in the subset of paths that show an improvement in the shortest length when ignoring overt weight. We define the average improvement in edges (for overt length trade-off) in Definition 41.

Definition 41. *Let $C_s^o(G)$ denote the set of vertices where the overt-length trade-off is non-zero, i.e:*

$$C_s^o(G) = \{(u,v)|t_s^o(u,v) > 0, u, v \in V(G)\} \tag{5.19}$$

Then the average improvement in edges (for overt-length trade-off) is the normalised sum of the non-zero overt-length trade-offs, which is given by:

$$I_s^o(G) = \frac{\sum_{(u,v)\in C_s^o(G)} t_s^o(u,v)}{|C_s^o(G)|} \tag{5.20}$$

The average improvement in edges for covert-length trade-off, $I_s^c(G)$, is defined similarly.

Definition 42. *Let $C_s^c(G)$ denote the set of vertices where the covert-length trade-off is non-zero, i.e:*

$$C_s^c(G) = \{(u,v)|t_s^c(u,v) > 0, u, v \in V(G)\} \tag{5.21}$$

Then the average improvement in edges (for covert-length trade-off) is the normalised sum of the non-zero covert-length trade-offs, which is given by:

$$I_s^c(G) = \frac{\sum_{(u,v)\in C_s^c(G)} t_s^c(u,v)}{|C_s^c(G)|} \tag{5.22}$$

The function to determine $I_s^o(G)$ is highlighted in Algorithm 12. The function to determine $I_s^c(G)$ follows similarly.

Algorithm 12 AVERAGEIMPROVEMENTOVERTLENGTH(G): Computes the average improvement in edges $I_s^o(G)$.

Input: Graph G

Output: The average improvement in edges $I_s^o(G)$

$c = 0$ ▷ Count improvements
$T = 0$ ▷ Total improvement
for $u, v \in C(G)$ **do** ▷ Loop over all connected pairs
    $t_s^o(u,v) =$OVERTLENGTHTRADEOFF(u,v)
    if $t_s^o(u,v) > 0$ **then** ▷ Non-zero trade-off
        $c = c + 1$
        $T = T + t_s^o(u,v)$
$I_s^o(G) = \frac{T}{c}$
return $I_s^o(G)$

5.2.4 Length-Overt Trade-off

Alternative to considering the differences in path length due to minimising overt and covert centrality, we can also consider the differences in overt and covert weight, when comparing optimised weight paths against the least weight of a shortest path.

We define the *length-overt trade-off* for a pair of vertices u and v in Definition 43.

Definition 43. *The length-overt trade-off for* u *and* $v \in V(G)$ *is defined by* $t_o^s(u,v)$*, where:*

$$t_o^s(u,v) = \Delta_o(P_s^o(u,v)) - \delta_o(u,v) \tag{5.23}$$

That is, $t_o^s(u,v)$ represents the increase in overt weight that results when prioritising shortest paths. Similarly, the *length-covert trade-off* in weight between a pair of vertices is denoted by $t_c^s(u,v)$.

Definition 44. *Then the length-covert trade-off for* u *and* $v \in V(G)$ *is defined by* $t_c^s(u,v)$*, where:*

$$t_c^s(u,v) = \Delta_c(P_s^c(u,v)) - \delta_c(u,v) \tag{5.24}$$

Algorithm 13 highlights the function LENGTHOVERTTRADEOFF, which computes the length-overt trade-off $t_o^s(u,v)$. When computing the length-overt trade-off on vertices u and v we compare a least overt path between u and v (i.e a path in $P_o(u,v)$) with a shortest path between u and v with the least overt weight edges (i.e a path in $P_s^o(u,v)$). Using the same notation from Algorithm 11 to find paths in $P_s^o(u,v)$ we must minimise path overtness with respect to length. Letting a denote the overt weight of an edge and b denote the length of an edge, we must weight edges with the dummy variable $bU_a + a$ under Dijkstra's Algorithm to compute $P_s^o(u,v)$. An upper bound on overt path weight $U_a = (|V(G)| - 1)(|V(G)| - 2) = |V(G)|^2 + 3|V(G)| + 2$. The algorithm for covert follows similarly.

Algorithm 13 LENGTHOVERTTRADEOFF(G, u, v): Computes $t_s^o(u, v)$.

Input: Graph G, $u, v \in V(G)$

Output: The length overt trade-off between u and v: $t_o^s(u, v)$.

for $(x, y) \in E(G)$ **do**

    $v_{(x,y)} = w_o(x, y)$

$P_o(u, v) =$DIJKSTRA_ALL_PAIRS$(u, v, weights = v)$

for $(x, y) \in E(G)$ **do** ▷ Calculate dummy weights

    $w_{(x,y)} = |V(G)|^2 - 3|V(G)| + 2 + w_o(x, y)$

$P_s^o(u, v) =$DIJKSTRA_ALL_PAIRS$(u, v, weights = w)$

$t_o^s(u, v) = w_o(p_1) - w_o(p_2)$ for any $p_1 \in P_s^o(u, v), p_2 \in P_o(u, v)$

return $t_s^o(u, v)$

As for path length, we also consider trade-off at a graph level. We define the average improvement in edges (for length-overt trade-off) in Definition 45:

Definition 45. *Let $C_o^s(G)$ denote the set of vertices where the length-overt trade-off is non-zero, i.e*

$$C_o^s(G) = \{(u, v) | t_o^s(u, v) > 0, u, v \in V(G)\} \tag{5.25}$$

The average improvement in edges (for length-overt trade-off) is then given by:

$$I_o^s(G) = \frac{\sum_{(u,v) \in C_o^s(G)} t_o^s(u, v)}{|C_o^s(G)|} \tag{5.26}$$

We define the average improvement in edges (for length-covert trade-off) $I_c^s(G)$ similarly, where the average improvement in edges represents the mean non-zero length-covert trade-off:

Definition 46. *Let $C_c^s(G)$ denote the set of vertices where the length-covert trade-off is non-zero, i.e*

$$C_c^s(G) = \{(u, v) | t_c^s(u, v) > 0, u, v \in V(G)\} \tag{5.27}$$

The average improvement in edges (for length-overt trade-off) is then given by:

$$I_c^s(G) = \frac{\sum_{(u,v)\in C_c^s(G)} t_c^s(u,v)}{|C_c^s(G)|} \tag{5.28}$$

The functions to compute $I_o^s(G)$ and $I_c^s(G)$ follow similarly to Algorithm 12.

5.2.5 Summary of Notation

In Table we summarise the notation used in Section 5.2. We should note that overt-length trade-off refers to the additional edges that result when prioritising overt edges, whilst length-overt trade-off refers to the increase in overt weight that results when prioritising shortest paths. Covert follows similarly.

Name	Notation
Overt weight of a path p	$w_o(p)$
Total covert weight of a path p	$w_c(p)$
Overt-length trade-off for u and v	$t_s^o(u,v)$
Covert-length trade-off for u and v	$t_s^c(u,v)$
Length-overt trade-off for u and v	$t_o^s(u,v)$
Length-covert trade-off for u and v	$t_c^s(u,v)$
Length-covert trade-off for u and v	$t_c^s(u,v)$
Average improvement in edges for overt-length trade-off	$I_s^o(G)$
Average improvement in edges for covert-length trade-off	$I_s^c(G)$
Average improvement in edges for length-overt trade-off	$I_o^s(G)$
Average improvement in edges for length-covert trade-off	$I_c^s(G)$

Table 5.1: A summary of the notation used in Section 5.2

5.3 Results

To understand how potential graph structure and edge density affect the presence and trade-offs concerning overt centrality, covert centrality and shortest paths, we assess synthesised networks having particular properties.

We construct directed Erdos-Renyi networks [18] (ER networks) with uniform random selection of edges and compare them against networks constructions designed to obtain long tailed distributions (random k-out and scale free network constructions originating from networkX [33]).

ER networks are random models, where we can control for density and number of nodes. Random k-out and scale free networks may more closely represent phenomena in real life, but we cannot control for larger densities. We acknowledge that these properties are typically employed in very large-scale networks (i.e., high number of vertices), however in order to maintain feasibility of run time for connectivity analysis while varying density, we control the number of vertices for each network. This also enables networks to be human interpret-able, and allows us to assess a considerable sample of networks (e.g., average statistics over 100 networks) when considering results.

For proof of concept purposes, we consider networks with either $25, 30, 35, 40, 45$ or 50 vertices, allowing shortest path calculations to be frequently repeated. For random k-out and scale free constructions, multi-edges are removed. Random k-out network exhibits preferential attachment. We input the initial weight of each vertex, α, then out-degree of each node, k, and the number of nodes. Nodes u with degree less than k are chosen at random, and an edge is plotted from u to node v, where v is chosen with probability proportional to its weight. The weight of a node increases as its in-degree increases. Thus, the higher the weight of a node, the more likely there is an edge directed into a node. For the random k-out construction, we fix the number of edges directed out of each vertex ($k = 3$) and vary the initial weight of each vertex ($0.1 \leq \alpha \leq 1$). Scale free networks are built on four variables: the number of nodes, α, β and γ. α and γ control for the probability of the addition of nodes connected to existing ones within the network: α according to the in-degree distribution and γ according to the out-degree distribution. β controls for the probability of adding an edge between an existing pair of nodes according to the in and out-degree distribution of the pair of nodes. For the scale free construction, we vary the probability of adding an edge between two

existing vertices ($0.9 \leq \beta \leq 1.0$). The construction parameters α, γ are set in the range $0.0 \leq \alpha \leq 0.1$ and $0.0 \leq \gamma \leq 0.1$.

We emphasise that the scale free network constructions are applied here as a generative alternative to random networks, and their small $|V(G)|$ limits the presence of statistically significant long-tail properties, but the construction is a useful alternative case study for relatively small numbers of vertices, as compared to ER networks. In each case we vary the density (directly for ER networks, indirectly for random k-out and scale free networks). For each density level, network size and network type we generate 100 network instances and consider the average value of $I_s^o(G)$, $I_s^c(G)$, $I_o^s(G)$ and $I_c^s(G)$.

Unfortunately we are unable to directly control density in random-k and scale free networks, therefore results produced may relate to a single, outlying network with unique density. This is unlike ER networks, where we can control density directly and therefore average over repeated instances of ER networks on the same density. Therefore, results in random-k and scale free networks are inconclusive, as we will see in Section 5.3.3.

5.3.1 Observations on Increasing Edge Density

Before we discuss results, we first highlight how a change in network density can affect the overtness/covertness of a path as a means to assess results concerning $I_s^o(G)$, $I_s^c(G)$, $I_o^s(G)$ and $I_c^s(G)$ across varying densities in the various generated networks. $I_s^o(G)$ and $I_s^c(G)$ respectively denote the average non-zero overt-length and covert-length trade-off for G (i.e the number of additional edges that result when prioritising minimising overtness/covertness over path length), whilst $I_o^s(G)$ and $I_c^s(G)$ respectively denote the average non-zero length-overt and length-covert trade-off for G (i.e the decrease in path overt/covert weight when prioritising minimising overtness/covertness over path length). This is useful for interpreting what happens when edge density is changed.

Suppose there exists an edge $(u, v) \in E(G)$. There are three ways to increase the covert

weight of this edge:

1 Add an edge (u_1, u) for some $u_1 \in V(G)$.

2 Add an edge (u, u_2) for some $u_2 \in V(G)$.

3 Add an edge (u_3, v) for some $u_3 \in V(G)$.

This compares to one way to increase the overt weight of the same edge, that is:

4 Add the edge (v, u_4) for some $u_4 \in V(G)$.

Increasing the overt/covert weight of the edge (u, v) simultaneously increases the overt/covert weight of any path that (u, v) is contained in. In Example 5.3.1, we consider the effect of adding an edge to an existing network in order to increase the overt/covert weight of an edge in a path, and observe its affect on the total weight of path p.

Example 5.3.1. *Suppose there exists a path p in network G between two vertices, and (u, v) is an edge in this path. Suppose there are a further two vertices: $u_1, u_2 \in V(G)$ such that (u_1, u) and (v, u_2) are edges in p, as shown in Figure 5.2.*

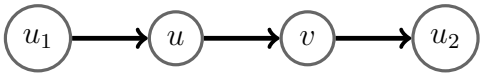


Figure 5.2: Path $p = (u_1, u, v, u_2)$.

To increase the overt weight of (u, v) we follow operation (4) outlined above: adding the edge (v, u_3) for some $u_3 \in V(G)$, as demonstrated by the red edge in Figure 5.3. Adding (v, u_3) to the network forms the triad $t_1 = (u, v, u_3)$ of type `021C` in which (u, v) is overt; increasing the overt weight of (u, v), and therefore $w_o(p)$, by one. In addition, the triad $t_2 = (v, u_2, u_3)$ of type `021D` is formed, wherein the edge (v, u_2) is covert, hence the $w_c(p)$ is increased by one.

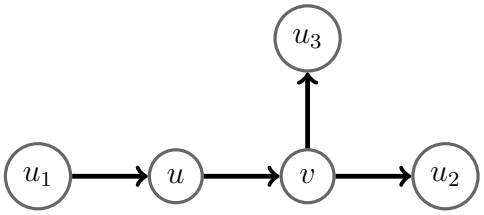


Figure 5.3: Path p with the addition of edge (v, u_3) in red.

To increase the covert weight of (u, v), we follow operations (1)-(3) outlined above. Hence, we add the edge either (u_4, u), (u, u_5) or (u_6, v) for some $u_4, u_5, u_6 \in V(G)$, as demonstrated in Figure 5.4 by adding the blue edges. Adding the edge (u_4, u) forms the triad $t_3 = (u_4, u, v)$ of type `021C` *in which (u, v) is covert. This also forms the triad $t_4 = (u_1, u, u_4)$ of type* `021U` *in which (u_1, u) is covert. Thus, adding (u_4, u) to the network increases the $w_c(p)$ by two. Adding the edge (u, u_5) forms the triad $t_5 = (u_5, u, v)$ of type* `021D` *in which (u, v) is also covert. Adding this edge also forms the triad $t_6 = (u_1, u, u_5)$ of type* `021C` *in which (u_1, u) is overt. Thus, adding (u, u_5) to the network increases $w_c(p)$ by one and the $w_o(p)$ by one. Finally, adding (u_6, v) to the network increases the $w_o(p)$ by two as it forms the triad $t_7 = (u, v, u_6)$ of type* `021U` *in which (u, v) is covert and the triad $t_8 = (u_6, v, u_2)$ of type* `021C` *in which (v, u_2) is covert.*

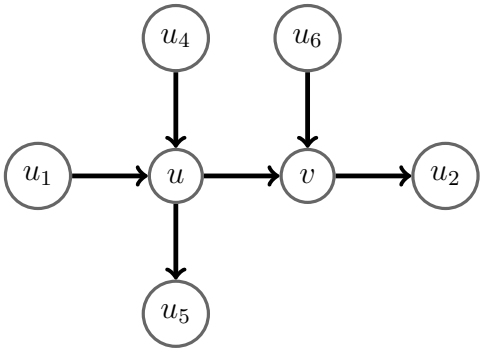


Figure 5.4: Path p with the addition of (u_4, u), (u, u_5) or (u_6, v).

Therefore, increasing the overt weight of (u, v) increases the overt and covert weight of p by one. Increasing the covert weight of (u, v) may increase the covert weight of p by two (if operations (2) and (3) are followed) or increases the overt and covert weight of

p by one (if operation (1) is followed). These results are dependent on the existence of edges (u_1, u) and (v, u_2); as well as the possible triads that can form between vertices.

On first consideration, it would appear that there are more ways to increase the covert weight of the path than the overt weight through operations (1)-(4). However, it is important to note that adding an edge to a network forms new triads or changes existing ones. The new edge can be contained simultaneously in multiple triads that share between zero and two edges with p. The likelihood of the class of triad the new edge can be contained may be affected by network density.

We predict that, subject to restrictions on possible triad classes which can occur at varying densities, carrying out operations (1)-(4) affects the overtness/covertness of paths in different ways. We also predict these rules affect the ability to perform operations (1)-(4). We show this in Example 5.3.2.

Example 5.3.2. *Recall Figure 5.2. Suppose we increase $w_c(p)$ by two by adding the edge (u_4, u) to the network, for some $u_4 \in V(G)$ (as in Figure 5.4). (u_4, u) is contained in the triad $t_1 = (u_4, a, b)$ of type* `021C`*. Suppose we now insist that this triad is denser, and must be of type* `111U`*. Then there must be an additional edge (u, u_4) as shown in Figure 5.5.*

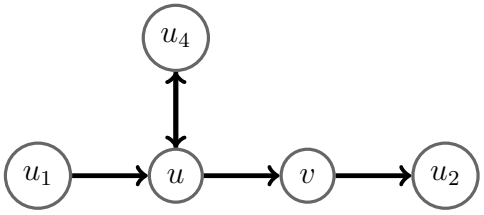


Figure 5.5: Path p with the addition of (u_4, u) and (u, u_4) in blue.

Now, (u, v) remains covert in triad t_1. However, (u_1, u) is now overt in triad $t_2 = (u_1, u, u_4)$ which has changed from `021U` *to* `111D`*, so $w_c(p)$ is reduced, whilst the $w_o(p)$ is increased.*

Therefore we state our first assumption:

- **A1:** When modifying an existing network and thereby changing the frequency of occurrence of each class of induced triad we change and restrict the ability to increase the overt and covert weight of a path p ($w_o(p)$ and $w_c(p)$ respectively). When sparser triads are induced (triads constructed on two edges) there are more ways to increase $w_c(p)$ than $w_o(p)$. As denser triads are induced (triads that contain at least three edges), the number of ways to increase $w_o(p)$ grows and overtakes the number of ways to increase $w_c(p)$; and $w_c(p)$ may even reduce.

In the following sections, we consider the results by network-type.

5.3.2 ER networks

We now discuss the various results on ER networks. We first look to ER networks (Between 25 and 50 vertices); first comparing density with $I_s^o(G)$, $I_s^c(G)$, $I_o^s(G)$ and $I_c^s(G)$ in Figure 5.6. We then discuss additional analyses on these small networks in Figures 5.7 - 5.13 and Tables 5.2 - 5.5. We then look to large ER networks in Figure 5.14.

ER networks

Figure 5.6 compares (in a clockwise order) $I_s^o(G), I_s^c(G), I_o^s(G)$ and $I_o^s(G)$ in ER networks (number of vertices ranges between 25 and 50 vertices in increments of 5) on varying density (density ranges between 0 and 1 in increments of 0.05). Values are averaged over 50 samples of each network.

In Figure 5.6 we see that an increase in network density leads to a decrease in $I_s^o(G)$. An increasing number of vertices present in the network decreases the maximum value $I_s^o(G)$ can be, causing a more rapid decline in $I_s^o(G)$ and no further results are produced at an earlier density. For example, in ER networks on 25 vertices, the maximum $I_s^o(G)$ value is 1.24 and no further results are produced at 0.55 density, whereas in ER networks

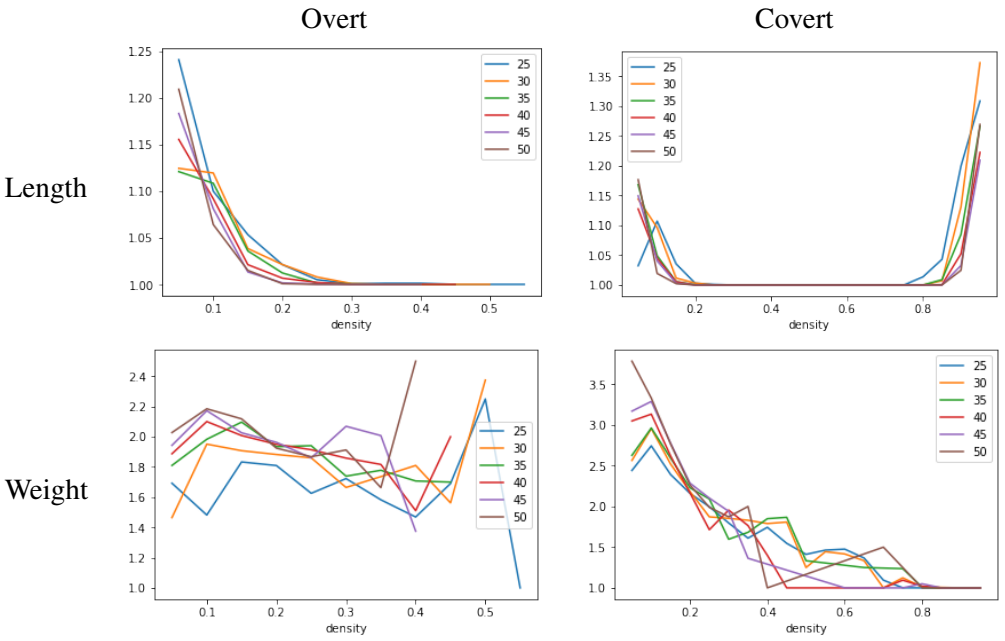


Figure 5.6: The average values (in a clockwise order) of $I_s^o(G)$, $I_s^c(G)$, $I_c^s(G)$ and $I_o^s(G)$, sampling over 50 instances of an ER-network.

on 35 vertices, the highest $I_s^o(G)$ value is 1.12 and no further results are produced at 0.45 density. The highest $I_s^o(G)$ value in all these ER networks is found in networks on very low density (0.05). $I_s^o(G)$ in all networks has plateud at 0.3 density (0.2 density in the largER networks), and no $I_s^o(G)$ value occurs in networks with greater than 0.55 density (which decreases to 0.4 density in networks on a greater number of vertices).

Conversely, an increase in network density leads to an increase in $I_s^c(G)$. We observe in 5.6 that at 0.05 density, $I_s^c(G)$ hovers around 1.15, which then decreases as density increases to 0.2. $I_s^c(G)$ stays fixed until approximately 0.8 density, at which point we see a rapid increase in $I_s^c(G)$ until 0.95 density is reached.

$I_o^s(G)$ does not necessarily follow a similar pattern to $I_s^o(G)$: whilst some networks see a decrease in $I_o^s(G)$ as density increases, others see a spike in cases in higher densities (such as the largest $I_o^s(G)$ value in networks on 50 vertices occurs around 0.35 density). Further $I_o^s(G)$ values tend to be greater than $I_s^o(G)$ values. Conversely to $I_s^o(G)$; an increasing network size (in terms of number of vertices) relates to a slight increase

in $I_o^s(G)$. $I_c^s(G)$ generally decreases as density increases, although this progression is jagged in its appearance.

To analyse the results in 5.6 we first make the following assumptions:

- **A2:** Changing network density affects its triad census through modifying existing triads, so that the higher the density of the network, the greater the number of denser triads contained in the network and the fewer the number of sparser triads.

