

Quantitative Analyse von Netzwerk-Massen

Usefulness und weitere Aspekte



Bachelor-Thesis im Studiengang Informatik

von

Luca Hostettler

Eingereicht bei:

Prof. Dr. Tobias Häberlein
Departement Informatik
Departementsleiter

Referent:

Ao. Prof. Dr. habil. Matthias Dehmer
Departement Informatik
Dozent Data Science

Zürich, 15. März 2023

Diese Arbeit widme ich meiner Familie und meiner Partnerin Nathalie, welche stets an meiner Seite standen und mich unterstützt haben. Ohne ihre Hilfe wäre dies nicht möglich gewesen ❤.

Mein Dank gilt meinem Betreuer Prof. Dr. Matthias Dehmer, welcher mich vor 2 Jahren in den Bereich der Graphentheorie eingeführt hat und mir diese spannenden Reise ermöglicht hat.

Ebenso bedanken will ich mich bei Marc, ein Sparringpartner grosser Klasse und ein wahrer Freund !

Zusammenfassung

In diesem Forschungsprojekt wurden nützliche topologische Indizes für bestimmte Klassen und Strukturen von Graphen im Bereich der Graphentheorie untersucht. Durch die Berechnung diverser Indizes für eine Reihe von Graphen und den Vergleich ihrer Leistung wurde in der Studie ermittelt, welche Indizes für verschiedene Graphenklassen am nützlichsten sind.

Auf der Grundlage dieser Erkenntnisse wurde eine Anwendung entwickelt, die anhand eines Eingabagraphen den einflussreichsten topologischen Index für die jeweilige Klasse bestimmt. Diese Anwendung hat das Potenzial, ein wertvolles Werkzeug für Forscher und Praktiker zu sein, die mit Graphdaten arbeiten, und könnte für ein breites Spektrum eingesetzt werden, darunter molekulare, bioinformatische, soziale und synthetische Graphen. Insgesamt bieten die Ergebnisse dieser Studie wesentliche Einblicke in die Verwendung von topologischen Indizes in der Graphenanalyse und können Forschung und Praxis in diesem Bereich potenziell beeinflussen.

Keywords— Graphentheorie, Topologische Indizes, Statistische Analyse, Graphenklassifikation

Abstract

This thesis presents the findings of a comprehensive study on the usefulness of topological indices for specific classes and structures of graphs in graph theory. Using a variety of graphs, including molecular, bioinformatics, social, and synthetic graphs, this study calculated a range of topological indices and compared their performance on each graph class.

The results indicated that different indices had varying degrees of influence on the principal components of graph classes, with some indices being more useful than others for particular classes of graph.

Based on these findings, the study developed an application that classifies an input graph into one of the four graph classes. The application then returns the most influential topological index for the given class, as determined by the results of the principal component analysis. The findings of this study have significant implications for the use of topological indices in a wide range of applications and provide valuable insights for researchers and practitioners working with graph data.

Keywords— Graph theory, Comparative analysis, Topological indices, Principal component analysis, Graph classification

Inhaltsverzeichnis

Abkürzungen	vii
1. Einleitung	1
1.1. Herleitung	1
1.1.1. Stand der Forschung	1
1.1.2. Wissenschaftlicher Bezug	3
1.2. Zentrale Fragestellung	3
2. Theorie	5
2.1. Was ist Graphentheorie?	6
2.2. Was ist ein Graph?	7
2.3. Netzwerkklassen	10
2.3.1. Random	11
2.3.2. Small World	12
2.3.3. Scale-free	13
2.3.4. Bäume	13
2.3.5. Vollständig	15
2.3.6. Path	16
2.4. Topologische Indizes	17
2.4.1. Anwendung von topologischen Indizes	18
2.4.2. Gradbasiert	20
2.4.3. Distanzbasiert	22
2.4.4. Zentrische Indizes	23
2.4.5. Informationstheoriebasiert	24
2.5. Graphenklassifikation	26
2.5.1. Klassifikation auf Basis von Isomorphismus	26
2.5.2. Klassifizierung durch strukturelle Masse	27
2.5.3. Machine Learning – Support Vector Machine	27
2.5.4. Machine Learning – Neural Networks	28
2.5.5. Unterschied von Graph-Embedding und Graph-Kernels	29
3. Eigene Resultate	30
3.1. Erwartete Resultate	31