To back-up assumption **A1**, we investigate the average volume of occurrence of each triad class through taking a triad census on ER networks sampled on 0.1, 0.5 and 0.9 density. We generate 50 iterations of ER network on each density and take the mean. The results for this are provided in Table 5.2. We observe that at first exploration, **A1** appears to hold true. For example, ER networks on 0.1 density (the lowest density we sample) contain a majority of triads of class `021D,021U` and `021C`, all triads constructed on two edges; whilst maintaining very few triads constructed on three or more edges. ER networks sampled on 0.9 density (the highest density we sample) contain a majority of triads of class `120C` (four edges), `210` (five edges) and `300` (six edges), whilst also maintaining a high proportion of triads of class `201,120D` and `120U` (all constructed on four edges). ER networks on 0.5 density are more balanced, they contain a number of sparse triads (such as `021D,021U` and `021C`) and dense triads (such as `120C,210` and `300`; whilst maintaining high volumes of triads of density somewhere in between (such as `111D,111U,030T`, all constructed on three edges).

We next propose our third assumption, which assumes through **A2** that network density induces a greater volume of denser triads and a lower volume of sparser triads, which in turn affects the overt/covert centrality of the network's edges:

- **A3:** Increasing the density of an empty network at first increases the overt and covert centrality of its edges. As density increases further, the rate at which the

	021D	021U	021C	111D	111U	030T	030C	201	120D	120U	120C	210	300
ER (0.1) Mean	84.4	86.7	167.5	19.1	19.3	20.5	5.9	1.3	1.2	1.1	2.1	0.2	0.0
ER (0.1) Std. Dev.	37.9	39.4	76.1	13.8	13.3	11.2	4.6	1.5	1.6	1.4	2.0	0.5	0.0
ER (0.5) Mean	184.3	178.4	359.7	398.5	405.0	393.8	131.6	222.7	219.2	216.8	438.0	497.0	94.5
ER (0.5) Std. Dev.	80.3	78.9	160.6	167.3	167.1	174.1	59.1	98.2	91.0	89.9	182.4	226.5	51.1
ER (0.9) Mean	0.8	1.2	2.3	20.7	19.4	19.7	7.0	86.6	87.2	87.0	173.2	1539.6	2256.1
ER (0.9) Std. Dev.	0.8	1.1	2.0	11.0	9.1	9.9	3.9	45.3	42.6	35.7	74.2	624.9	938.0
Random k (0.1) Mean	11.0	363.3	44.0	49.7	1.5	20.4	1.0	1.0	13.7	0.2	1.5	0.8	0.2
Random k (0.1) SD	8.2	123.5	28.7	21.6	2.1	9.8	1.4	1.1	7.9	0.4	1.4	1.0	0.4
Random k (0.5) Mean	35.3	198.2	116.2	36.8	7.1	22.7	3.9	1.0	4.0	0.7	2.2	0.5	0.0
Random k (0.5) SD	13.7	62.1	43.5	12.5	4.9	7.4	2.3	1.0	2.6	0.7	1.3	0.7	0.0
Random k (0.9) Mean	41.1	151.4	128.8	38.0	9.9	17.9	3.8	1.4	3.0	0.7	2.5	0.5	0.0
Random k (0.9) SD	13.0	45.0	40.5	15.5	4.0	6.1	2.3	1.5	2.9	1.0	1.4	0.6	0.0
Scale Free (0.91) Mean	103.7	58.7	88.7	25.0	33.6	8.7	0.0	1.4	4.5	8.8	0.7	1.8	1.0
Scale Free SD	57.3	30.9	42.6	10.3	13.0	7.0	0.0	1.7	2.9	5.3	0.9	1.6	1.0
Scale Free (0.95) Mean	91.6	67.0	97.5	29.2	35.2	7.6	0.0	1.3	5.6	11.6	0.6	1.5	1.8
Scale Free (0.95) SD	41.4	39.6	40.2	16.1	16.6	6.7	0.0	2.4	3.3	5.4	0.8	1.6	1.3
Scale Free (0.99) Mean	102.9	59.2	108.1	33.6	42.4	3.9	0.0	1.8	5.7	11.0	0.9	2.1	2.1
Scale Free (0.99) SD	52.7	34.9	49.4	16.0	18.6	4.3	0.0	2.7	4.0	6.2	0.9	1.7	1.9

Table 5.2: Triadic census over all generated networks, across specific densities. ER is sampled over $p = 0.1, 0.5$ and 0.9. Random k is sampled over $\alpha = 0.1, 0.5$ and 0.9. Scale free is sampled over $\beta = 0.91, 0.95$ and 0.99, with α and $\gamma = \frac{1-\beta}{2}$ in each case.

> overt centrality of edges increases speeds up, whilst the rate of which the covert centrality of edges increases slows down and eventually the covert centrality of edges decreases.

To evidence this, we observe the number of overt and covert edges present across all 13 triads classes (see Figure 4.3 in Section 4.2). There are three triads on two edges: `021D`, `021U` and `021C`. Of these triads there are five covert edges and one overt edge. There are four triads on three edges: `111D`, `111U`, `030T` and `030C`. Of these triads there are six overt edges and six covert edges. There are four triads on four edges: `201`, `120D`, `120U` and `120C`, amongst which there are nine overt edges and seven covert edges. `210` is the only triad on five edges, which contains one covert and four overt edges; and `300`, the only triad on six edges, contains only overt edges. Therefore, on triads containing two edges, it is more likely an edge is covert than overt. In triads containing three edges there is an equal likelihood an edge is overt as covert. In triads above three edges, there is an increasing likelihood an edge is overt than covert.

We also must note it is possible for covert edges to be transformed into overt edges

but not visa versa. Therefore the covert centrality of an edge can decrease through transforming its behaviour from covert to overt in the triads it is contained in. This is not true for overt: the overt centrality of an edge can only stay the same or increase.

Finally, we are in a position to make our final assumption. Through **A3** we have assumed that increasing the density of a network changes the overt and covert centrality of its edges. However, we are interested in how changing the overt and covert centrality of edges of a network affect the overt/covert weight of paths ($w_o(p)$ and $w_c(p)$ respectively), in order to explore the overt-length/covert-length ($t_s^o(u,v)$ and $t_s^c(u,v)$) and length-overt/length-covert ($t_o^s(u,v)$ and $t_c^s(u,v)$) trade-offs. In Section 5.3.1 we explore how modifying the overt/covert centrality of an edge affect $w_o(p)$ and $w_c(p)$, concluding with the assumption ($A1$) that restrictions in the network on the frequency of occurrence of each class of triad affect the ability to increase $w_o(p)$ and $w_c(p)$. We now make our final assumption:

- **A4:** Increasing the density of an empty network at first increases $w_o(p)$ and $w_c(p)$, with the initial increase in $w_c(p)$ being greater than the initial increase in the $w_o(p)$. As density increases further, the rate at which $w_o(p)$ increases speeds up, whilst the rate of which $w_c(p)$ increases slows down and eventually $w_c(p)$ decreases.

These assumptions are important because each assumption is used to build the following assumption, resulting in assumption **A4**. In Figure 5.6 we repeatedly compare least length paths with least overt/covert paths. **A4** indicates that network density affects $w_o(p)$ and $w_c(p)$ of both least length and least overt/covert paths, and thereby $t_s^o(u,v)$,$t_s^c(u,v)$ and $t_o^s(u,v)$, $t_c^s(u,v)$. Crucially, we predict an increase in density strengthens the relationship between path length and $w_o(p)$, whilst weakening the relationship between path length and $w_c(p)$.

Increased $w_o(p)$ is derived from increased overt weight of edges within them. This means that $w_o(p)$ is directly influenced by $l(p)$. Therefore to minimise $w_o(p)$, the

number of edges in a path must be minimised (i.e. $l(p)$). This results in the difference in length between a minimum overt and a minimum length path decreasing, thus the overt-length trade-off for a pair of vertices (t_s^o) decreases. After a density of approximately 0.6, accordingly we observe the least weighted path becoming a shortest path, hence no further results are produced.

In contrast, we expect a reduction in $w_c(p)$ as density increases; although it's plausible from Figure 5.6 to expect an initial increase in $w_c(p)$ at low densities (between 0.05 and 0.2). A reduction in $w_c(p)$ weakens the relationship between $l(p)$ and $w_c(p)$, giving the potential for minimum covert paths to be longer. Unlike overt paths, it is possible to construct weightless covert paths (an example is given in Example 5.3.3):

Example 5.3.3. *Suppose there exists a path* $p = (u_1, u_2, u_3, u_4, u_5, u_6)$. *Then the blue edges in Figure 5.7 ensure that* $w_c(p) = 0$.

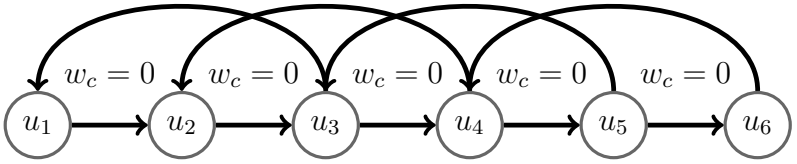


Figure 5.7: Example path $p = (u_1, u_2, u_3, u_4, u_5, u_6)$ **where** $w_c(p) = 0$**.**

So, from the above we can expect the overt weight of a path, $w_o(p)$, to increase as density increases in the network. This can occur in a minimum length path (increasing the length-overt trade-off $t_o^s(u, v)$ between a minimum overt path and a path on minimum edges), a minimum overt path (decreasing the length-overt trade-off between a minimum overt path and path on minimum edges) or both. Furthermore, either increase in weight of a minimum edge or minimum overt path could force the algorithm to choose another path. Therefore, $t_o^s(u, v)$ could stay the same could stay the same, increase or decrease. Therefore, overall there is no general trend for $I_o^S(G)$: in Figure 5.6 we see no general correlation between the difference in overt weight and edge density. We observe that this occurs up to a density of 0.6, where afterwards least weighted overt paths and shortest

paths are the same and no further results occur. At the same time, we observe a negative correlation between edge density and length-covert trade-off between a shortest path and a path that minimises covert weight ($I_s^c(G)$). This is due to covert paths getting lighter and lighter as edge density increases in ER networks, meaning that there is less scope for a difference in weight (which may possibly be zero).

Additional Results on ER networks

To gain further insight into $I_s^o(G)$, $I_s^c(G)$, $I_o^s(G)$ and $I_c^s(G)$, additional analysis is undertaken on ER networks. Firstly, we compare network with density with proportion of paths in the network where there is a difference between a minimum overt/covert path and shortest path (see Figures 5.8 and 5.12). Next, we compare the frequency of trade-offs $t_o^s, t_c^s, t_s^o, t_s^c$ across the networks that are averaged to provide $I_s^o(G)$, $I_s^c(G)$, $I_o^s(G)$ and $I_c^s(G)$ (see Figures 5.9, 5.10, 5.13, and 5.14). Finally, we compare $I_s^o(G)$, $I_s^c(G)$, $I_o^s(G)$ and $I_c^s(G)$ against the corresponding average weight and length of shortest paths in the network (see Figure 5.11).

In Figure 5.8 we counted the total number of least overt paths which are not shortest paths occurring at each vertex density pair across all 50 simulations of this vertex-density pair. We then calculate the proportion of these paths of the total number of such paths at each vertex-density pair.

Figure 5.8 demonstrates that the highest volume least overt paths which are not themselves shortest paths (i.e, the paths yielding non-zero overt-length and length-overt trade-offs) occur when the density of the network is between 0.1 and 0.25 in networks on 25 vertices; between 0.05 and 0.2 in networks on between 30 and 40 vertices; and between 0.05 and 0.15 in networks on between 45 and 50 vertices. The peak volume of these paths occurs in at 0.15 density in networks on 25 vertices; and 0.1 density in networks on between 30 and 50 vertices. In all cases, low densities indicate a greater volume of least overt paths of interest. Moreover, we show that these paths occur

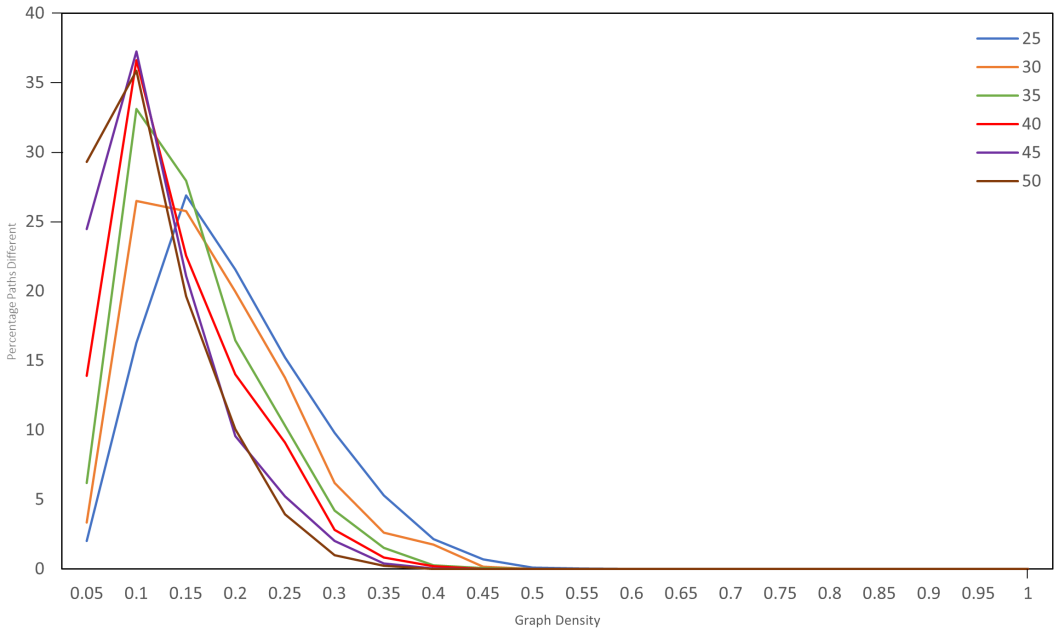
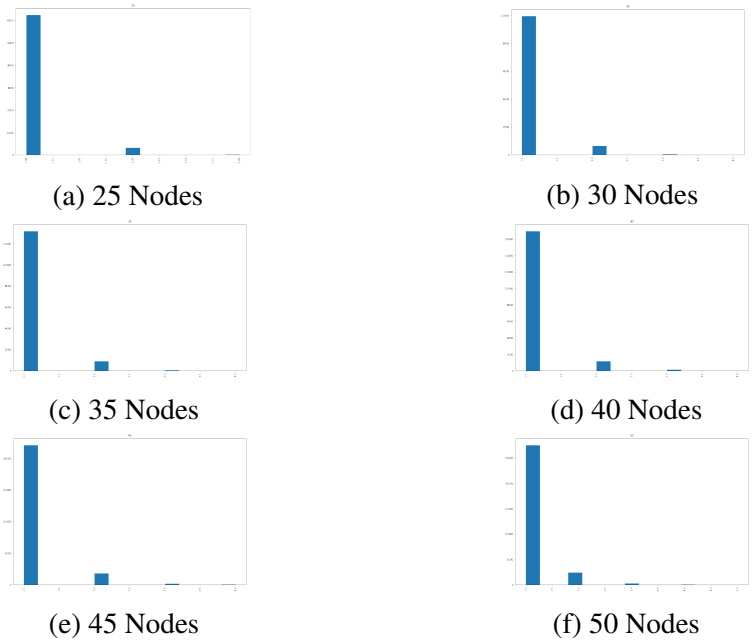


Figure 5.8: Percentage of all least overt paths with non-zero t_s^o and t_o^S that occur within one of 50 simulations at each vertex-density pair.

frequently, for instance: in networks on 50 vertices and 0.1 density, 10843 such paths occur.

In Figure 5.9 we count the frequency of length-overt trade-offs (i.e t_s^o) across all least overt paths from every simulation we ran to compute the results in Figure 5.6. We plot the frequency of each t_s^o value in the histograms, which are separated by network size. Table 5.3 counts the proportion of t_s^o out the total number of trade-offs against each network size, and tells us the values for the bars in the histrograms in Figure 5.9.

We repeat this process with length-overt trade-offs (i.e t_o^s) in Figure 5.10 and Table 5.4.



(a) 25 Nodes (b) 30 Nodes

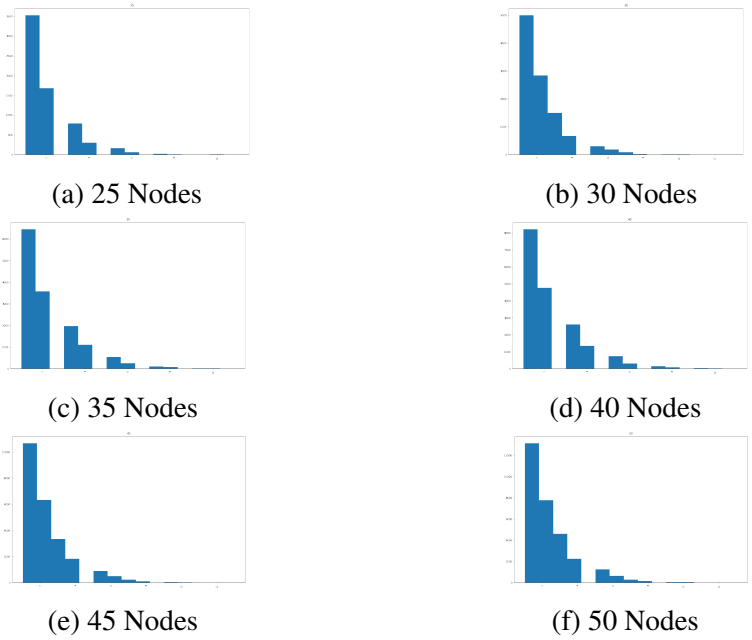
(c) 35 Nodes (d) 40 Nodes

(e) 45 Nodes (f) 50 Nodes

Figure 5.9: Frequency histograms plotting occurrence of t_s^o values, separated by number of vertices in the network.

Number of vertices	1	2	3	4	5
25	95.00	4.82	0.18	0.00	0.00
30	93.62	5.99	0.38	0.02	0.00
35	93.31	6.37	0.31	0.01	0.00
40	92.94	6.26	0.75	0.05	0.00
45	91.89	7.41	0.63	0.07	0.00
50	91.06	7.97	0.87	0.11	0.00

Table 5.3: Percentage of all least overt paths with non-zero t_s^o for each network size that have the corresponding t_s^o value, corresponding to Figure 5.9.



(a) 25 Nodes (b) 30 Nodes

(c) 35 Nodes (d) 40 Nodes

(e) 45 Nodes (f) 50 Nodes

Figure 5.10: Frequency histograms plotting occurrence of t_o^s values, separated by number of vertices in the network.

Number of Nodes	1	2	3	4	5	6	7	8	9	10	11	12	13
25	53.78	25.61	12.00	4.66	2.52	0.91	0.37	0.08	0.00	0.08	0.00	0.00	0.00
30	47.02	26.73	14.09	6.31	2.86	1.73	0.83	0.21	0.13	0.07	0.03	0.00	0.02
35	45.73	25.35	13.95	7.86	3.85	1.82	0.72	0.51	0.13	0.06	0.01	0.00	0.00
40	45.03	26.09	14.31	7.35	4.03	1.72	0.79	0.42	0.16	0.09	0.01	0.00	0.00
45	44.44	26.40	13.90	7.64	3.76	2.09	0.99	0.46	0.19	0.06	0.03	0.02	0.01
50	43.49	25.70	15.25	7.45	4.12	2.16	0.93	0.51	0.20	0.16	0.03	0.00	0.00

Table 5.4: Percentage of all least overt paths with non-zero t_o^s for each network size that have the corresponding t_o^s value, corresponding to Figure 5.10.

Figure 5.9 and Table 5.3 shows that most t_s^o values are one (an overwhelming 91-95 percent of length trade-offs are one in networks on 25-50 vertices). There is a minor decrease in the percentage of paths where the t_s^o equals one as the number of vertices in the network increases, and a tendency to induce more paths where t_s^o is greater than one; although the volume is negligible.

In terms of weight, from Figure 5.10 and Table 5.4, whilst the greatest population of t_o^s values is still one, there is a much greater range than found in Figure 5.9; and the

percentage of t_o^s values equal to one is much smaller (between 43 and 53 percent in networks on 25-50 vertices).

In Figure 5.11 we plot $I_s^o(G), I_o^s(G), I_c^s(G), I_c^s(G)$ from Figure 5.6 against the corresponding length and weight of shortest paths that induced these values.

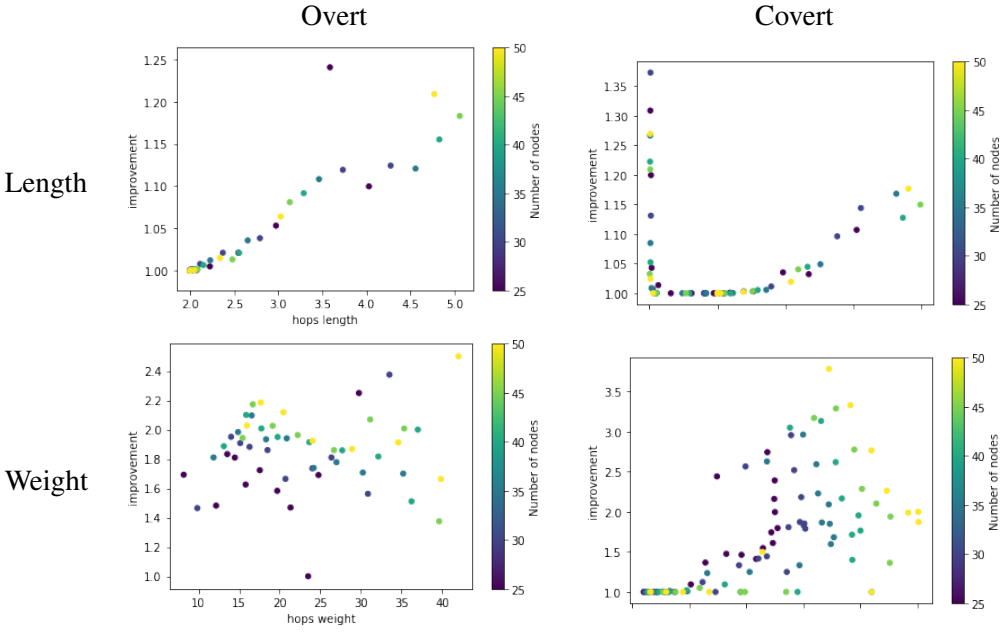


Figure 5.11: The average values of $I_s^o(G), I_s^c(G), I_c^s(G)$ and $I_o^s(G)$ (in clockwise order) against the corresponding length/weight of the shortest path, sampling over 50 instances of an ER-network.

Figure 5.11 shows that the highest $I_s^o(G)$ values (which come from networks on a low density from Figure 5.6) occur when a shortest path has between 4-5 hops. Figure 5.11 also shows that shorter shortest paths tend to induce smaller $I_s^o(G)$, and thus occur at higher densities. However, when there is the highest volume of least overt paths that induce a overt-length tradeoff from Figure 5.8 (i.e, at 0.15 density in networks on 25 vertices and 0.1 density in networks on between 30 and 50 vertices) shortest paths have between 3 and 4 hops.

Meanwhile, the highest $I_o^s(G)$ values occur when the overt weight of a shortest path is between 30 and 45, although there are many shortest paths of similar weight that do not induce a high $I_o^s(G)$. At densities which induce the highest volume of least overt

paths that induce a length-overt tradeoff, the path weights of the shortest paths cluster together in Figure 5.11, and are between 12 and 20 total overt weight.

In Figure 5.12 we ran the same process as in Figure 5.8 but this time with least covert paths which are not shortest paths and therefore yielding non-zero covert-length and length-covert trade-offs.

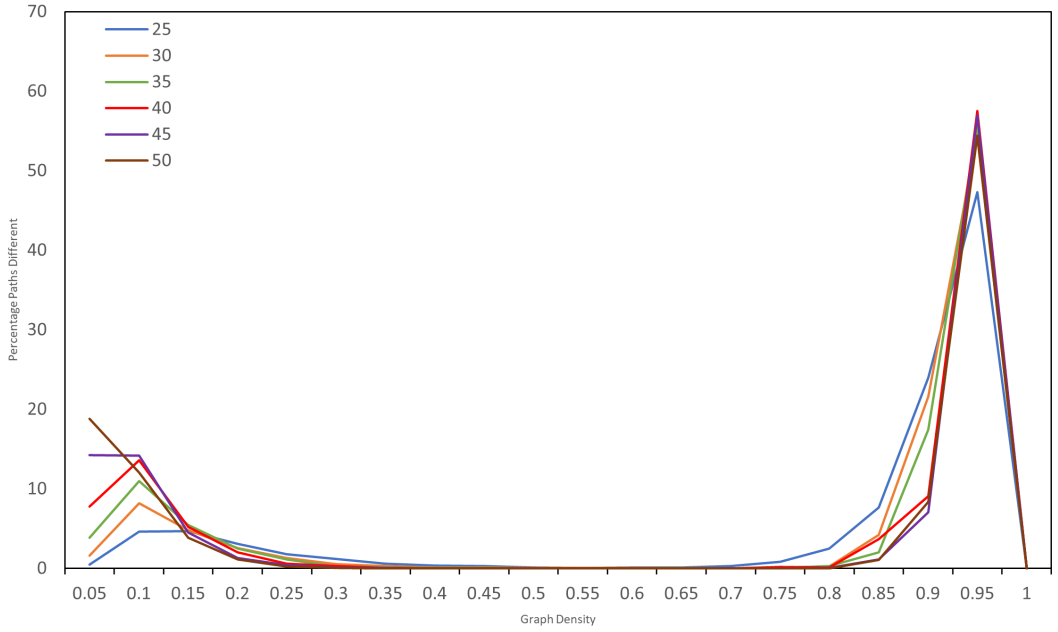
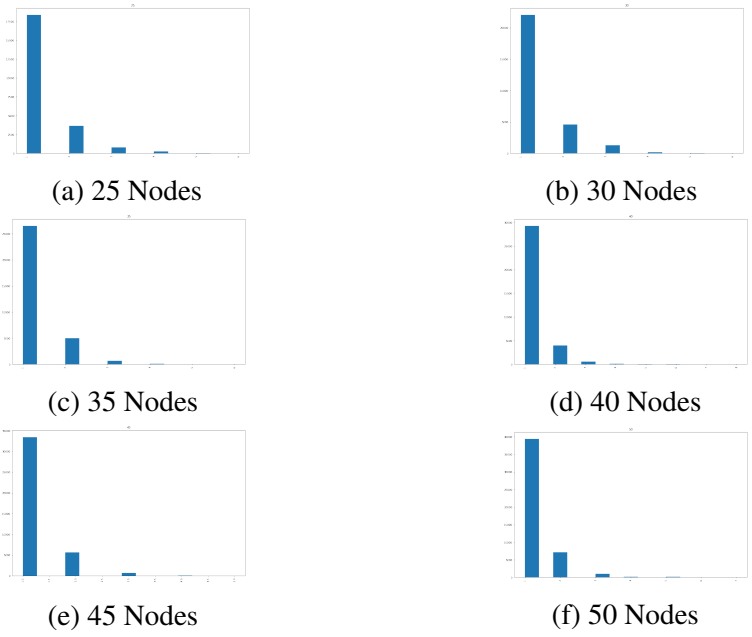


Figure 5.12: Percentage of all least covert paths with non-zero t_s^c and t_c^S that occur within one of 50 simulations at each vertex-density pair.

Figure 5.12 demonstrates that the highest volume of least covert paths which yielding non-zero covert-length and length-covert trade-offs occur when the density of the network is 0.95 in networks on between 25 and 50 vertices. Figure 5.12 looks very similar to the $I_s^c(G)$ results in 5.6; with an initial peak of these paths occurring in networks on between 0.05 and 0.15 density; although in networks on between 25 and 35 vertices, the lower quartile for the frequency of these paths still occurs at above 0.8 density due to an overwhelming proportion of paths occurring at 0.9 and 0.95 density. However, in networks on between 40 and 50 vertices, due to a drop in the percentage of these paths occurring in networks on 0.9 density, we see the lower quartile in frequency

occurs at a lower density of between 0.05 and 0.15. The upper quartile in frequency sits between 0.9 and 0.95 density across networks on 25-50 vertices. Therefore, in paths in networks on between 40 and 50 vertices, there is an increase in the range of densities corresponding to the interquartile range of frequency of occurrence of these paths in networks on between 25 and 40 vertices.

In Figure 5.13 and 5.14 we ran the same process as in Figure 5.9 and 5.10 but this time with least covert paths to calculate the frequency of various t_s^c and t_c^s values, with corresponding Table 5.5 and 5.6.



(a) 25 Nodes (b) 30 Nodes

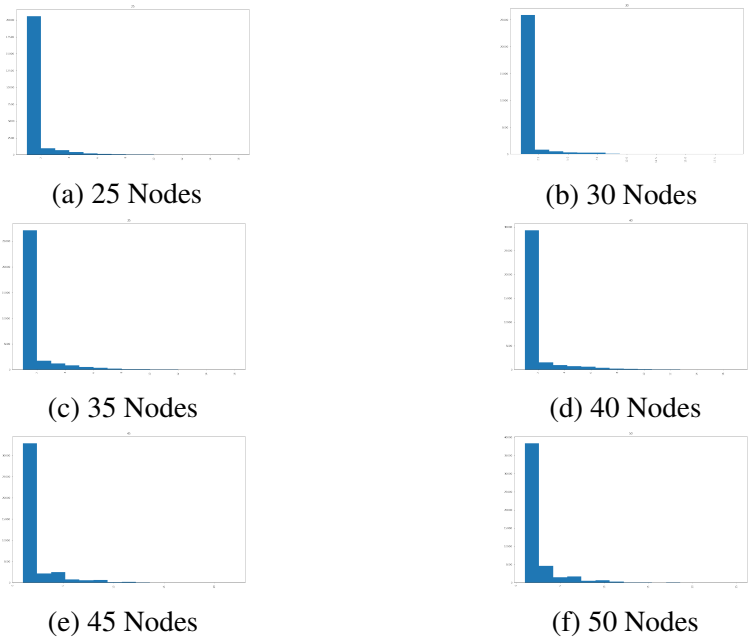
(c) 35 Nodes (d) 40 Nodes

(e) 45 Nodes (f) 50 Nodes

Figure 5.13: Frequency histograms plotting occurrence of t_s^c values, separated by number of vertices in the network.

Number vertices	1	2	3	4	5	6	7	8
25	79.47	15.76	3.48	1.13	0.15	0.00	0.00	0.00
30	78.31	16.29	4.58	0.65	0.16	0.01	0.00	0.00
35	81.82	15.51	2.19	0.39	0.09	0.01	0.00	0.00
40	85.53	11.79	1.67	0.36	0.29	0.27	0.10	0.00
45	83.83	14.21	1.84	0.13	0.00	0.00	0.00	0.00
50	82.10	14.96	2.09	0.41	0.35	0.09	0.00	0.00

Table 5.5: Percentage of all least covert paths with non-zero t_s^c for each network size that have the corresponding t_s^c value, corresponding to Figure 5.13.



(a) 25 Nodes (b) 30 Nodes

(c) 35 Nodes (d) 40 Nodes

(e) 45 Nodes (f) 50 Nodes

Figure 5.14: Frequency histograms plotting occurrence of t_c^s values, separated by number of vertices in the network.

Number vertices	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
25	89.47	4.25	2.94	1.75	0.80	0.04	0.38	0.17	0.13	0.04	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30	87.33	4.51	2.97	1.77	1.08	0.93	0.50	0.40	0.21	0.12	0.01	0.03	0.05	0.03	0.00	0.02	0.00	0.00	0.01	0.00	0.00	0.00
35	83.73	5.37	3.71	2.58	1.69	1.16	0.64	0.41	0.31	0.10	0.14	0.08	0.02	0.03	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
40	79.27	6.44	4.43	2.82	2.20	1.76	1.11	0.68	0.46	0.33	0.14	0.13	0.07	0.08	0.06	0.03	0.01	0.00	0.00	0.00	0.00	0.00
45	74.59	7.75	5.46	3.50	2.70	1.89	1.39	1.04	0.54	0.31	0.31	0.17	0.13	0.08	0.04	0.03	0.03	0.01	0.01	0.02	0.00	0.00
50	73.04	6.87	5.41	4.16	3.16	1.77	1.87	1.11	0.83	0.60	0.33	0.28	0.19	0.15	0.09	0.03	0.07	0.01	0.01	0.00	0.00	0.01

Table 5.6: Percentage of all least overt paths with non-zero t_C^s for each network size that have the corresponding t_C^s value, corresponding to Figure 5.14.

Similarly to results in Figure 5.9 , the most frequent t_s^c value is one (between 79 and

82 percent of paths in networks on between 25 and 30 vertices). However, there is a slightly higher volume of t_s^c values which are not one (eg. 15 percent of paths have a t_s^c value of two in networks on 25 vertices, as compared to 4 percent of paths in the same sized networks have a t_s^o value of two). The range of t_s^c values is slightly larger than the range of t_s^o values. In contrast to Figure 5.10; there is a larger proportion of t_c^s values of one (between 73 and 89 percent of paths), with a minor decrease in the percentage of paths as the number of vertices in the network increases. The range of t_c^s values results is the greatest out of all cases we've considered.

From Figure 5.6 we observe that the highest $I_s^c(G)$ values occur at 0.95 density, where from Figure 5.11 these occur where the length of the shortest path is one. There is also a cluster of $I_s^c(G)$ that occur at 0.9 density, when the length of the shortest path is also one.

Conversely, the initial $I_s^c(G)$ values we see between 0.05 and 0.15 density in Figure 5.6 occur when the shortest path is longer (between 3 and 5 hops in length). From Figure 5.12 we know that the highest volume of contributing paths occur at higher densities, where from Figure 5.11 the shortest path is shorter.

Highest $I_c^s(G)$ values occur when the covert weight of a shortest path is between 30 and 40. However, when the highest volume of contributing paths occur at 0.95 density, paths have between 0 and 10 total covert weight. This decrease in covert weight corresponds to the shortest paths being shorter.

5.3.3 Hub and Spoke format networks

In Figures 5.16 and 5.15 we consider alternative 'hub and spoke'type networks created through scale-free network constructions. Note that in these constructions and unlike ER networks, density cannot be directly controlled, only influenced by other parameters. This means that some networks represented in Figures 5.16 and 5.15 may have a unique density, and therefore the average statistics for a given density may be calculated from a

single network. As a consequence peaks result in Figures 5.16 and 5.15 which may refer to potentially outlying results. Furthermore, the nature of these networks, in terms of their governing long tail degree distributions, does not allow for high density networks to be constructed.

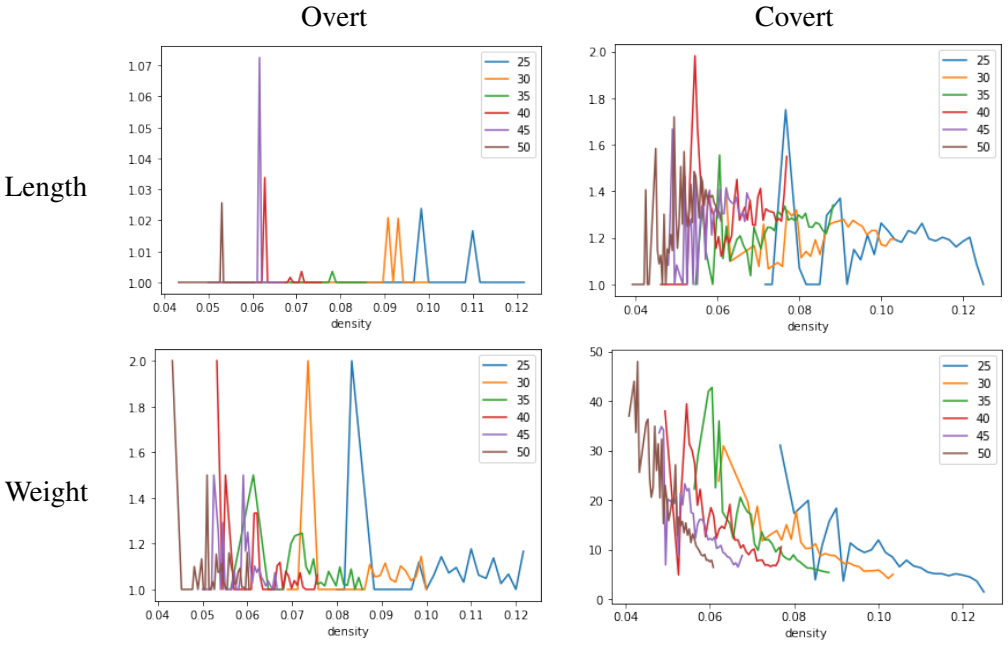


Figure 5.15: The average values of $I_s^o(G)$, $I_s^c(G)$, $I_c^s(G)$ and $I_o^s(G)$ (clockwise order), sampling over 100 instances of the random k network construction.

We observe that random k-out networks (Figure 5.15) are characterised by the most frequently occurring connected triads being `021U` and `021C`. Similarly the most frequently occurring connected triads in the scale free networks (Figure 5.16) are `021D`, `021U` and `021C`. The triad census of these networks do not follow the same trajectory as ER networks with increasing density, therefore, this means operations (1)-(4) are performed under a different set of restrictions to ER networks.

In `021D` and `021U` all edges are covert and in `021C` one edge is covert, and another overt. This means that on average, we predict the covert edge centrality is greater than the overt edge weight in these networks. We believe that `021C` triads are positioned so that the overt edges do not overlap, resulting in a low overt weight per edge. If true,

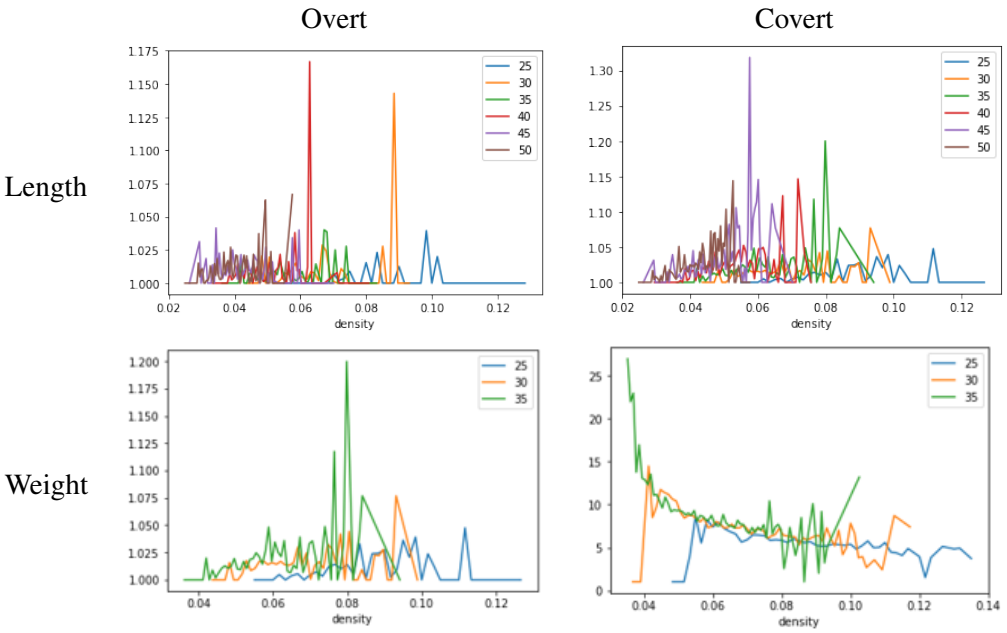


Figure 5.16: The average values of $I_s^o(G)$, $I_s^c(G)$, $I_c^s(G)$ and $I_o^s(G)$ (clockwise order), sampling over 100 instances of a scale-free network construction.

such networks are heavily weighted towards covert edges. These triads also pivot with high frequency around hub vertices.

The results (Figures 5.16 and 5.15) show the presence of least weight overt and covert paths that are longer than shortest paths ($I_s^o(G)$ and $I_s^c(G)$), with these appearing more prevalent for the covert weighting, consistent with covert edges being more prevalent in the underlying dominant triads.

In terms of edge weights, we find that high levels of jitter are present due to the inability to sample at a particular density level. However the results establish the presence of paths weighing less than shortest paths for both overt and covert paths (i.e., $I_o^s(G)$ and $I_c^s(G)$). This appears marginally more prevalent at lower density (although this is not established statistically).

5.4 Summary

In this chapter we have introduced an alternative analysis approach for paths in graphs, addressing the extent to which they provide opportunity for knowledge or content travelling along a path to be contained or widely disseminated, depending on the scenario. Weighting edges using their overt or covert centrality allows us to assess this using local metrics.

Using these metrics, we have examined the existence of least weight overt and covert paths and their relationship to shortest paths using two classes of synthesised network (ER networks and scale-free networks). Synthesised networks have been used because this problem is sensitive to edge density. Comparing these two classes of paths is applicable when choosing whether a path of communication need have less steps (through choosing the shortest path) or control the dissemination of information (minimising overt path weight of a path would reduce the spread of information, whilst minimising covert path weight would increase).

We have introduced four measures to discuss the difference between the two classes of paths: $I_s^o(G), I_o^s(G), I_s^c(G)$ and $I_c^s(G)$. $I_s^o(G)$ and $I_s^c(G)$ respectively denote the average non-zero overt-length and covert-length trade-off for G (i.e the number of additional edges that result when prioritising minimising overtness/covertness over path length), whilst $I_o^s(G)$ and $I_c^s(G)$ respectively denote the average non-zero length-overt and length-covert trade-off for G (i.e the decrease in path overt/covert weight when prioritising minimising overtness/covertness over path length). These measures are important as $I_s^o(G)$ and $I_s^c(G)$ can be viewed as a cost to a system in terms of number of hops when choosing a path of minimum overt/covert weight, whilst $I_o^s(G)$ and $I_c^s(G)$ a benefit in terms of dissemination of information. We conclude that major differences between paths occur in networks which are relatively small in terms of the number of vertices they are built on.

The values of $I_s^o(G), I_o^s(G), I_s^c(G)$ and $I_c^s(G)$ are affected by the induced substructures

which overlap with paths in the network. The type of network and its density affects which substructures occur and which type of edges are present. At higher densities, the volume of overt edges in the induced substructures present out weights the volume of covert edges. In this chapter we have concluded this by looking at the triad census of networks at varying densities.

Further, we have shown that the highest volume of least overt paths (which are not themselves shortest paths and therefore produce an overt-length trade-off) occurring in networks with between 25 and 50 vertices occur when the density of the network is between 0.05 and 0.25, with a peak in frequency of these paths generally occurring between 0.1 and 0.15 density. The overt-length and length-overt trade-off is most frequently one, although this is less pronounced in length-overt trade-off than overt-length. The highest $I_s^o(G)$ value occurs when a shortest path is between four and five hops in length, although, when paths most frequently occur between 0.1 and 0.15 density, the length of a shortest path is between three and four hops. Highest $I_o^s(G)$ occur when a shortest path has overt weight between 30 and 45, however, when most least overt paths occur between 0.1 and 0.15 density the shortest paths have total overt weight between 12 and 20. To summarise, *the biggest difference between paths of minimum overt weight and paths of minimum edges occur when a network has low density (0.1-0.15 density), a shortest path is short but not a single edge (3-4 edges long) and moderate overt weight (*$12 \leqslant w_o(p) \leqslant 20$*).*

We have also shown the highest volume of least covert paths (which are not themselves shortest paths and therefor produce a covert-length and length-covert trade-off) in networks on between 25 and 50 vertices occur when the density of the network is between 0.85 and 0.95; with a significant peak in frequency at 0.95 density. Again, the most frequent covert-length and length-covert trade-off is one. The highest $I_s^c(G)$ values occur when a shortest path has one hop (i.e it is an edge), and highest $I_c^s(G)$ occur when a shortest path covert weight is between 30 and 40. The most frequent paths occur at 0.95 density, when total covert weight of a shortest path is between zero

and 10 and its length is one. To summarise, *the biggest difference between paths of minimum covert weight and paths of minimum edges occur when a network has high density density (0.8-0.95 density), a shortest path is a single edge (3-4 edges long) and moderate overt weight (*$30 \leqslant w_c(p) \leqslant 40$*).*

Overall, this shows that overt and covert characteristics of paths is highly sensitive to the edge density in networks. Therefore, we conclude that to minimise spread of a message, it may be useful to control for overtness in low-density networks. Maximum spread, on the other hand, occurs most in high density networks when choosing longer paths over single edges.

This chapter revisits the research question:

RQ3: The role of edges of a triad in paths: combining RQ1 and RQ2, how can the edges which enable connectivity within the triad affect paths, and thereby connectivity within a graph?

We do this through applying overt and covert centrality to paths and comparing trade-offs, resulting in the fourth contribution:

C4: *A new method to understand the potential spread of a message through a local centrality metric for path problems in networks.*

Chapter 6

Assessing Edge Criticality through Overt and Covert Centrality

Continuing with the theme of understanding how graph connectivity can be characterised through the fundamental properties of overt and covert edges, in this chapter, we turn attention to the issue of *edge criticality*. Generally speaking, edge criticality represents how important an edge is to overall graph connectivity, and may be defined to reflect particular aspects of connectivity that are important to a scenario. Here we consider the concepts of overt and covert centrality from Chapter Four in combination, which we call overt and covert criticality (OCC). We hypothesise that this consideration can capture how critical an edge is to maintaining the structure of a graph, due to the importance of an edges role within induced triads. We test this new criticality metric on the 34 datasets described in Chapter Four, by sequentially removing edges from a graph according to their rank and observing the disruption in graph structure. We compare this with betweenness centrality [29], which is a well understood edge criticality metric that has been shown to identify bridge-like connectors in a graph. This comparison is interesting because overt and covert criticality is a localised criticality concept, while betweenness centrality is globally derived, based on the role of edges supporting important paths between vertices. We examine these differences in detail to understand the similarities and differences of the localised approach.

6.1 Overview

In Chapter Two, we defined edge criticality [79] as a measure of edge importance in terms of maintaining the graph structure. Those edges deemed critical are the ones which, when removed, cause the graph to separate into a greater number of connected components (we refer to this as the graph 'disintegrating'or 'collapsing'). We formally define a *critical edge* in Definition 47:

Definition 47. *Let $G = (V(G), E(G)$ be a graph, and let $(u, v) \in E(G)$. Let $G' = G - (u, v)$ be the resulting graph when (u, v) is removed from G. Then the edge (u, v) is critical if and only if the number of connected components in G' is greater than the number of connected components in G.*

Edges which are not critical to the overall structure and connectivity of the graph will instead cause the graph to shrink (by reducing the number of edges present) but will not collapse the structure. Consider Example 6.1.1.

Example 6.1.1. *Suppose $G = (V(G), E(G)$ is a graph as shown in Figure 6.1. Then the removal of any black edge will not break down the graph into more components, instead the removal will just reduce the number of edges in the graph. However, were we to remove the red edge (u, v) then the graph would break into two connected components, as shown in Figure 6.2. Therefore, in this example, (u, v) is a critical edge. We should note that eventually all edges eventually become critical once a sufficient number of edges are removed from G. For example, if G' were the resulting graph from removing $(u_2, u_3), (u_3, u_2), (u_2, u), (u, u_2)$ and (u, u_3) from G, then the edge (u_3, u) would be critical to G'.*

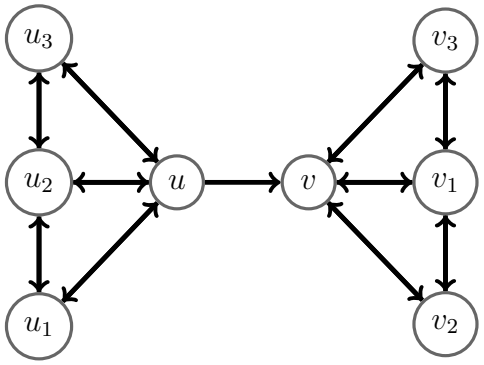


Figure 6.1: Graph G, where the removal of any black edge will not disconnect the graph, but the removal of the red edge (u, v) will disconnect the graph.

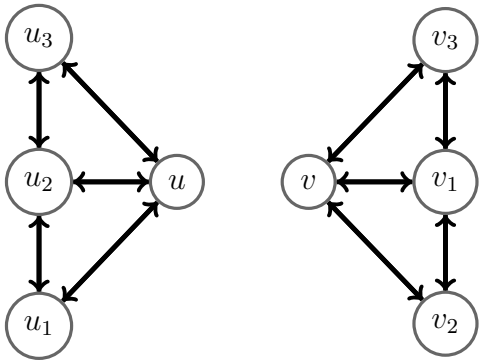


Figure 6.2: G', the resulting graph by removing (u, v) from G in Figure 6.1. G' has seperated into two connected components.

In more complex graphs, it can be difficult to determine which edges are critical. Further, some edges may not be a critical edge in the graph's present state, but through repeatedly removing edges and forming new graphs, they may become critical. How then do we determine which edges to remove first? Criticality metrics on edges are designed to identify which edges are *most critical to* a graph. The process of repeatedly removing edges allows us to assess which edges are *more critical to* a graph. That is, they cause the graph to collapse more quickly because they require the removal of fewer edges before they become a critical edge, than edges which are less critical to a graph. Edges are weighted under the criticality metric, ordered in terms of their weight and subsequently removed. Typically, measures such as the susceptibility index (see section) or size of giant component (see section) are repeatedly taken after the removal of each edge to

identify the *critical point*: at which a graph breaks down.

Criticality metrics can be compared by assessing the rate of which a graph decomposes by edge removal in accordance to the criticality metric, or the amount of damage the removal of edges causes. In Chapter Two we highlighted various criticality metrics, including Cheng et al.'s [77] bridgeness index, Girvan and Newman's [29] betweenness centrality for edges and Yu et. al.'s [79] betweenness centrality and clique model (BCCMOD). We also learned that strength of ties can be an important factor in determining edge criticality: in some graphs (such as social graphs) weak ties can actually be ties most critical according to the theory of weak ties [32]. This holds true for example 6.1.1, where (if strength of ties is determined by the degree to which they are reciprocated) the removal of the weakest tie (u, v) causes the disintegration of the graph.

Criticality metrics are used to assess which edges of a graph, when removed, cause the graph to most rapidly partition into a more connected components than the original graph, or cause the graph to partition into the greatest number of connected components. In mathematics, a similar problem exists, namely the minimum k-cut of a graph. The minimum k-cut asks for the edges whose removal partitions a graph into k components, where the total weight of removing edges is minimised.

Definition 48. *Let $G = (V(G), E(G))$ be a graph where each edge $(u, v) \in E(G)$ has weight $w(u, v)$. For some $k \in \{2, 3, \ldots |V(G)|\}$, partition $V(G)$ into k disjoint sets $F = C_1, C_2, \ldots C_k$ whilst minimising:*

$$\sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \sum_{v_1 \in C_i, v_2 \in C_j} w(\{v_1, v_2\}) \tag{6.1}$$

The minimum k-cut is equivalent to our problem when certain restrictions are introduced. Firstly, we weight edges based on their criticality metrics in order to assess which edges to remove, but we are trying to minimise the number of edges we remove. This is equivalent to the minimum k-cut problem when all edges have equal weight $w_(u, v)$. Further, k is not a fixed integer: since we simply want the removal of edges to cause the

graph to break down into a greater number of connected components than the original graph. Indeed, if k is greater removing a certain subset of edges as compared with another, then we would conclude the this subset of edges are more critical to the network. Thus, k is bounded below by the number of connected components in the original graph.

Existing approaches to solve the minimum k-cut problem are complex. For example, they involve approximating the minimum k-cut through constructing Gometry-Hu trees [31]; or creating minimum cuts (partitioning the graph into at least two connected components) in each connected component and removing the heaviest one. Creating these constructions are themselves combinatorial problems and require multiple computations. In contrast, when considering edge criticality through computing OCC is simple: OCC relies simply on the number of triads an edge acts as overt in, and the number of triads an edge acts as covert in. In Chapter Four Section 4.4.1 we show this simply relies on the edges adjacent to the one we are interested in, and therefore is scalable and computationally simple.

In this chapter we construct our own localised criticality metric from the overt and covert centrality metrics in Chapter Four. Previously in this thesis we've considered these measures in isolation, but here it is insightful to consider these metrics simultaneously. Note that this can offer a more local, less computationally complex, approach to criticality than existing models, which rely either on computation of cliques (such as the bridgeness index), computation of all shortest paths in a graph (such as in betweenness centrality for edges) or both (such as in BCCMOD). We define our criticality metric, OCC in Definition 49:

Definition 49. *The joint overt and covert centrality of an edge $(u, v) \in E(G)$ is defined by:*

$$OCC(u, v) = (O_{uv}, C_{uv}) \tag{6.2}$$

where O_{uv} denotes the overt centrality for (u, v) and C_{uv} denotes its covert centrality.

6.2 Considering Overt and Covert Centrality Simultaneously

As aforementioned, we have previously considered overt and covert centrality separately. In this chapter we move to considering the two measures in tandem, using this as a cricitality metric to apply to edges (OCC). We investigate the effect of both overt and covert centrality simultaneously through exploring the KDE plots of the joint overt and covert centrality (OCC) across the 34 data sets. This builds on Chapter Four, which plots the individual frequency distributions of overt and covert centralities across the 34 data sets.

These KDE Plots visualise the probability of overt or covert centrality. In the following plots 6.3, - 6.5 across the top of the x axis we plot the KDE plot for overt centrality, along the top of the y axis we show the KDE plot for covert centrality, and in the centre of the plot we show the probability of an edge having a joint overt-covert centrality. This main plot is a contour plot: when lines are closer together this represents a peak in the third dimension. Multiple data sets are plotted in one plot, differentiated between by the difference in colour of the lines (which correspond to the line colour in the legend).

From Figures 6.3, 6.4 and 6.5 we can see that the plots loosely fit into three categories.

There are graphs (such as Airport and WWW) whose KDE plot forms a triangle in the lower quartile of the axes. This means there are some edges with a low overt centrality but high covert centrality (those which from the point in the triangle at the top of the y axis but have a low value in the x axis); some edges with a high overt centrality but low covert centrality (those which form the second point in the triangle high in the x axis but low in the y axis); and a high concentration of edges with both low overt and covert centrality. These are characterised by having a greater range in the individual KDE plots of overt and covert centrality.

In contrast to the triangle in the lower quartile, graphs such as Internet and Organise,

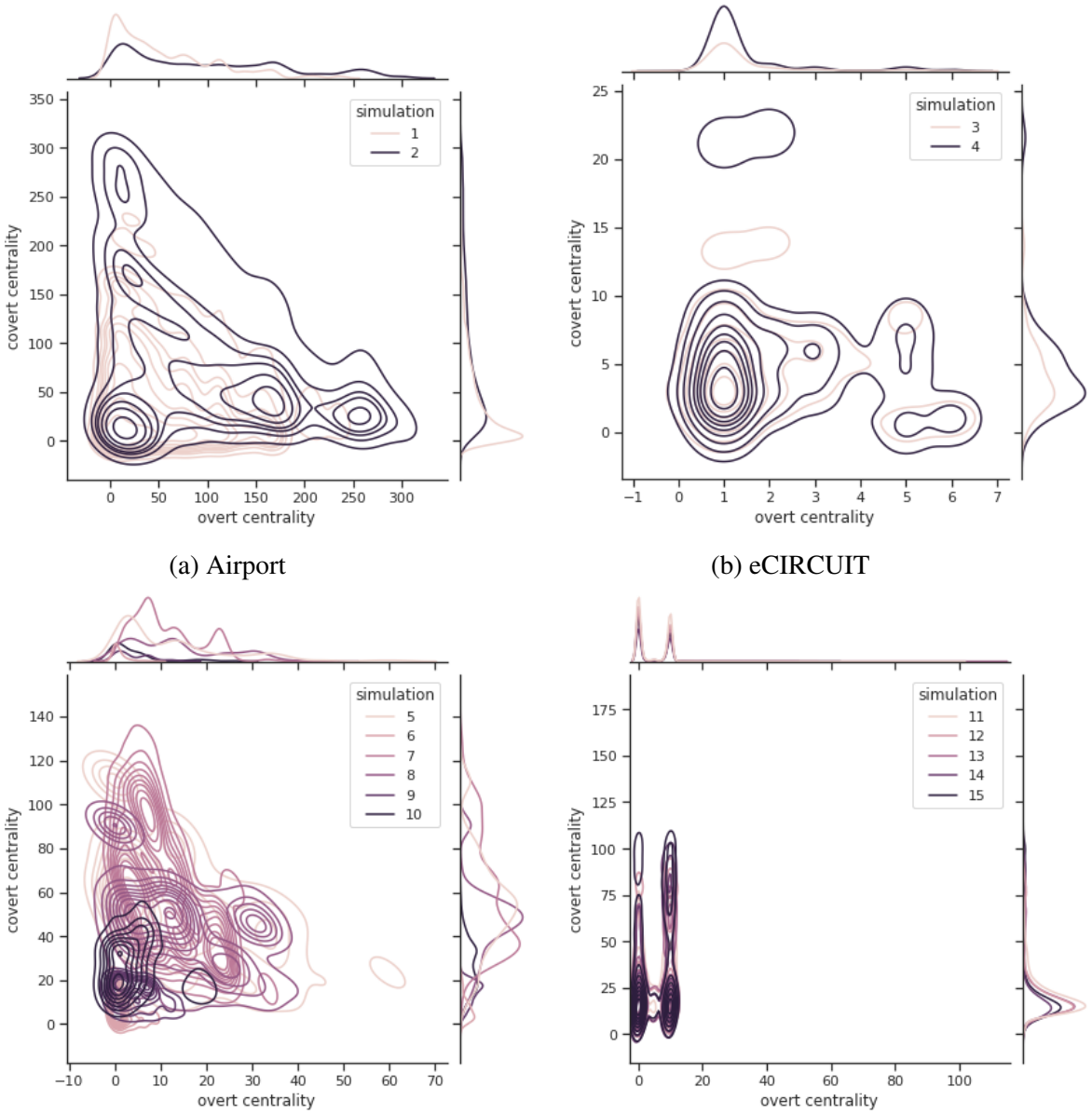
form a much thinner line, with little variety in the x-axis, but do have variety in the y axis. These are often heavily centered around a paired overt-covert centrality close to $(0, 0)$. In the separate overt and covert centrality KDE plots on the axis, these are characterised by a sharp peak in overt centrality at zero. Covert centrality can also have a sharp peak close to 0, though this is less sharp. Importantly, it's the range of values in covert centrality that are stretching out the plots, making them form this thin line, rather than a point at $(0, 0)$.

Finally there are a variety of plots that sit somewhere in between, such as Food web or Neural networks. There is a lot of variety within the category of graph itself; for example in the Neural networks, Rattus Norvegicus forms three distinct peaks, with the largest peak being centered on (400, 75), whereas Mouse Retina looks a little like the thin line plots but with a greater variation in overt centrality.

Observing overt and covert centrality simultaneously like this allows us to observe the two measures in a new way. Previously we were unable to establish the degree of overlap between the two metrics. For example, in the organise network Eva: if we were to only look at the overt centrality KDE plot we would miss the second peak with a much higher covert centrality that occurs in the paired KDE plot; meaning we would be unaware of the difference between the edges of low overt centrality and low covert centrality and the edges with low overt centrality yet high covert centrality. This could be solved by looking at the covert centrality KDE plot as well, since we would establish some variety in covert centrality yet little to no variety in overt centrality. However, in plots where there is more variety in both overt and covert centrality, it becomes much harder to read results from looking at the KDE plots for the individual metrics. For example, in Airport graphs we now see that those edges with high overt centrality tend to have a low covert centrality and visa versa due the triangular shape in the lower quadrant the KDE plots form. As there are no points in the upper quadrant, we can establish at lack of edges with both a high overt and high covert centrality.

Overt centrality is built on the basis that edges which are more central to a graph are those

which enable the flow of information to reach nodes one hop away from the intended recipient, with the intention that this effect on multiple edges will exponentially increase the number of nodes a message reaches. Covert centrality is built on the opposite assumption. We therefore hypothesise that those edges that are most overt will be more critical to the structure of a graph. However, as we've seen in the KDE Plots (Figures 6.3, 6.4 and 6.5) it is also important to consider in tandem covert centrality as the edges with a higher covert centrality, as well as a high overt centrality, may play a different role from those edges with a high overt centrality but low covert centrality. Therefore, covert centrality forms the secondary variable in OCC to determine an edge's criticality.



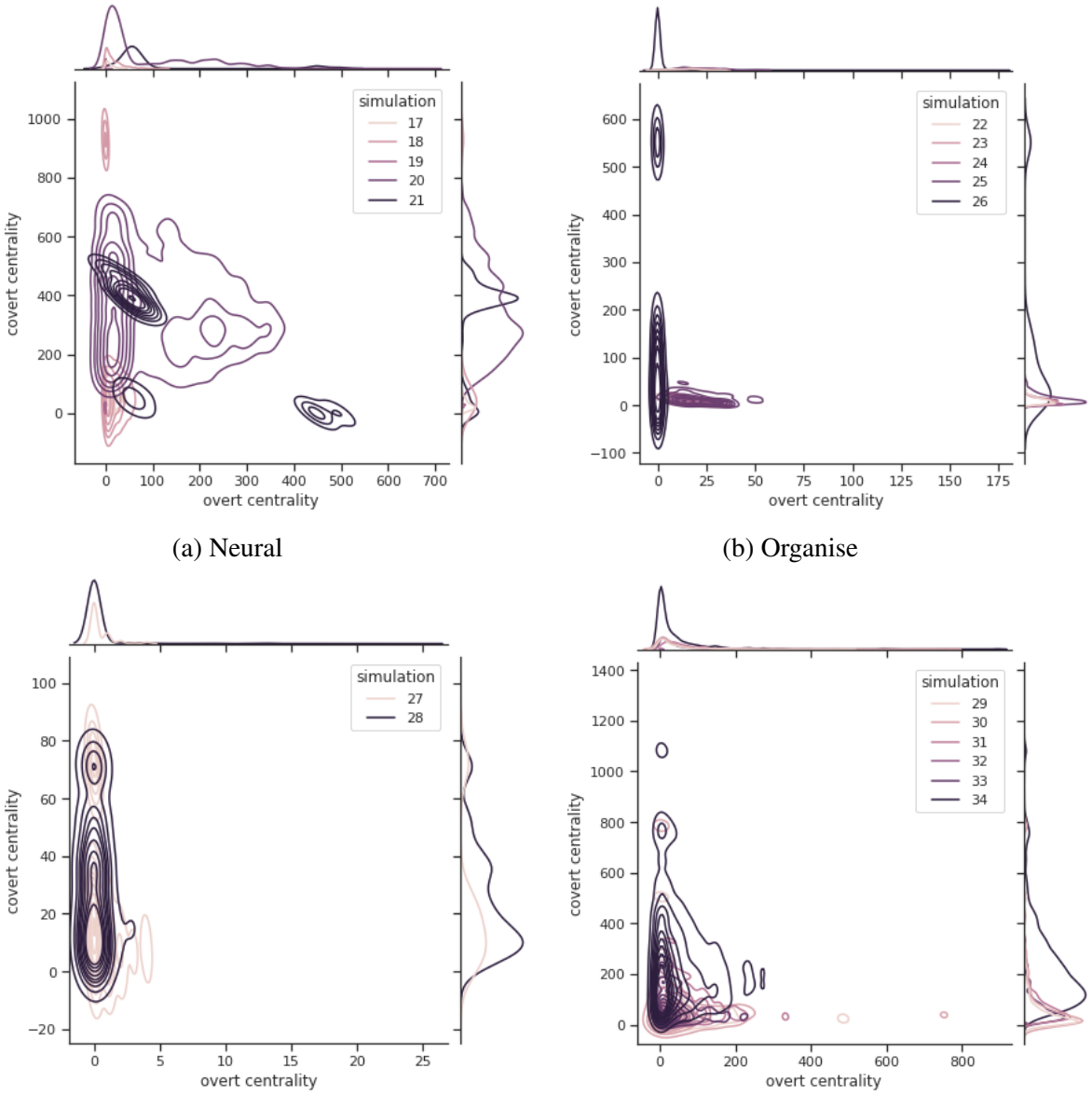
(a) Airport

(b) eCIRCUIT

(c) Foodweb

(d) Internet

Figure 6.3: KDE Contour plots to show the probability of paired overt-covert weight edges.



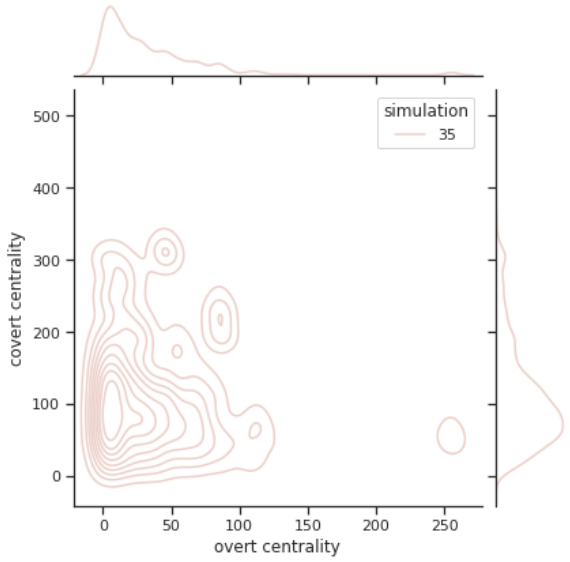
(a) Neural

(b) Organise

(c) Regulatory

(d) Social

Figure 6.4: KDE Contour plots to show the probability of paired overt-covert weight edges.



(a) WWW

Figure 6.5: KDE Contour plots to show the probability of paired overt-covert weight edges.

6.3 Methodology

We weight edges in terms of their OCC (see Definition 49), and rank them in descending order, with overt taking precedence over covert. Edges with a tied value are randomised, and bi-directional edges are treated as separate edges. To define descending order, we first consider what it means for the OCC of one edge to be greater than OCC of another:

Definition 50. *Let* $G = (V(G), E(G)$ *be a graph and* (u_i, v_i) *and* $(u_{i+1}, v_{i+1}) \in E(G)$. *Then* $OCC(u_i, v_i) > OCC(u_{i+1}, v_{i+1})$ *if and only if* $O_{u_i,v_i} > O_{u_{i+1},v_{i+1}}$ *or* $O_{u_i,v_i} = O_{u_{i+1},v_{i+1}}$ *and* $C_{u_i,v_i} > C_{u_{i+1},v_{i+1}}$. $OCC(u_i, v_i) = OCC(u_{i+1}, v_{i+1})$ *if and only if* $O_{u_i,v_i} = O_{u_{i+1},v_{i+1}}$ *and* $C_{u_i,v_i} = C_{u_{i+1},v_{i+1}}$.

We can now define the vector of edges sorted in descending order according to their OCC:

Definition 51. *Let $G = (V(G), E(G)$ be a graph on n vertices and $OCC(u, v)$ denote the overt and covert centrality of an edge $(u, v) \in E(G)$.*

Then E_{OCC} is the vector containing the edges in G ranked in descending order in terms of their OCC value, i.e:

$$E_{OCC} = ((u_1, v_1)(u_2, v_2), \ldots (u_n, v_n)) \tag{6.3}$$

where $OCC(u_i, v_i) \geqslant OCC(u_{i+1}, v_{i+1})$.

The function for sorting edges in descending order according to OCC, EDGESORTOCC, can be found in Algorithm 14. The algorithm for sorting edges in descending order according to their betweenness centrality follows similarly. This algorithm uses functions OVERTCENTRALITY (Algorithm 6) and COVERTCENTRALITY (Algorithm 7) from Chapter Four.

Algorithm 14 EDGESORTOCC((G)): Sorts edges in descending order according to their OCC.

Input: Some graph G.
Output: $Edgelist$, an ordered list (descending order) of edges according to OCC.

1: **for** $(u, v) \in E(G)$ **do**
2: **if** $u \neq v$ **then**
3: $OCC(u, v) = (\text{OVERTCENTRALITY}((u, v)), \text{COVERTCENTRALITY}((u, v)))$
4: $Edgelist = ((u_1, v_1), (u_2, v_2), \ldots (u_n, v_n))\ OCC((u_i, v_i)) \geqslant OCC((u_{i+1}, v_{i+1}))$
▷ Sort edges in descending order according to their OCC value. Note, we apply a random shuffle to the original egdelist in G before sorting so that if ties occur, they appear in $Edgelist$ in a random order, not the original order the programme reads them in.

From each graph G, we remove each edge in order of their occurrence in E_{OCC} from G and record the size of the largest weakly connected component (Definition 52) and susceptibility index (Definition 53).

Definition 52. *Let C denote the set of all weakly connected components in a graph G. Then the size of the largest (in terms of the number of nodes) weakly connected*

component in G is defined by $\sigma(G)$, where:

$$\sigma(G) = max\{|V(c)| : c \in C\} \tag{6.4}$$

Definition 53. *The normalised susceptibility index of a graph G is given by $S(G)$ where:*

$$S(G) = \sum_{s<\sigma(G)} \frac{n_s s^2}{|V(G)|} \tag{6.5}$$

where $\sigma(G)$ is the size of the size of largest weakly connected component in G, n_s is the number of components whose size equals s and $|V(G)|$ is the number of vertices in G [79].

The susceptibility index and size of largest weakly connected component are plotted as edges are removed. An obvious peak in $S(G)$, or a sharp fall in $\sigma(G)$ indicates the precise moment of graph disintegration [77].

The function CRITICALITYOCC for repeatedly calculating the susceptibility index $S(G)$ and size of largest connected component $\sigma(G)$ as edges are removed can be found in Algorithm 15.

Algorithm 15 CRITICALITYOCC(G): Produces the susceptibility index and size of largest connected component as edges are removed from G in descending order.

Input: Graph G

Output: Sequence of largest connected component sizes and susceptibility indexes as edges removed

lccs $= []$
susceptibilities $= []$
$Edgelist$ = EDGESORTOCC(G)
for $(u, v) \in Edgelist$ **do**
    $G = G(V(G), E(G) - (u, v))$
    $GCC = (g_1, g_2, \ldots g_n)$ ▷ Ordered connected components i.e. $|V(g_i)| \geq |V(g_{i+1})|$
    lccs.append($|V(g_1)|$)
    $s_{list} = []$
    for $0 \leq s \leq |V(g_1)|$ **do**
        $n_s = 0$
        for $g \in GCC$ **do**
            if $|V(g)| = s$ **then**
                $n_s = n_s + 1$
            $s_{list}.append\left(\frac{n_s s^2}{|V(G)|}\right)$
    susceptibilities.append(sum(s_{list}))
return lccs, susceptibilities

6.4 Results

In Figures 6.6 to 6.19 we plot the normalised susceptibility index $S(G)$ and size of largest connected component $\sigma(G)$ as edges are removed from each data set. We remove edges in descending order according to their OCC; then compare this with betweenness

centrality for edges. The parameter p denotes the proportion of edges removed from the graph G, ie.:

$$p = \frac{\text{number of removed edges from } G}{\text{total number of edges in } G} \tag{6.6}$$

We are interested in finding the point at which the graph disintegrates when edges are removed (that is, the point at which the graph separates into a greater number of connected components than the original graph). An obvious peak in $S(G)$, or a sharp fall in $\sigma(G)$ indicates the precise moment of graph disintegration. A criticality metric *outperforms* if it causes a graph to disintegrate more quickly (i.e for smaller p, there is a peak in $S(G)$ or a decline in $\sigma(G)$) or the damage caused to a graph is greater (i.e there is a larger peak in $S(G)$ or greater fall in $\sigma(G)$).

Observing Figures 6.6 - 6.19 there are several instances where OCC outperforms betweenness centrality in terms of an edge's importance to the structure of a network. For instance, across the Internet networks (Figure 6.10 and 6.11) the peak in S occurs at a smaller p when removing edges according to their OCC than betweenness centrality. This is consistent across all Internet networks, and in some cases there is no obvious peak for betweenness centrality at all; yet there is an obvious peak for OCC (p2p-gnutella04, p2p-gnutella05 and p2p-gnutella06 [47]). This corresponds to the results for largest connected component. Across networks p2p-gnutella04 to p2p-gnutella09 there is at first a faster decrease in σ for betweenness centrality, although it appears that this only represents a decrease in network size and not disintegration of the network, since there is no corresponding peak in S for the same p. However, we see that when $p = 0.85$, there is a rapid decrease in σ for OCC which corresponds to a peak in S for the same p. At this point, OCC overtakes betweenness centrality for a rapid reduction in σ.

A similar pattern emerges across the Social networks (Figure 6.16 and 6.17). Certainly in Bitcoin Alpha [47], Bitcoin OTC [47], UCIrvine [63] and Wikivote [47], an earlier and more obvious peak in S occurs in OCC than betweenness centrality, at the precise moment when the decrease in σ for OCC overtakes betweenness centrality. For the

Email EU Core [47] and Prison inmate [5] networks, the peak in S occurs around the same p value for betweenness centrality and OCC, though the peak is more obvious for OCC. This means that the damage to the network caused by removing edges based on OCC is greater than that produced by betweenness centrality.

Neural networks also tend to favour OCC over betweenness centrality in terms of network disintegration. Across the Neural networks (Figure 6.12 and 6.13), OCC outperforms betweenness centrality in the C. Elegans [40], Drosophila Medilla 1 [58] and Mouse Retina networks [58]; where no obvious peak in S exists for betweenness centrality. However, in the Rattus Norvegicus [58] network, a more obvious peak occurs in betweenness centrality than OCC for the same p, though the difference in peak size is small.

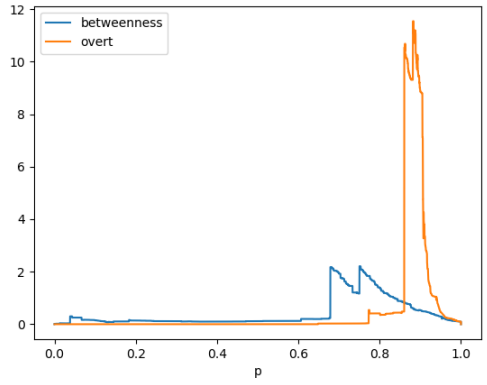
There are of course, those networks that sit somewhere in the middle, such as Airport (Figure 6.6), Electrical circuit (Figure 6.7) and Regulatory (Figure 6.18) networks. Across the Food web networks (Figure 6.8 and 6.9) OCC appears to outperform betweenness centrality in the Baywet [74], Grassland [14] and Ythan [1] networks, yet perform worse in Little Rock Lake [42] and St.Marks Seagrass [74] networks. Regulatory networks (Figure 6.18) tend to produce better results when using betweenness centrality. A sharper peak in S occurs in the Political Blogs network [2] (Figure 6.19) for OCC than betweenness centrality, but for the same p value. This means though the networks dissolve after a similar number of edges are removed, the damage caused to network structure is greater in OCC than betweenness centrality (in terms of number of connected components occurring by removing edges is greater). Organise networks (Figure 6.14 and 6.15) are also mixed, tending to collapse quicker under betweenness than OCC, with the exception of Freemans EIES n48 2 [63] (where a peak in S occurs around the same p, but is stronger for OCC) and Eva [41] (where a peak in S occurs for smaller p in OCC than betweenness, yet the peak is larger for betweenness and therefore causes more damage to the network).

It appears OCC is most effective (as compared with betweenness centrality) at ranking

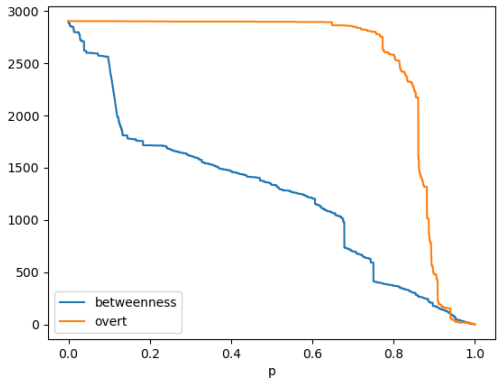
edges in descending order such that their removal in this order dissolves the network for smaller p in networks such as Internet networks, whose KDE plots of the joint overt and covert centrality (Figure 6.3) form thin lines in the bottom left quadrant; and in Figures 4.19 - 4.23 the overt and covert centrality profiles are long-tailed. In Chapter Four Section 4.4.2 we discussed the possibility that networks which are hub-and-spoke in structure contain few edges with high overt/covert centrality (those edges between hub nodes) and many edges with lower overt/covert centrality (those edges between hub nodes and spoke nodes). Thus, removing edges with high OCC may remove edges between hub nodes, effectively deleting the bridges in the network and collapsing the network into more connected components, resulting in an peak in S for smaller p. Therefore, OCC could be a good candidate for finding bridges in particular constructions of network.

Certainly the opposite argument seems to correspond: in networks with bell shaped overt and covert centrality profiles and triangular OCC profiles, like the Organise networks, betweenness centrality seems to outperform OCC in terms of dissolving the network for smaller p when edges are ranked in descending order and removed. This however doesn't hold true for Freemans EIES n48 2 [63]. Further, we often relate Eva [41] to Internet networks because they have similar statistics in Table 2.4 and similar overt/covert centrality profiles in Chapter Four Figures 4.19 - 4.23 and 4.33 . Yet whilst OCC dissolves the Eva network [41] for smaller p than betweenness centrality, betweenness centrality appears to cause the network to break down into a larger number of connected components than OCC (observe the larger peak in S for betweenness centrality than OCC). Further counter examples can be found across Regulatory networks, where the KDE plots of the joint overt and covert centrality (Figure 6.3) forms a thin line in the bottom left quadrant; and the overt and covert centrality distributions in Figures 4.40 and 4.41 are long-tailed; yet in Figures 6.18 we see that betweenness centrality is more effective than OCC in dissolving a network for smaller p. It seems there is more behind this than a network being hub-and-spoke. Directionality could be a driving factor here. Since OCC prioritises overt over covert,

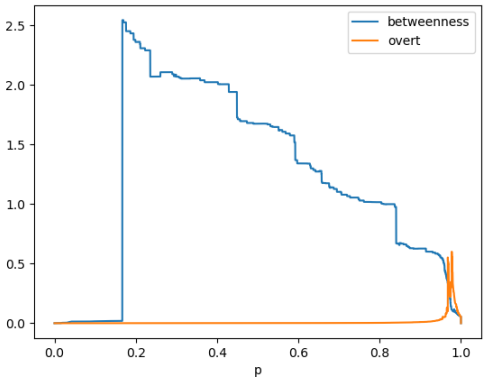
it will first look for edges with the highest overt centrality. It could be that the edges between the hub nodes in Internet networks are predominantly overt. In networks such as Eva, edges between the hub nodes could be contained in many triads but act as covert, effectively giving these edges a high covert centrality but low overt centrality. This would mean OCC would not rank these edges as highly as edges which are contained in fewer triads but have a higher overt centrality, such as edges connecting hub nodes with spoke nodes. Therefore, OCC may ignore bridge edges if they act as covert in the triads they are contained in. This could certainly explain differences between effectiveness of OCC and betweenness centrality in dissolving a network when the networks are largely similar such as food web networks. In Figures 6.8 and we see OCC is more effective than betweenness centrality in dissolving the network for Baywet [74], Grassland [14] and Mangwet [74] networks yet betweenness centrality is more effective in dissolving the network than OCC for Little Rock Lake [42] and St.Marks Seagrass [74].



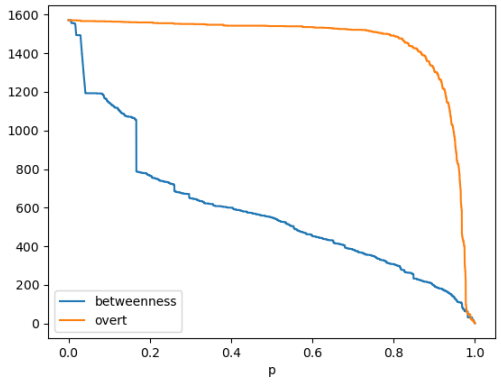
(a) Open Flights [67] Susceptibility Index Comparison



(b) Open Flights [67] Size of Largest Connected Component

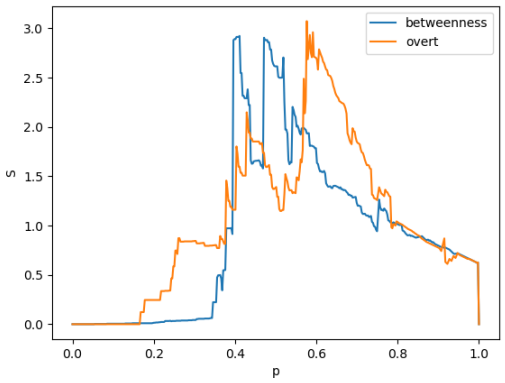


(c) US Airports [42] Susceptibility Index Comparison

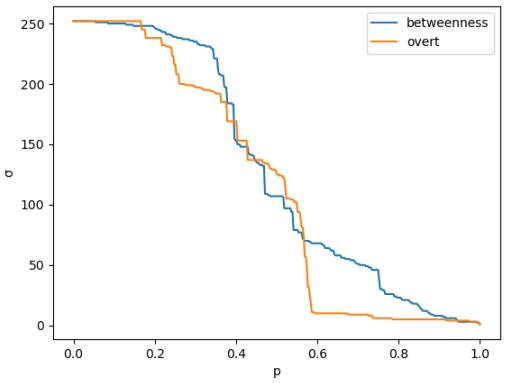


(d) US Airports [42] Size of Largest Connected Component

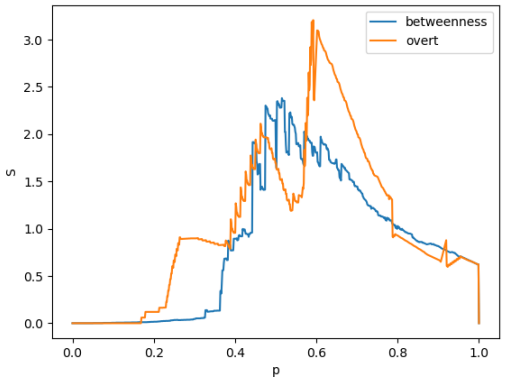
Figure 6.6: Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in all Airport data sets.



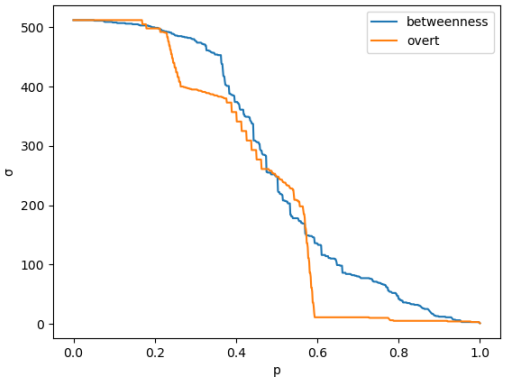
(a) s420 [5] Susceptibility Index Comparison



(b) s420 [5] Size of Largest Connected Component

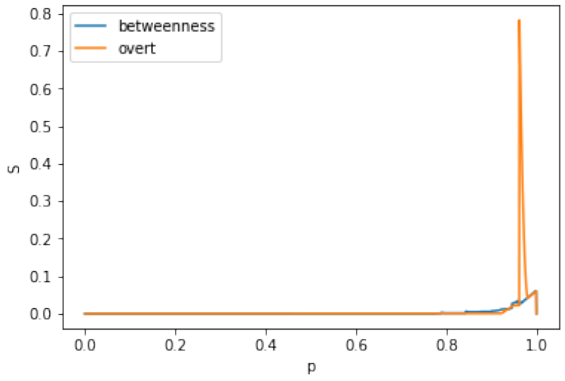


(c) s838 [5] Susceptibility Index Comparison

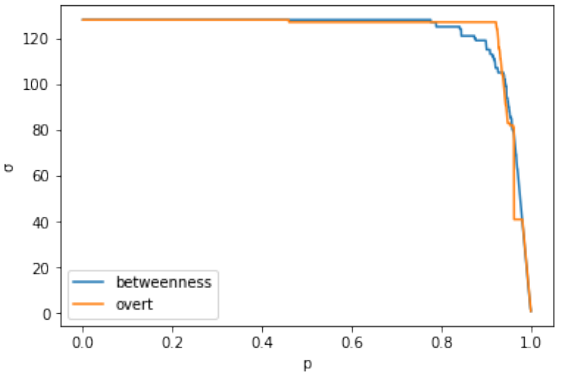


(d) s838 [5] Size of Largest Connected Component

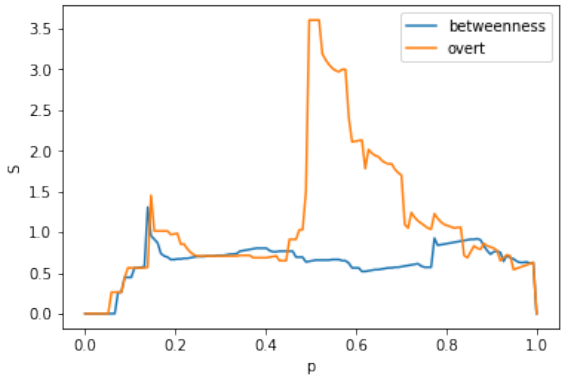
Figure 6.7: Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (left) against the proportion of edges removed, p, in all Electrical circuit data sets.



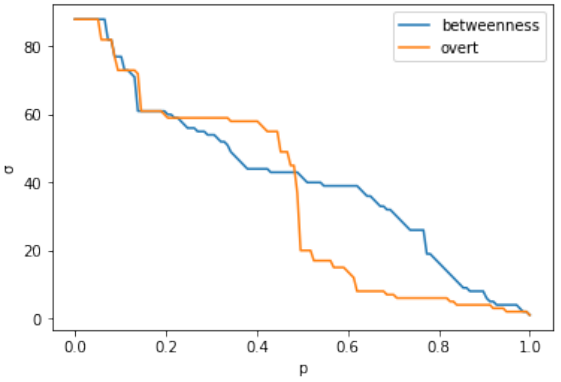
(a) Baywet [74] Susceptibility Index Comparison



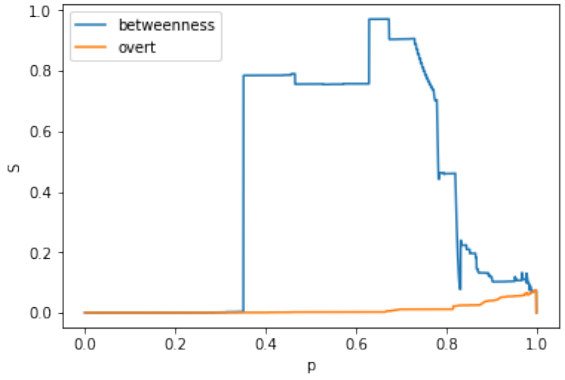
(b) Baywet [74] Size of Largest Connected Component



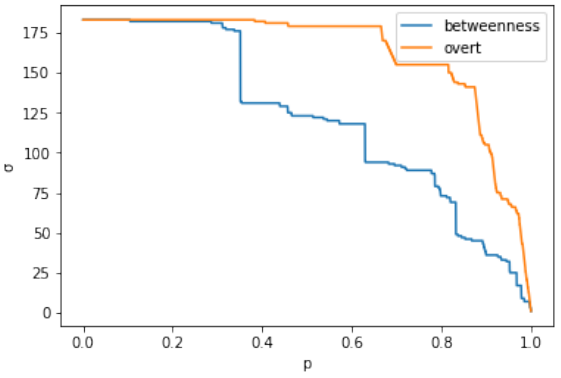
(c) Grassland [14] Susceptibility Index Comparison



(d) Grassland [14] Size of Largest Connected Component

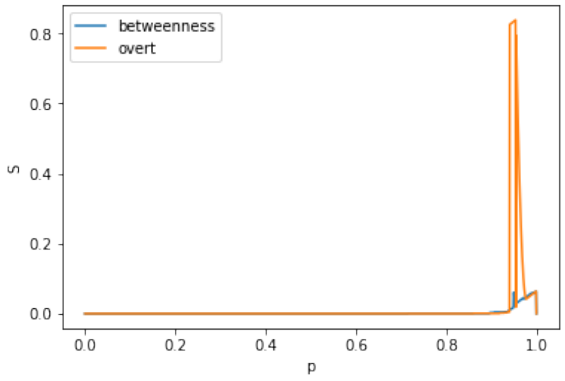


(e) Little Rock Lake [42] Susceptibility Index Comparison

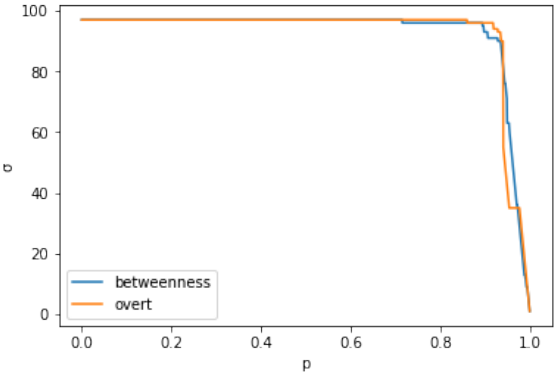


(f) Little Rock Lake [42] Size of Largest Connected Component

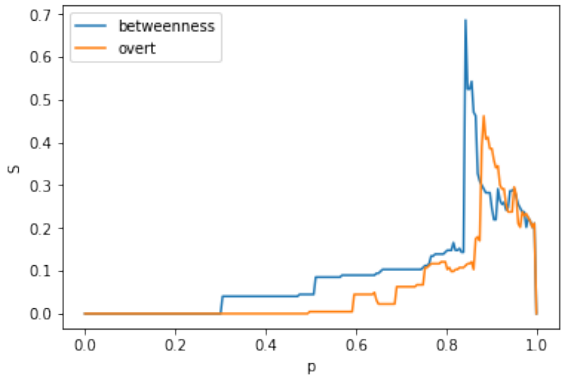
Figure 6.8: Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in all Food web data sets.



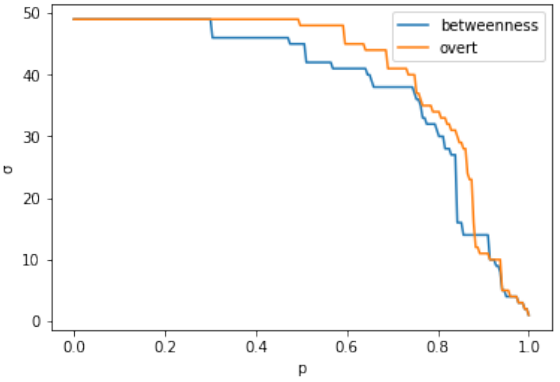
(a) Mangwet [74] Susceptibility Index Comparison



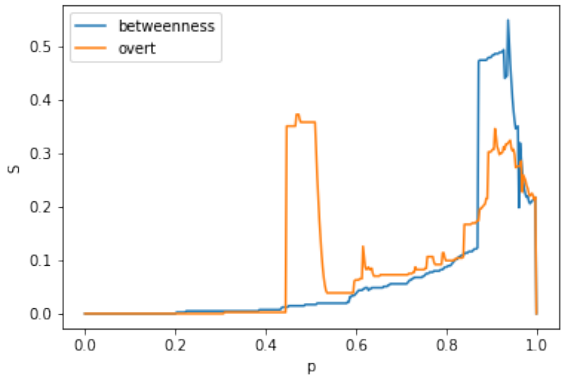
(b) Mangwet [74] Size of Largest Connected Component



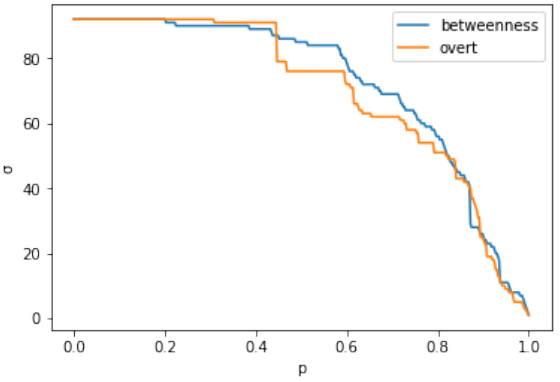
(c) St. Marks Seagrass [14] Susceptibility Index Comparison



(d) St. Marks Seagrass [14] Size of Largest Connected Component

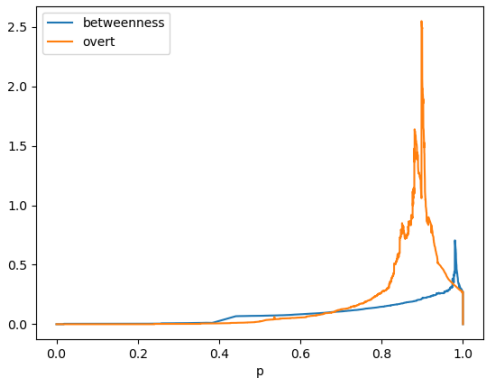


(e) Ythan [1] Susceptibility Index Comparison

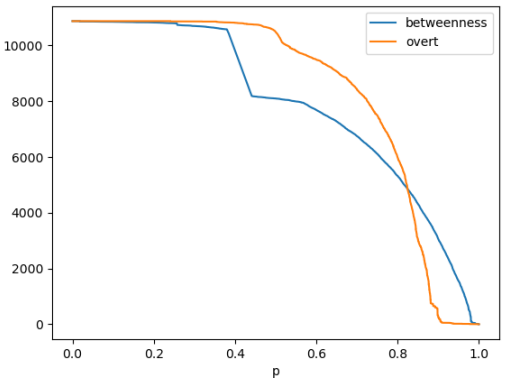


(f) Ythan [1] Size of Largest Connected Component

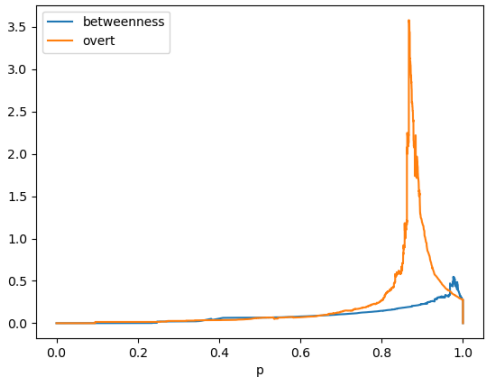
Figure 6.9: Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (left) against the proportion of edges removed, p, in all Food web data sets.



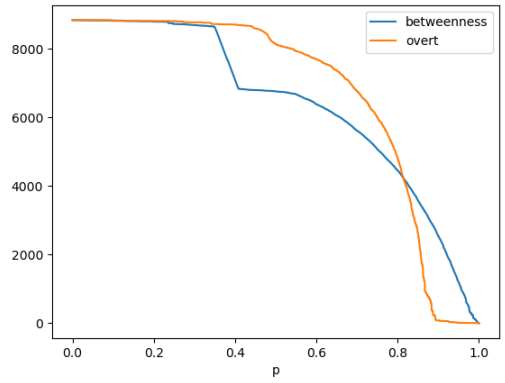
(a) p2p-gnutella04 [47]Susceptibility Index Comparison



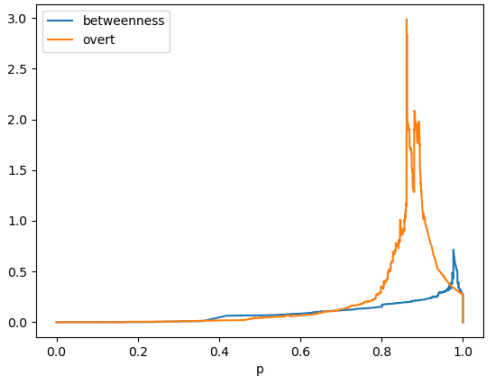
(b) p2p-gnutella04 [47] Size of Largest Connected Component



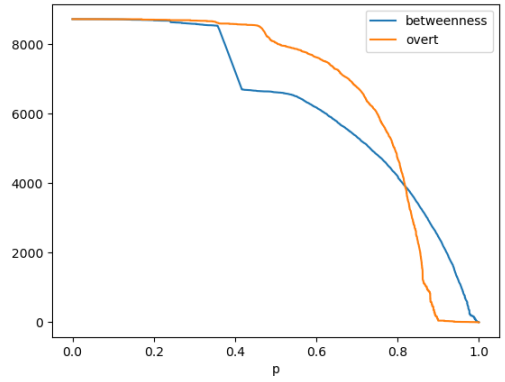
(c) p2p-gnutella05 [47] Susceptibility Index Comparison



(d) p2p-gnutella05 [47] Size of Largest Connected Component

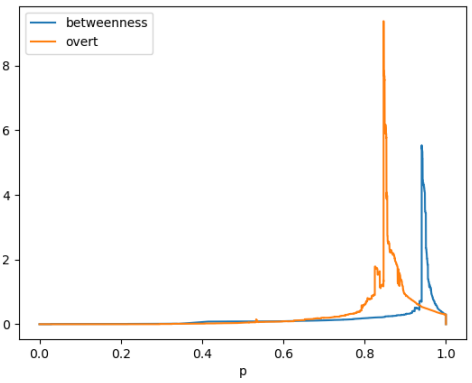


(e) p2p-gnutella06 [47] Susceptibility Index Comparison

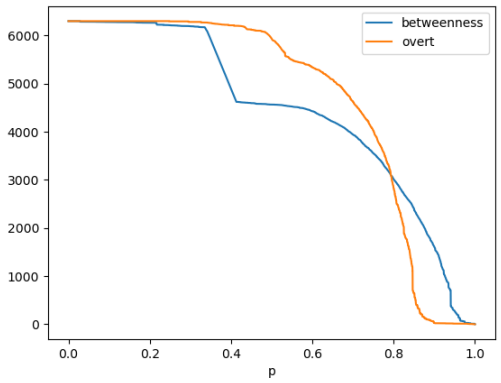


(f) p2p-gnutella06 [47] Size of Largest Connected Component

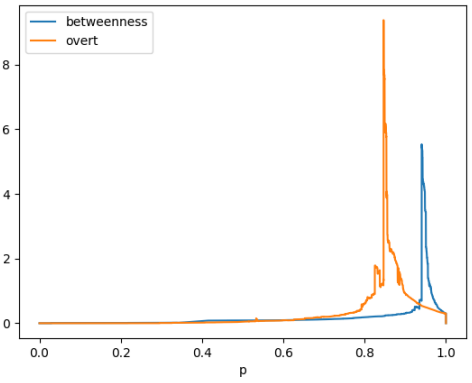
Figure 6.10: Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in all Internet data sets.



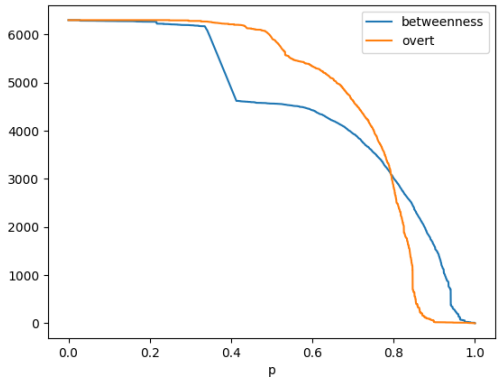
(a) p2p-gnutella08 [47] Susceptibility Index Comparison



(b) p2p-gnutella08 [47] Size of Largest Connected Component

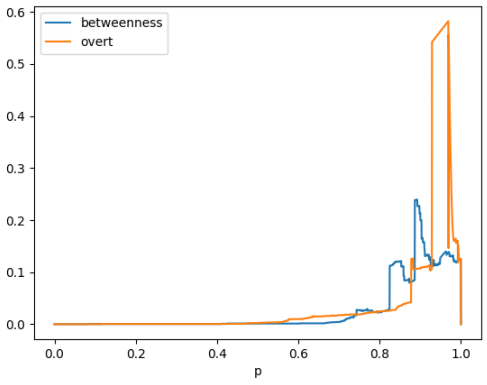


(c) p2p-gnutella09 [47] Susceptibility Index Comparison

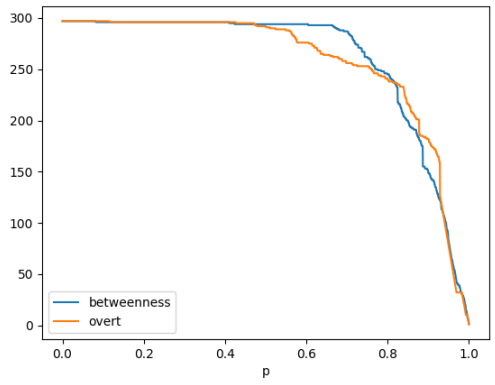


(d) p2p-gnutella09 [47] Size of Largest Connected Component

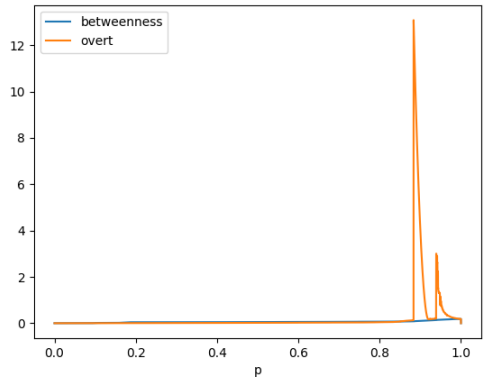
Figure 6.11: Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in all Internet data sets.



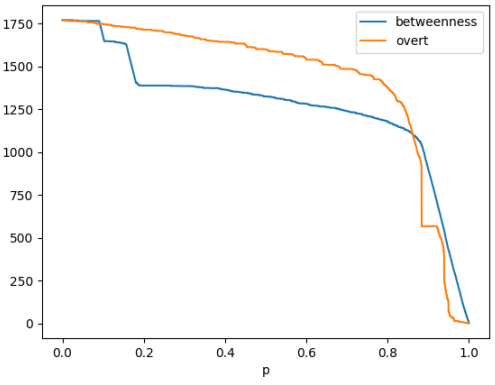
(a) C. Elegans [40] Susceptibility Index Comparison



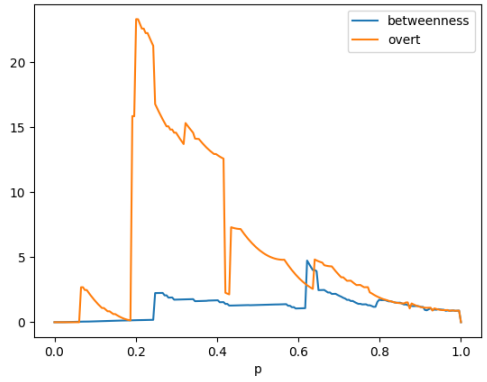
(b) C. Elegans [40] Size of Largest Connected Component



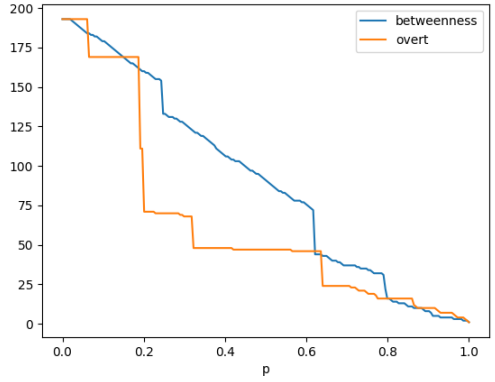
(c) Drosophila Medilla 1 [58] Susceptibility Index Comparison



(d) Drosophila Medilla 1 [58] Size of Largest Connected Component

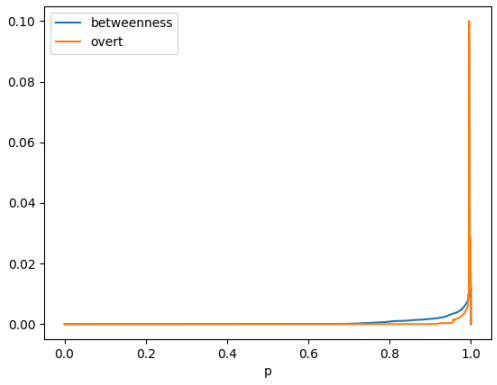


(e) Mouse Visual Cortex 2 [58] Susceptibility Index Comparison

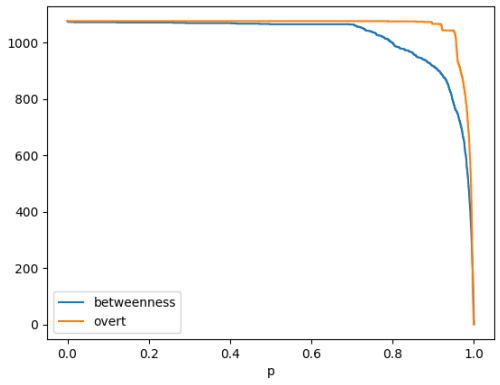


(f) Mouse Visual Cortex 2 [58] Size of Largest Connected Component

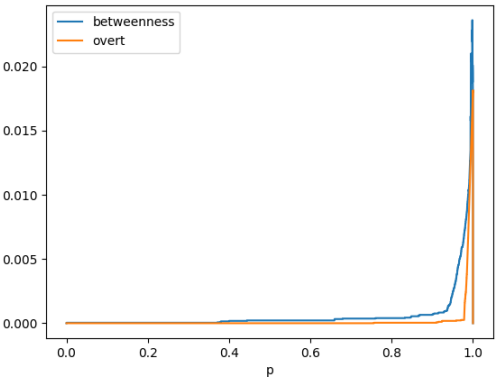
Figure 6.12: Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in all Neural data sets.



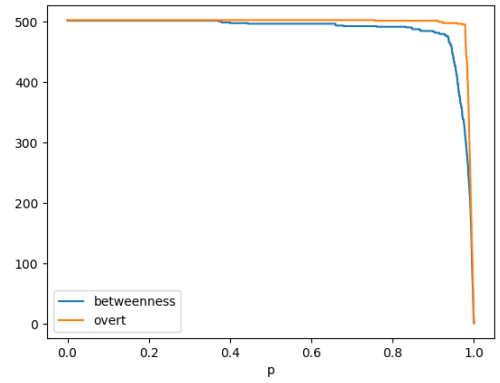
(a) Mouse Retina 1 [58] Susceptibility Index Comparison



(b) Mouse Retina 1 [58] Size of Largest Connected Component

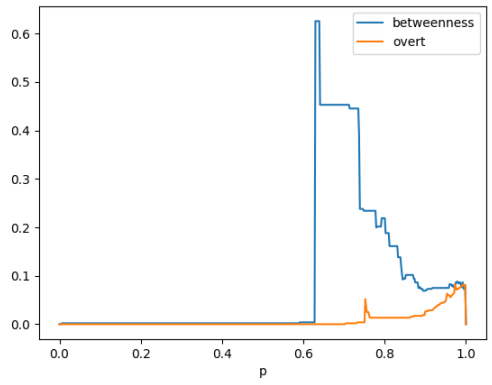


(c) Rattus Norvegicus [58] Susceptibility Index Comparison

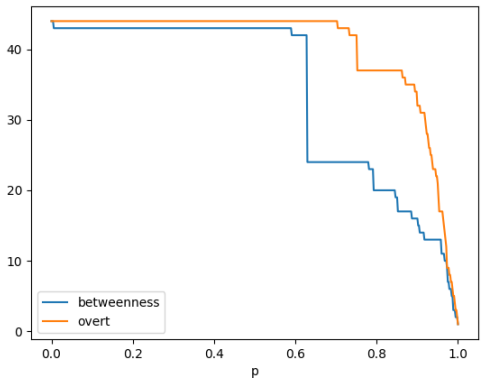


(d) Rattus Norvegicus [58] Size of Largest Connected Component

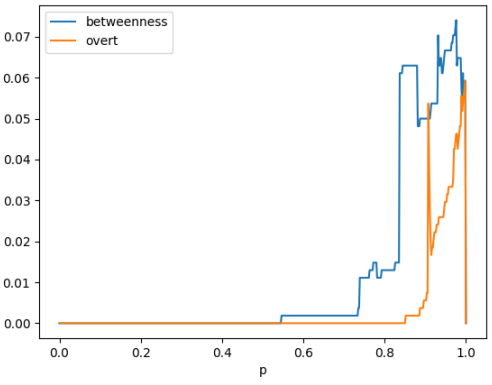
Figure 6.13: Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in all Neural data sets.



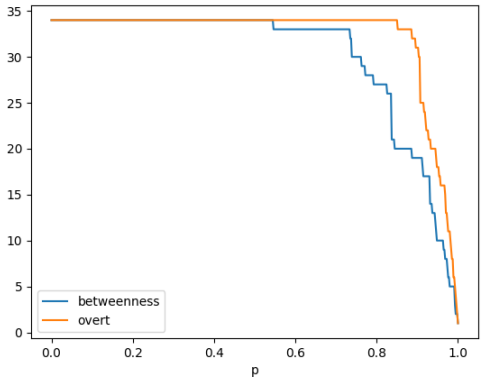
(a) Cross Parker Consulting [63] Susceptibility Index Comparison



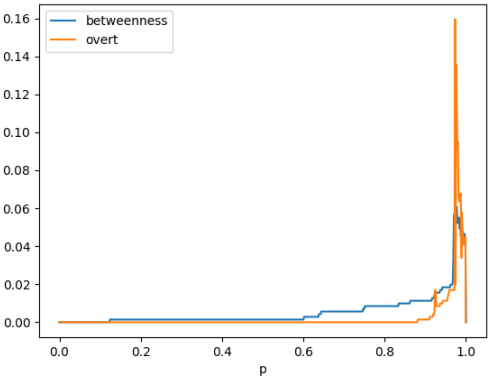
(b) Cross Parker Consulting [63] Size of Largest Connected Component



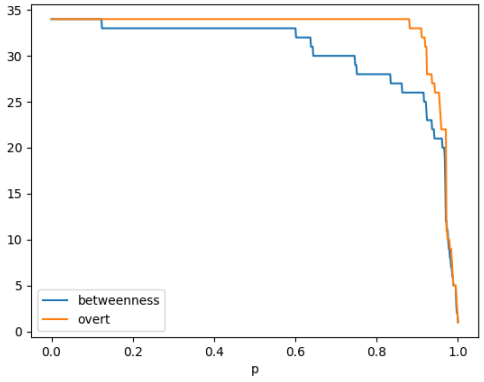
(c) Freemans EIES n48 1 [63] Susceptibility Index Comparison



(d) Freemans EIES n48 1 [63] Size of Largest Connected Component

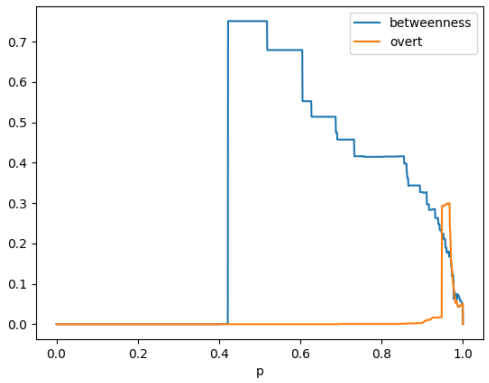


(e) Freemans EIES n48 2 [63] Susceptibility Index Comparison

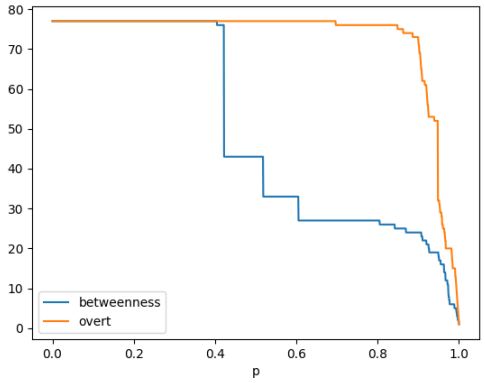


(f) Freemans EIES n48 2 [63] Size of Largest Connected Component

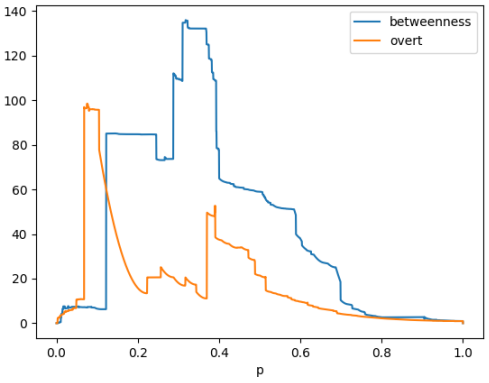
Figure 6.14: Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in all Organise data sets.



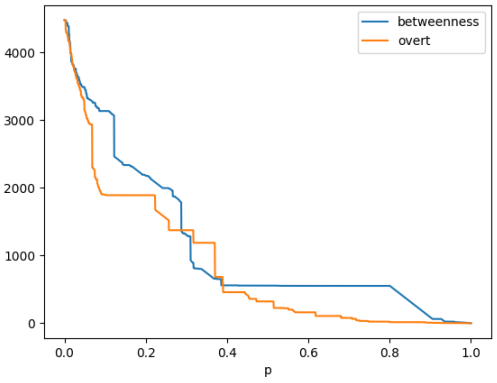
(a) Cross Parker Manufacturing [63] Susceptibility Index Comparison



(b) Cross Parker Manufacturing [63] Size of Largest Connected Component

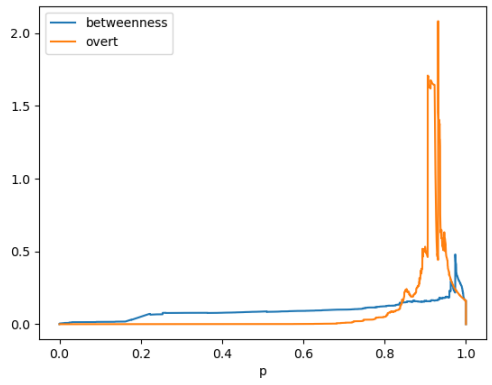


(c) Eva [41] Susceptibility Index Comparison

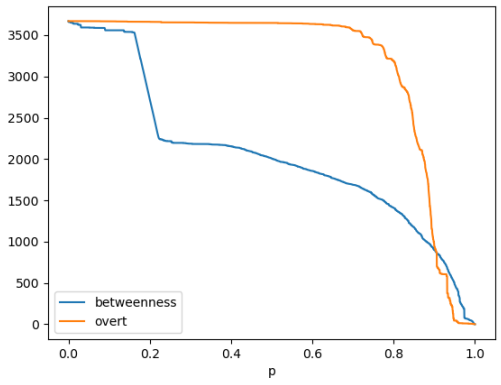


(d) Eva [41] Size of Largest Connected Component

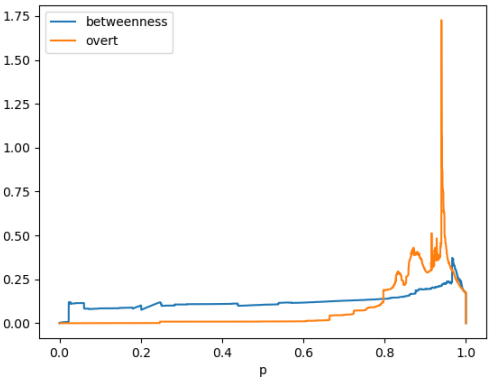
Figure 6.15: Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in all Organise data sets.



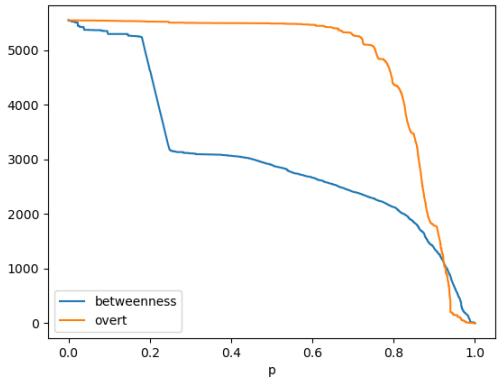
(a) Bitcoin Alpha [47] Susceptibility Index Comparison



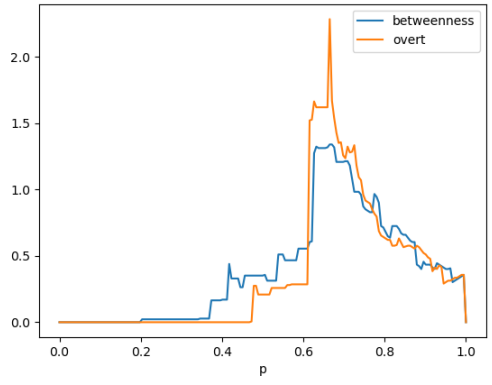
(b) Bitcoin Alpha [47] Size of Largest Connected Component



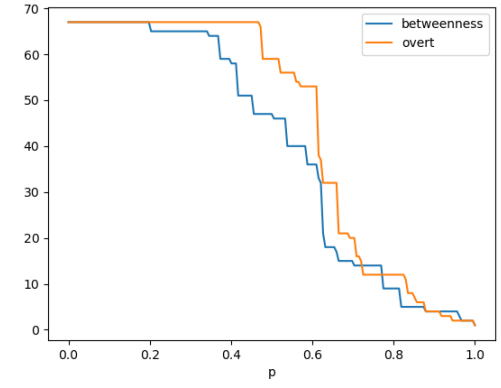
(c) Bitcoin OTC [47] Susceptibility Index Comparison



(d) Bitcoin OTC [47] Size of Largest Connected Component

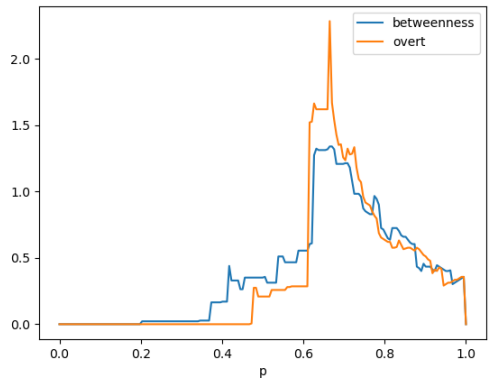


(e) Email EU Core [47] Susceptibility Index Comparison

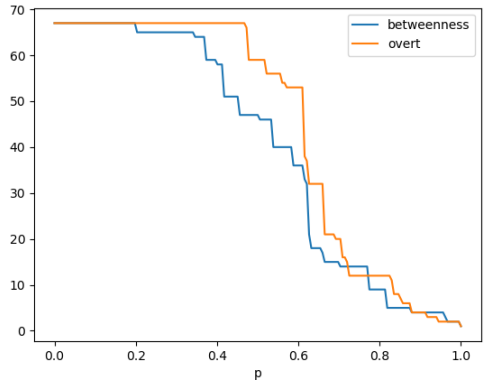


(f) Email EU Core [47] Size of Largest Connected Component

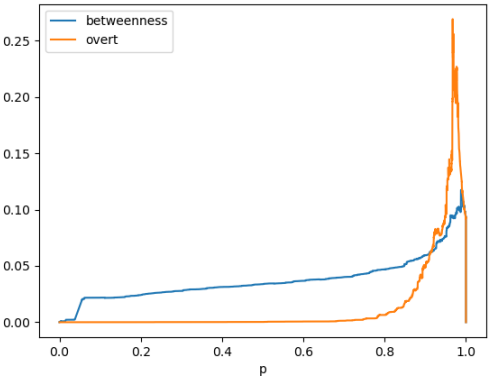
Figure 6.16: Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in all Social data sets.



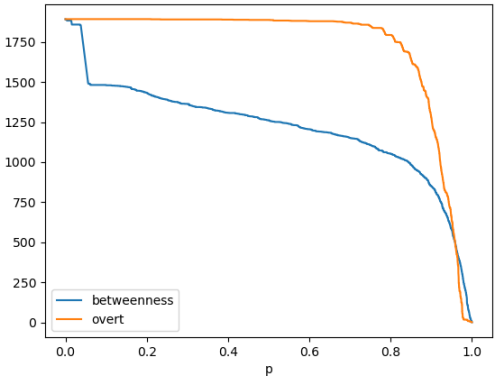
(a) Prison Inmate [5] Susceptibility Index Comparison



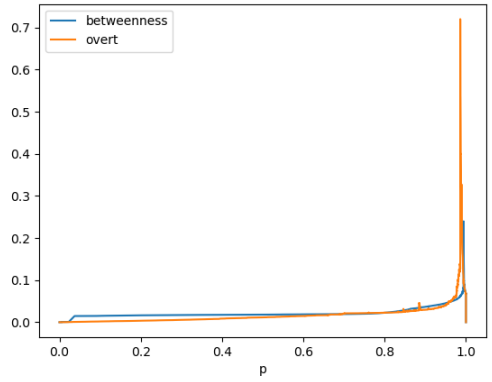
(b) Prison Inmate [5] Size of Largest Connected Component



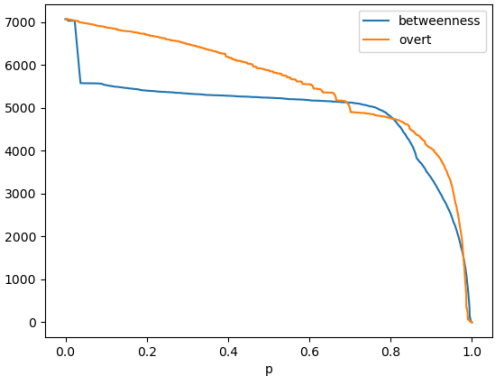
(c) UCIrvine [63] Susceptibility Index Comparison



(d) UCIrvine [63] Size of Largest Connected Component

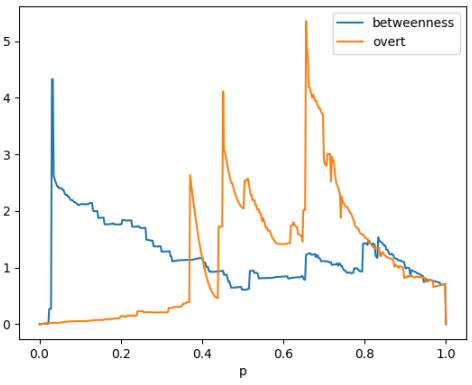


(e) WikiVote [47] Susceptibility Index Comparison

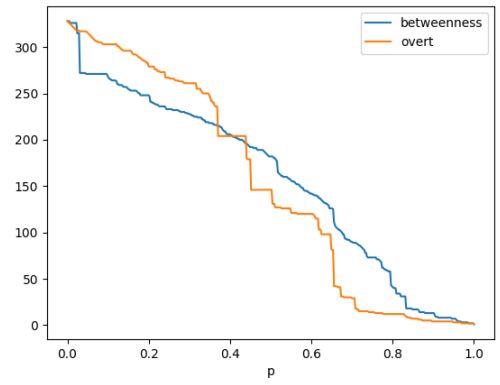


(f) WikiVote [47] Size of Largest Connected Component

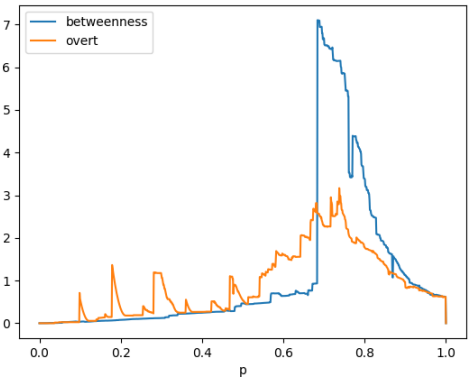
Figure 6.17: Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in all Social data sets.



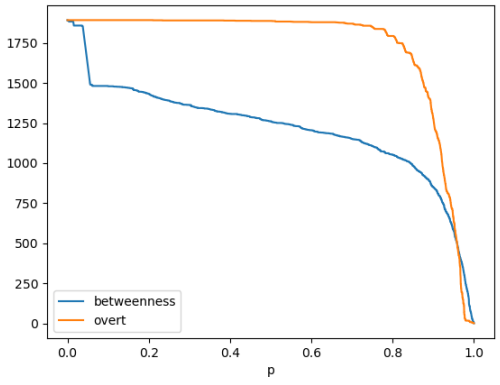
(a) E. coli transcription [5] Susceptibility Index Comparison



(b) E. coli transcription [5] Size of Largest Connected Component

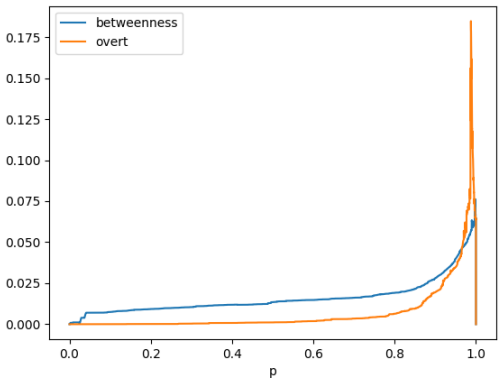


(c) Yeast transcription [5] Susceptibility Index Comparison

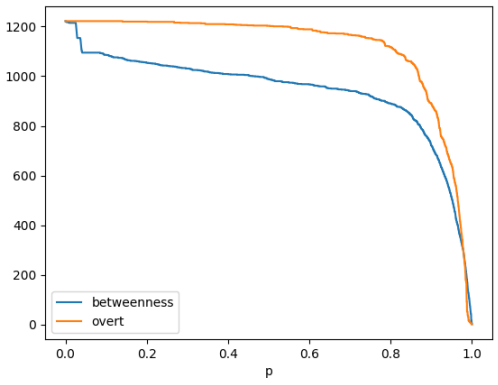


(d) Yeast transcription [5] Size of Largest Connected Component

Figure 6.18: Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in all Regulatory data sets.



(a) Political Blogs[2] Susceptibility Index Comparison



(b) Political Blogs[2] Size of Largest Connected Component

Figure 6.19: Susceptibility Index $S(G)$ (left) and Size of Largest Weakly Connected Component $\sigma(G)$ (right) against the proportion of edges removed, p, in WWW data set.

6.5 Summary

In this chapter we have assessed the 34 real world networks from an alternative perspective: through defining the combined overt and covert centrality as a criticality metric (OCC) to highlight the importance of edges to the structure of the network. To do this, we rank the edges in descending order (highest first) of their combined overt and covert centrality and remove edges, repeatedly recording the susceptibility index and size of largest connected component. We compare these results with an alternative measure for edge criticality, namely the betweenness centrality for edges. We find that in some cases OCC provides a stronger criticality metric than betweenness centrality for edges, as removing edges with a higher OCC first causes a faster disintegration in the network (as in the Food web network Ythan [1]), a greater level of damage to a system (as in the Airport network Open Flights [67]) or both (as seen across the Internet networks). Betweenness centrality for edges highlights which edges contain the intersection of most shortest paths, often acting as bridge-like connectors. Edges which have a high betweenness centrality may represent strong ties (as seen in networks that obey the

global efficiency principle [30, 50]) or weak link (as seen in networks that obey the weak ties theory [32, 61]. Thus, if our metric OCC outperforms betweenness centrality for edges as a criticality metric in some cases then OCC is a potentially useful measure for criticality. Further, betweenness centrality for edges is computationally complex, since to compute betweenness centrality requires knowledge of all shortest paths across a network. In contrast, OCC only requires knowledge of the edges at most two hops away from the edge we are measuring, and therefore offers a more local alternative than betweenness centrality which captures edge criticality reasonably well.

This chapter addresses the final research question:

RQ3: The role of edges of a triad in paths: combining RQ1 and RQ2, how can the edges which enable connectivity within the triad affect paths, and thereby connectivity within a graph?

We do this through applying overt and covert centrality of edges as a criticality metric and considering both simultaneously: **C5:** *A new locally derived criticality metric for edges based on the centrality metric in C2 to observe the importance of an edge with respect to maintaining overall network connectivity. This supports research question* **RQ3**.

Chapter 7

Final Conclusions and Future Work

In this chapter we review the contributions made in each chapter and relate them back to the research questions introduced in Chapter One. We also address questions that have arisen as a result of the work in Chapters Three, Four, Five and Six, offering these as future avenues for research with accompanying insights. Finally, we offer our overall observations to conclude the thesis.

7.1 Research Questions and Contributions

In Chapter One we presented the overarching hypothesis motivating our work - namely that *classifying the role of edges in providing connectivity in networks with respect to induced triads, can provide additional insights to conventional graphlet-based analysis, that are beyond standard metrics of network connectivity*.

This hypothesis motivated our investigation of triads and specifically the role of edges within them, and their relationship to connectivity. It is useful to recall the four research questions (RQ1-RQ3) originally introduced in Chapter One, presented as a basis for exploration:

RQ1: The role of triads in paths: if connectivity in a graph is dependent on the existence of paths, then what is the relationship between triads and paths?

RQ2: The role of edges in triad connectivity: if connectivity within the triad is depend-

ent on the arrangement of its edges, then how do we identify which edges are most fundamental to enabling connectivity within the triad?

RQ3: The role of edges of a triad in paths: combining RQ1 and RQ2, how can the edges which enable connectivity within the triad affect paths, and thereby connectivity within a graph?

In this section we discuss how we addressed **RQ1** - **RQ3**, indicating to what extent they have been achieved. We also highlight the significant findings in each chapter and state the relevant contributions that result.

RQ1 is addressed first in Chapter Three, which explores the triads occurring most frequently along shortest paths by adapting a Triadic Census [39] for shortest paths. We proposed two new measures, Path Induced Triadic Census (Definition 22) and Triadic Edge Betweenness Census (Definition 25) in Section 3.2. Triadic Edge Betweenness Census is edge based, whilst Path Induced Triadic Census is node based, allowing us to respectively compare with a node and edge based version of Triadic Census to asses which triads may occur more frequently across shortest paths than occur in the graph in general. We ran our census measures on a selection of networks from Section 2.5, which introduced 34 data sets used throughout Chapters Three to Six.

We found two factors are influential: the maximum number of edges of a triad that can be simultaneously contained in the same shortest path, and the maximum number of shortest paths that overlap on one edge of a triad. The first factor allows us to classify triads into three types: Type I, II and III. Type II triads are triads which are, in general, more prevalent across shortest paths than in the general graph. These triads can contain two edges simultaneously in a single shortest path. Type I triads, in contrast, can contain at most one edge in a single shortest path, and therefore they are contained less frequently in shortest paths than in the overall graph. Although they may occur less frequently across shortest paths, they have a crucial role in facilitating paths between pairs of vertices. Finally, Type III triads can contain two edges simultaneously in a single shortest path and may occur more frequently in shortest paths; but this relies

entirely on the graph structure, which determines the maximum number of shortest paths that overlap on one edge of a triad. As a result of this classification and the introduction of new path based triadic census measures, Chapter Three provides *new triad census measures to establish the proportion of triads that occur more frequently along shortest paths, as compared with the entire network.* This supports research question **RQ1** and represents contribution **C1**.

In Chapter Four, we introduced an edge classification (**RQ2**) based on the edge's ability to disseminate information within a triad (Section 4.2). The term *overt* was introduced to capture an edge which enables local dissemination across the triad, otherwise the edge is called *covert* (Definition 27). Overt and covert offer alternative definitions to those which already exist based on the structure and direction of edges in triads, such as transitivity [75], noting that overt and covert classify an individual edge with respect to a triad, unlike transitivity which classifies triads in their entirety.

We further generalised the concepts of overt and covert edges to account for the multiple roles an edge can simultaneously play in different induced triads. Specifically, in Section 4.4.1 we introduced the overt and covert centrality of an edge, which is computed by summing the number of triads in which an edge acts as overt or covert (Definition 29). Both overt and covert centrality can be computed very simply from the degree or neighbourhood of incident vertices (Propositions 3 and 4). We consider this as a centrality metric with interesting characteristics: one that satisfies the gap in the literature between centrality metrics for edges, and centrality metrics built upon substructures. We also note that our metric can represent the flow of information in a graph, similar to betweenness centrality for edges [29]; yet our measure is entirely local. In contrast, betweenness centrality for edges requires knowledge of all paths in a network. Overt and covert centrality are much more easily computable and scalable due to the local simplicity of their formulation.

We ran our centrality metric on the 34 data sets (first discussed in Section 2.5) and considered the frequency distributions of overt and covert edge weights (Section 4.4.2).

Further, we explored the correlation between overt and covert centrality and other existing metrics that describe the structure and connectivity of a graph (Section 4.5). We concluded that in many cases, the resulting frequency distributions are effective in associating networks originating from same context or domain. Some classes of network also exhibit greater weighting for one particular type of centrality. Further, we found that overt and covert centrality correlate significantly with many existing metrics, and that all significant relationships between overt/covert centrality with existing metrics are at least moderate, though many are strong or very strong. The easily computable nature of overt/covert centrality make them useful proxies for more complex global centrality measures.

Consequently Chapter Four results in two main contributions:

C2: *A new classification for edges based on their role in supporting connectivity within triads.*

C3: *A new local centrality metric for edges based on our new edge classification in* **C2**.

These contributions address research question **RQ2**.

Chapter Five used the edge classification of overt and covert from Chapter Four to observe the role of edges within triads in shortest paths. In Chapter Five, we weighted paths based on their total overt or covert centrality (Definition 32 and 33 in Section 5.2.1). A path that minimises overt edge centrality supports reduced overall dissemination across a graph; whilst minimising covertness will maximise spread.

By looking at specific paths (those involving minimal edges, those involving minimal overt weight and those involving minimal covert weight) we constructed four new trade-off measures. In Section 5.2.3 we introduced the overt-length (Definition 37) and covert-length (Definition 38) trade-off assessments. These capture the number of additional edges that result when prioritising minimised overtness or covertness over path length. In Section 5.2.4 we introduced length-overt (Definition 43) and length-covert (Definition 44) trade-off assessments. These capture the decrease in path overt

or covert weight when prioritising minimising overtness/covertness over path length. By finding the average of the non-zero trade-offs and normalising, we constructed the *global improvement in edges* as a measure for each each type of trade-off (Definitions 41, 42, 45 and 46). This global measure allowed us to observe the difference in the potential for spread of a message across a network when repeatedly choosing paths of minimum overtness/covertness as compared with shortest paths.

In Section 5.3 we applied our metrics to synthesised networks so that we could assess the effect of changing edge density, specifically random ER networks (Section 5.3.2) and scale free constructions (Section 5.3.3). This allowed us to observe what happens to the global improvement in edges as we vary density in the network. ER-networks allow us an unconstrained range of edge density, and so particular consideration was given to these types of networks. From this we were able to determine some characteristics of the paths being compared, as well as understanding how characteristics of a graph influence the various trade-offs we have constructed.

We conclude that the biggest difference between paths of minimum overt weight and paths with a minimum number of edges occur when a graph has low density. Here, the minimum length paths are short and have moderate overt weight. The biggest difference between paths of minimum covert weight and paths with a minimum number of edges occur when a graph has high density, and the minimum length paths are single edges and have moderate overt weight.

Consequently Chapter Five has resulted in the following contribution: **C4:**

A new method to understand the potential spread of a message through a local centrality metric (from **C2***) for path problems in networks.*

This also supports research question **RQ3.**

Finally, in Chapter Six we explore the simultaneous consideration of overt and covert centrality as an alternative measure for edge criticality. Edge criticality measures are used to highlight how important an edge is in terms of a graph's structure: the removal

of a critical edge will cause the graph to break down into a greater number of connected components. Consequently overt and covert centrality can have a role to play in making such assessments. Building on the definitions of overt and covert centrality from Chapter Four, we considered overt and covert centrality simultaneously, using a single edge criticality metric (OCC). This allowed us to profile networks using a multi-dimensional KDE plot of this paired measure, revealing new insights into the underlying criticality of alternative networks.

As OCC is based on overt and covert centrality, it is locally defined and easily computable meaning it is scalable. We find evidence to suggest it can be used as a reasonable proxy to some global (and computationally expensive) centrality metrics, such as betweenness centrality for edges [29].

In Section 6.4 we ran our criticality metric on the 34 data sets from Section 2.5 and compared them with betweenness centrality for edges. In order to do this, we ranked edges in descending order according to their OCC value and removed them sequentially, recalculating the susceptibility index and size of largest connected component upon each edge removal. We observed the affect on susceptibility index and size of largest connected component, looking for the point at which a network disintegrates into a greater number of connected components. We concluded that our criticality metric performs well, often outperforming betweenness centrality for edges, by causing either a more rapid network dis-connection, or a greater amount of damage (i.e separates the network into more connected components) to a network. Chapter Six has resulted in the following contribution:

C5: *A new locally derived criticality metric for edges based on the centrality metric in C2 to observe the importance of an edge with respect to maintaining overall network connectivity. This supports research question* **RQ3**.

7.2 Future Research

In this section we discuss possible future research that has been motivated by observations made during the course of this thesis. The concepts of overt/covert edges, and associated definitions of centrality and criticality lead us to consider ways in which further research and development could take place. In particular, these relate to *temporal networks, the importance of particular nodes,* and *more general induced substructures*. We present each of these issues as a possible future research direction.

Future Direction 1 - *The concepts of overt and covert edges be useful in examining the behaviour of temporal networks and temporal motifs.*

In order to introduce and develop overt and covert edges and related concepts, we have necessarily considered them in the context of static networks. However, in many complex network scenarios, edges are transient with a temporal pattern of behaviour.

In Chapter Four (Section 4.4.2) we compare overt and covert frequency profiles of different networks, exploring how factors such as the effect of changing network density effect the frequency of triads in which an edge acts as overt or covert. We noted that adding edges to a network can cause the number of triads in which an edge is overt or covert to increase through inducing new triads; or the nature of an edge can be changed from covert to overt through changing existing triads. The reverse is true for removing edges, that is: the number of triads in which an edge is overt or covert can reduce through breaking triads into dyads, or an edge can change from overt to covert through changing remaining triads. In Section 5.3.1 we also observed the effects of increasing the overt or covert centrality of edges on the paths they are contained in.

Therefore incorporating such changes in the context of overt and covert edges is a useful future directions. There are different options to consider. For example, modifying the definition of edges through making them dynamic (i.e subject to change), is one option. Alternatively a starting point could be building on the definition of *temporal motifs* [65]. Here, Paranjape and Benson introduce these as induced subgraphs on sequences

of temporal edges in a network. Therefore it is possible that overt and covert edges could be used to track local changes. We have already observed that covert edges can transform into overt edges and this is relatively easy to compute. This type of transformation could be used to indicate changes to overall network connectivity and paths.

Future Direction 2 - *The concepts of overt and covert are edge based. It is possible they can be used to identify vertices that are interesting due to distinctive positioning within a network.*

By definition, overt and covert edges relate pairs of nodes together and classify this relationship within a triad. It may be possible to build intelligence concerning nodes based on their incidence with different overt and covert edges. A simple example to build on this is given in Figure 7.1 for the triad `030T`.

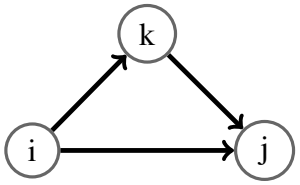


Figure 7.1: 030T

In Figure 7.1 the edge (i, k) is overt, yet (k, j) and (j, k) are covert. Then vertices i and k sit between one overt and covert edge, yet j is adjacent incident with two covert edges. Does j play a different role in the triad than i or k? Certainly, since it is incident with zero overt edges it is unable to relay the message to the rest of the triad. Further, even though i and k are both between an overt and covert edge, do they play different roles to one another due to the relation between the overt and covert edge in question? For example, i is adjacent to two edges which have the power to flood the triad, whereas k is only able to relay a message to j. This motivates considering and assessing nodes taking into account the different roles that they place. Certainly this could have a significant number of practical applications where the nodes are important.

Future Direction 3 - *Generalising overt and covert edges for application in larger induced substructures such as tetrads.*

The work in this thesis is based on triads, the smallest non-trivial substructures beyond an edge. However analysis using larger induced sub-structures such as *tetrads* (induced substructures on four nodes) has become of more interest in the wider literature (e.g., [22]), noting that these substructures can convey latent information beyond considering pairs or triads of nodes. The challenge however, is handling the increased number of such substructures, and understanding the significance of differences between them.

Its therefore relevant to ask whether concepts of overt and covert can be generalised to higher order induced substructures, so that additional insights can be determined for more complex larger structures such as tetrads.

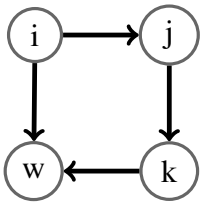


Figure 7.2: Tetrad T

For example, in Figure 7.2, continuing the definitions of overt and covert as they exist for triads, then edges (i, j), (j, k) and (i, w) are overt, and (k, w) covert. However, the definitions of overt and covert were built on the ability of an edge to flood a triad by disseminating a message through adjacent edges: something only the edge (i, j) is capable of supporting in T. This could mean an alternative definition of overt and covert edges in tetrads is necessary, perhaps with more than a binary classification of edges.

7.3 Final Remarks

Traditionally, popular techniques to assess complex networks often concern assessing the relative volume of particular induced triads within a network (such as a Triadic

Census [39], motif profiling [54] or global clustering coefficient [49]). Further, when triads have been used in other ways (such as in centrality measures [19, 66]) these measures do not focus on the roles that *edges* play.

This thesis has taken the idea of using induced triads as a unit of analysis for complex networks, and examined triads locally, from a graph theoretic perspective. The thesis aims to show that despite the simple nature of triads, closer examination of them through the role that their edges play in local connectivity, is useful. Not all edges are equal and there are differences of importance that edges play within triads. In order to fulfil this we have classified edges as *overt or covert* based on their role within a particular triad. This reflects an edge's ability to disseminate information within a triad, and by extension, an entire graph. This approach can also capture the importance of an edge accounting for its role in multiple triads. We believe this is a fundamental categorisation, of value due to the widespread use of triads in complex network analysis.

We also offer this as an entirely local, computationally simple way of determining edge importance within a graph. To exemplify this, we have applied our metric both for edge centrality assessment, and edge criticality assessment. Our measures are comparable to betweenness centrality for edges, in that they represent flow of information in a network, yet our measures do not require knowledge of all the shortest paths in a network. Not only do our measures assess which edges are important to the flow of information in a network, but they can also highlight edges which are important to the structural integrity of a network. In addition to this, we have explored the role of induced triads across shortest paths. We have determined which triads may be of greater importance to shortest paths, and applied our edge centrality metric to shortest paths as an assessment for potential containment or spreading on content from a path. These contributions sit between complex networks and graph theory, with the potential for applicability to real world scenarios while simulating further directions for future fundamental research.

Bibliography

[1] A. R. Cirtwill and A. Eklof. Feeding environment and other traits shape species' roles in marine food webs, 2020.

[2] Lada A. Adamic and Natalie Glance. The political blogosphere and the 2004 U.S. Election: Divided they blog. In *3rd International Workshop on Link Discovery, LinkKDD 2005 - in conjunction with 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 36–43, Chicago, Illinois, 2005. ISBN 1595932151. doi: 10.1145/1134271.1134277.

[3] L.V. Ahn. Thresholds and Collective Action, 2008. URL `http://scienceoftheweb.org/15-396/lectures/lecture07.pdf`.

[4] Tharaka Alahakoon, Rahul Tripathi, Nicolas Kourtellis, Ramanuja Simha, and Adriana Iamnitchi. K-path centrality: A new centrality measure in social networks. In *Proceedings of the 4th Workshop on Social Network Systems, SNS'11*, pages 1–6, 2011. ISBN 9781450307284. doi: 10.1145/1989656.1989657.

[5] U. Alon. Collection of complex networks. URL `https://www.weizmann.ac.il/mcb/UriAlon/download/collection-complex-networks`.

[6] Albert László Barabási and Réka Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, 1999. ISSN 00368075. doi: 10.1126/science.286.5439.509. URL `https://www.science.org/doi/abs/10.1126/science.286.5439.509{%}7D,{%}0Aeprint={%}7Bhttps://www.science.org/doi/pdf/10.1126/science.286.5439.509`.

[7] A. Barrat, M. Barthélemy, R. Pastor-Satorras, and A. Vespignani. The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(11):3747–3752, 2004. ISSN 00278424. doi: 10.1073/pnas.0400087101.

[8] Vladimir Batagelj and Andrej Mrvar. A subquadratic triad census algorithm for large sparse networks with small maximum degree. *Social Networks*, 23(3): 237–243, 2001. ISSN 03788733. doi: 10.1016/S0378-8733(01)00035-1.

[9] Alex Bavelas. Communication Patterns in Task-Oriented Groups. *Journal of the Acoustical Society of America*, 22(6):725–730, 1950. ISSN NA. doi: 10.1121/1.1906679.

[10] J. Benson, A.R. Gleich, D.F Leskovec. Higher-order organization of complex networks. *Science*, 353(6295):163–166, 2016. ISSN 00107514. doi: 10.1126/science.aad9029. URL https://www.science.org/doi/abs/10.1126/science.aad9029{%}7D,{%}0Aeprint={%}7Bhttps://www.science.org/doi/pdf/10.1126/science.aad9029.

[11] Stephen P. Borgatti. Centrality and network flow. *Social Networks*, 27(1):55–71, 2005. ISSN 03788733. doi: 10.1016/j.socnet.2004.11.008.

[12] Ulrik Brandes. On variants of shortest-path betweenness centrality and their generic computation. *Social Networks*, 30(2):136–145, 2008. ISSN 03788733. doi: 10.1016/j.socnet.2007.11.001.

[13] Dorwin Cartwright and Frank Harary. Structural balance: a generalization of Heider's theory. *Psychological Review*, 63(5):277–293, 1956. ISSN 0033295X. doi: 10.1037/h0046049.

[14] COSIN. The cosin network data and analysis. URL http://www.cosinproject.eu/extra/data/foodwebs/WEB.html.

[15] Pasquale De Meo, Emilio Ferrara, Giacomo Fiumara, and Angela Ricciardello. A novel measure of edge centrality in social networks. *Knowledge-Based Systems*, 30:136–150, 2012. ISSN 09507051. doi: 10.1016/j.knosys.2012.01.007.

[16] R. Diestel. *Graph Theory*. New York: Springer-Verlag Heidelberg, 2005. ISBN 978-3-662-53621-6.

[17] E W Dijkstra. Dijkstra.Pptx. *Numer. Math.*, 271:269–271, 1959.

[18] Paul Erds and Alfréd Rényi. On Random graphs I. *Publicationes Mathematicae*, 6:290–297, 1959. URL https://www.bibsonomy.org/bibtex/2420b83c1533188c0b54bd1f6eea2b782/krevelen.

[19] Ernesto Estrada and Juan A. Rodríguez-Velázquez. Subgraph centrality in complex networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 71 (5):1–9, 2005. ISSN 15393755. doi: 10.1103/PhysRevE.71.056103.

[20] Katherine Faust. A puzzle concerning triads in social networks: Graph constraints and the triad census. *Social Networks*, 32(3):221–233, 2010. ISSN 0378-8733. doi: https://doi.org/10.1016/j.socnet.2010.03.004. URL https://www.sciencedirect.com/science/article/pii/S0378873310000158.

[21] Scott L Feld. The Focused Organization of Social Ties. *The American journal of sociology*, 86(5):1015–1035, 1981.

[22] Diane Felmlee, Cassie McMillan, and Roger Whitaker. Dyads, triads, and tetrads: a multivariate simulation approach to uncovering network motifs in social graphs. *Applied Network Science*, 6(1), 2021. ISSN 23648228. doi: 10.1007/s41109-021-00403-5. URL https://doi.org/10.1007/s41109-021-00403-5.

[23] Diane Felmlee, Cassie McMillan, and Roger Whitaker. Dyads, triads, and tetrads: a multivariate simulation approach to uncovering network motifs in social graphs. *Applied network science*, 6(1):1–26, 2021. ISSN 2364-8228.

[24] Alex Fornito, Andrew Zalesky, and Edward T. Bullmore. Paths, Diffusion, and Navigation. In *Fundamentals of Brain Network Analysis*, pages 207–255. 2016. ISBN 9780124079083. doi: 10.1016/b978-0-12-407908-3.00007-8.

[25] Santo Fortunato, Vito Latora, and Massimo Marchiori. Method to find community structures based on information centrality. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 70(5):13, 2004. ISSN 1063651X. doi: 10.1103/PhysRevE.70.056104.

[26] L. C. Freeman. A Set of Measures of Centrality Based on Betweenness. Sociometry. 40(1):35–41, 1977. doi: 10.2307/3033543. URL https://www.jstor.org/stable/3033543.

[27] L. C. Freeman. Centrality in social networks. *Social Networks*, 1(3):215–239, 1979. ISSN 03788733. doi: 10.1016/0378-8733(78)90021-7.

[28] Linton C. Freeman, Stephen P. Borgatti, and Douglas R. White. Centrality in valued graphs: A measure of betweenness based on network flow. *Social Networks*, 13(2):141–154, 1991. ISSN 03788733. doi: 10.1016/0378-8733(91)90017-N. URL https://www.sciencedirect.com/science/article/pii/037887339190017N.

[29] M. Girvan and M. E.J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821–7826, 2002. ISSN 00278424. doi: 10.1073/pnas.122653799.

[30] D. Goh, K.-I. and Kahng, B. and Kim. Universal Behavior of Load Distribution in Scale-Free Networks. *Phys. Rev. Lett.*, 87(27):278701, 2001. doi: 10.1103/PhysRevLett.87.278701.

[31] R. E. Gomory and T. C. (1961) Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics.*, 83(4):551–570, 1961. doi: 10.1137/0109047.

[32] Mark S Granovetter. The Strength of Weak Ties. *American Journal of Sociology*, 78(6):1360–1380, 1973. URL http://www.jstor.org/stable/2776392.

[33] A A Hagberg, D A Schult, and P J Swart. Exploring network structure, dynamics, and function using NetworkX. In *7th Python in Science Conference (SciPy 2008)*, Los Alamos, NM (United States), 2008. URL https://www.osti.gov/biblio/960616.

[34] Fritz Heider. The Psychology of Interpersonal Relations. *The Psychology of Interpersonal Relations*, 37(3):322, 1958. doi: https://doi.org/10.2307/2572978.

[35] Tomaž Hočevar and Janez Demčar. Computation of Graphlet Orbits for Nodes and Edges in Sparse Graphs. *Journal of Statistical Software*, 71(10), 2016. ISSN 1548-7660. doi: 10.18637/jss.v071.i10.

[36] Tomaž Hočevar and Janez Demšar. Combinatorial algorithm for counting small induced graphs and orbits. *PLoS ONE*, 12(2):1–17, 2017. ISSN 19326203. doi: 10.1371/journal.pone.0171428.

[37] P.W. Holland and S. Leinhardt. Transitivity in Structual Models of Small Groups. *Comparative Group Studies*, 2(2):107–124, 1971. doi: https://doi.org/10.1177/104649647100200201.

[38] S. Holland, P. W., & Leinhardt. A Method for Detecting Structure in Sociometric Data. *American Journal of Sociology*, 76(3):492–513, 1970. URL http://www.jstor.org/stable/2775735.

[39] S. Holland, P. W., & Leinhardt. Local Structure in Social Networks. *Sociological Methodology*, 7(1976):1–45, 1976. doi: https://doi.org/10.2307/270703. URL https://www.jstor.org/stable/270703.

[40] J.D. Johnson. Ucinet: A software tool for network analysis, 1987. URL http://www.analytictech.com/ucinet/description.htm.

[41] John Chuang Kim Norlen , Gabriel Lucas , Michael Gebbie. EVA: Extraction, Visualization and Analysis of the Telecommunications and Media Ownership Network. In *Proceedings of International Telecommunications Society 14th Biennial Conference*, Seoul Korea, 2002. CiteSeer. doi: 10.1.1.210.6691.

[42] Jérôme Kunegis. KONECT - The koblenz network collection. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1343–1350, Rio de Janeiro, Brazil, 2013. Association for Computing Machinery. ISBN 9781450320382. doi: 10.1145/2487788.2488173. URL https://doi.org/10.1145/2487788.2488173.

[43] V. Latora and M. Marchiori. Economic small-world behavior in weighted networks. *European Physical Journal B*, 32(2):249–263, 2003. ISSN 14346028. doi: 10.1140/epjb/e2003-00095-5.

[44] V. Latora and M. Marchiori. A measure of centrality based on network efficiency. *New Journal of Physics*, 9(6):188, 2007. ISSN 13672630. doi: 10.1088/1367-2630/9/6/188.

[45] Vito Latora and Massimo Marchiori. Efficient behavior of small-world networks. *Physical Review Letters*, 87(19):198701–1–198701–4, 2001. ISSN 10797114. doi: 10.1103/PhysRevLett.87.198701.

[46] Vito Latora and Massimo Marchiori. Vulnerability and Protection of Critical Infrastructures. pages 11–14, 2004. doi: 10.1103/PhysRevE.71.015103. URL `http://arxiv.org/abs/cond-mat/0407491{%}0Ahttp://dx.doi.org/10.1103/PhysRevE.71.015103`.

[47] A. Leskovec, J. and Krevl. SNAP: A General-Purpose Network Analysis and Graph-Mining Library. URL `http://snap.stanford.edu/data`.

[48] J.M Lopez-Fernandez, L ;Robles, G ; Gonzalez-Barahona. Applying social network analysis to the information in CVS repositories. In *26th International Conference on Software Engineering - W17S Workshop "International Workshop on Mining Software Repositories (MSR 2004)*, pages 101–105, 2004. doi: 10.1049/ic:20040485.

[49] R. Duncan Luce and Albert D. Perry. A method of matrix analysis of group structure. *Psychometrika*, 14(2):95–116, 1949. ISSN 00333123. doi: 10.1007/BF02289146.

[50] A. Maritan, F. Colaiori, A. Flammini, J. R. Banavar, and M. Cieplak. Universality Classes of Optimal Channel Networks. *Science*, 272(5264):984–986, 1996. doi: 10.1126/science.272.5264.984.

[51] Giovanni Mastrobuoni and Eleonora Patacchini. Organized crime networks: An application of network analysis techniques to the American Mafia. *Review of Network Economics*, 11(3), 2012. ISSN 14469022. doi: 10.1515/1446-9022.1324.

[52] Cassie McMillan and Diane Felmlee. Beyond dyads and triads: A comparison of tetrads in twenty social networks. *Social psychology quarterly*, 83(4):383–404, 2020. ISSN 0190-2725.

[53] S. Milgram. The small world problem. *Psycology Today*, 2:60–67, 1967. ISSN 00134252.

[54] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network Motifs: Simple Building Blocks of Complex Networks. *The Structure and Dynamics of Networks*, 9781400841(October):217–220, 2011. doi: 10.1515/9781400841356.217.

[55] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network Motifs: Simple Building Blocks of Complex Networks. *The Structure and Dynamics of Networks*, 9781400841(October):217–220, 2011. doi: 10.1515/9781400841356.217.

[56] U. Milo, R. Itzkovitz, S. Kashtan, N. Levitt, R. Alon. Superfamilies of Evolved and Designed Networks. *Science*, 308(APRIL):1109723–1109723, 2005. URL `http://www.jstor.org/stable/3836575`.

[57] Enys Mones, Lilla Vicsek, and Tamás Vicsek. Hierarchy measure for complex networks. *PLoS ONE*, 7(3):1–10, 2012. ISSN 19326203. doi: 10.1371/journal.pone.0033799.

[58] NeuroData. Animal connectomes. URL `https://neurodata.io/project/connectomes/`.

[59] M. E.J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003. ISSN 00361445. doi: 10.1137/S003614450342480.

[60] J. Newman, M. E. J. and Forrest, S. and Balthrop. Email networks and the spread of computer viruses. *Phys. Rev. E*, 66(3):4, 2002. doi: 10.1103/PhysRevE.66.035101. URL `https://link.aps.org/doi/10.1103/PhysRevE.66.035101`.

[61] J. P. Onnela, J. Saramäki, J. Hyvönen, G. Szabó, D. Lazer, K. Kaski, J. Kertész, and A. L. Barabási. Structure and tie strengths in mobile communication networks. *Proceedings of the National Academy of Sciences of the United States of America*, 104(18):7332–7336, 2007. ISSN 00278424. doi: 10.1073/pnas.0610245104.

[62] Jukka Pekka Onnela, Jari Saramäki, János Kertész, and Kimmo Kaski. Intensity and coherence of motifs in weighted complex networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 71(6):065103–065103, 2005. ISSN 15393755. doi: 10.1103/PhysRevE.71.065103.

[63] T. Opsahl. Network datasets. URL `https://toreopsahl.com/datasets/`.

[64] Tore Opsahl and Pietro Panzarasa. Clustering in weighted networks. *Social Networks*, 31(2):155–163, 2009. ISSN 03788733. doi: 10.1016/j.socnet.2009.02.002.

[65] Ashwin Paranjape, Austin R. Benson, and Jure Leskovec. Motifs in temporal networks. *WSDM 2017 - Proceedings of the 10th ACM International Conference on Web Search and Data Mining*, pages 601–610, 2017. doi: 10.1145/3018661.3018731.

[66] Nataša Pržulj. Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):177–183, 2007. ISSN 14602059. doi: 10.1093/bioinformatics/btl301.

[67] N. K. Rossi, R. A. and Ahmed. The Network Data Repository with Interactive Graph Analytics and Visualization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 4292–4293, Austin, Texas, 2015. AAAI Press. doi: 0262511290. URL `http://networkrepository.com`.

[68] S Borgatti, M. G. Everett, and L. C. Freeman. UCINET 5 for Windows: Software for Social Network Analysis. *Natick: Analytic Technologies.*, (January), 2002.

[69] Hua Wei Shen, Xue Qi Cheng, and Jia Feng Guo. Quantifying and identifying the overlapping community structure in networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(7):1–9, 2009. ISSN 17425468. doi: 10.1088/1742-5468/2009/07/P07042.

[70] Huawei Shen, Xueqi Cheng, Kai Cai, and Mao Bin Hu. Detect overlapping and hierarchical community structure in networks. *Physica A: Statistical Mechanics and its Applications*, 388(8):1706–1712, 2009. ISSN 03784371. doi: 10.1016/j.physa.2008.12.021.

[71] Shai S. Shen-Orr, Ron Milo, Shmoolik Mangan, and Uri Alon. Network motifs in the transcriptional regulation network of Escherichia coli. *Nature Genetics*, 31(1):64–68, 2002. ISSN 10614036. doi: 10.1038/ng881.

[72] Peng Gang Sun and Yang Yang. Methods to find community based on edge centrality. *Physica A: Statistical Mechanics and its Applications*, 392(9):1977–1988, 2013. ISSN 03784371. doi: 10.1016/j.physa.2012.12.024. URL `http://dx.doi.org/10.1016/j.physa.2012.12.024`.

[73] Andreia Sofia Teixeira, Pedro T Monteiro, João A Carriço, Mário Ramirez, and Alexandre P Francisco. Spanning edge betweenness. In *Eleventh Workshop on Mining and Learning with Graphs*, number 24, 2013. ISBN 9781450323222.

[74] V. Batagelj and A. Mrvar. Pajek datasets. URL `http://vlado.fmf.uni-lj.si/pub/networks/data/bio/foodweb/foodweb.htm`.

[75] Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*, volume 36. 1995. ISBN 0521382696. doi: 10.2307/3322457.

[76] DJ Watts and SH Strogatz. Collective dynamics of Small World Networks. *Nature*, 393(6684):440–442, 1998. ISSN 0028-0836.

[77] Cheng Xue-Qi, Ren Fu-Xin, Shen Hua-Wei, Zhang Zi-Ke, and Zhou Tao. Bridgeness: A local index on Edge significance in maintaining global connectivity. *Journal of Statistical Mechanics: Theory and Experiment*, 2010(10):1–10, 2010. ISSN 17425468. doi: 10.1088/1742-5468/2010/10/P10011.

[78] Cheng Xue-Qi, Ren Fu-Xin, Shen Hua-Wei, Zhang Zi-Ke, and Zhou Tao. Bridgeness: A local index on Edge significance in maintaining global connectivity.

Journal of Statistical Mechanics: Theory and Experiment, 2010(10), 2010. ISSN 17425468. doi: 10.1088/1742-5468/2010/10/P10011.

[79] En Yu Yu, Duan Bing Chen, and Jun Yan Zhao. Identifying critical edges in complex networks. *Scientific Reports*, 8(1):1–8, 2018. ISSN 20452322. doi: 10.1038/s41598-018-32631-8.

[80] B; Zhang and S. Horvath. A general framework for weighted gene co-expression network analysis. *Statistical Applications in Genetics and Molecular Biology*, 4 (1), 2005. doi: 10.2202/1544-6115.1128.