Inhaltsverzeichnis

3.2. Datenaufbereitung	33
3.2.1. Simulation	33
3.2.2. Netze und Klassen	34
3.2.3. Datenaufbereitung und Datenstruktur	35
3.3. Vergleich der topologischen Indizes	36
3.3.1. Erste Erkenntnisse und Resultate	36
3.3.2. Vergleich der topologischen Indizes	38
3.3.3. Korrelation der topologischen Indizes	40
3.3.4. Principal Component Analysis	44
3.4. Klassifizierung der Graphen	54
3.4.1. Beschreibung des Modells	54
3.4.2. Training des Modells	55
3.5. Testen der Hypothesen	57
4. Implementierung und Tests	58
4.1. Die Entwicklungsumgebung	59
4.1.1. Gründe für den Einsatz von Python	59
4.1.2. NetworkX	59
4.1.3. GrinPy	59
4.1.4. Matplotlib	60
4.1.5. PyTorch Geometric	60
4.1.6. Pandas	60
4.2. Applikation und Code	61
4.2.1. Eingabe des Graphen	61
4.2.2. Das GCN-Modell	62
4.2.3. Code für die Klassifikation des Graphen	63
4.2.4. Empfehlungen der topologischen Messwerte	63
5. Diskussion	65
5.1. Analyse	66
5.1.1. Beantwortung der Forschungsfragen	66
5.1.2. Datenaufbereitung	68
5.1.3. Statistische Analyse	68
5.1.4. Klassifikation	69
5.1.5. Vorschlag zum Ansatz von Ma et al.	69
5.2. Fazit	70
5.3. Ausblick	71
A. Anhang	72
A.1. Ergänzende Darstellungen zu den Resultaten	72
A.1.1. Vergleich der topologischen Indizes pro Klasse	72

Inhaltsverzeichnis

A.1.2. Einzel-Vergleich der topologischen Indizes	76
A.1.3. Einfluss aller topologischen Indizes auf die Hauptkomponenten	83
A.1.4. Erklärbare Varianzen und Usefulness-Scores aller Indizes	85
A.1.5. Weights and Biases Dashboard für die Visualisierung der Ergebnisse	88
Abbildungsverzeichnis	89
Tabellenverzeichnis	91
Definitionsverzeichnis	92
List of Listings	93
Literaturverzeichnis	94

Abkürzungen

BioChem	Biochemie
ChemInf	Chemieinformatik
CNN	Convolutional Neural Network
DB	Database
DGCNN	Deep Graph Convolutional Neural Network
FFHS	Fernfachhochschule Schweiz
GCN	Graph Convolutional Network
GNN	Graph Neural Network
PC	Principal Component
PCA	Principal Component Analysis
QSAR	Quantitative Structure-Activity Relationship
QSPR	Quantitative Structure-Property Relationship
SVM	Support Vector Machine
WWW	World Wide Web

1. Einleitung

Die Netzwerktheorie ist ein untergeordnetes Forschungsfeld der Mathematik und Data Science. Seit über 80 Jahren wird im akademischen Bereich der Graphentheorie aktiv geforscht [1, 2]. Kozyrev fasst in seiner Analyse von 1974 die bedeutendsten Meilensteine in der Vergangenheit der Netzwerktheorie zusammen. Weiter diskutiert er über deren Ursprung und verweist dabei auf Euler's, die berühmten „sieben Brücken von Königsberg“ aus dem Jahr 1726.

Im Jahr 1977, also drei Jahre nach Kozyrevs Veröffentlichung, ist ein Artikel von Linton Freeman erschienen, in welchem dieser die Zentralitätsmasse eines Netzes auf Basis der Forschung von Bavelas definiert [2, 3]. Über die Jahre sind zahlreiche Masse entwickelt worden, um die Struktur von Netzwerken zu analysieren, wie der Wiener-Index [4] oder der Randić-Index [5]. Bei den Massen geht es darum, die Struktur eines Netzwerkes zu charakterisieren oder einzigartige Muster in Netzen zu erkennen. In den darauffolgenden Jahren sind immer mehr Masse veröffentlicht worden. Mit der Zeit wird es kompliziert, für ein Netzwerk die richtigen Masse zu finden, welche eine kräftige Aussage über dessen Topologie geben.

1.1. Herleitung

Von Balaban et al. [6] werden über 1983 topologische Netzwerkindizes zusammengefasst. Viele davon eignen sich nicht für alle Netzwerkklassen und einige überschneiden sich mit anderen [7–9]. Je nach Betrachtung gibt es sogar Redundanzen von bestimmten Massen.

1.1.1. Stand der Forschung

Der Begriff „Usefulness“ ist in der Graphentheorie ausgesprochen schwer zu definieren [10, p. 144], es gibt aber verschiedene anwendungsspezifische Ansätze, wie die „Usefulness“ eines Messwertes definiert werden kann [11, p. 932] [12, p. 581]. Es wird beabsichtigt, die Usefulness von topologischen Indizes in einer quantitativen Studie von vordefinierten Netzwerkklassen zu untersuchen. Diese Arbeit wird sich auf einzelne topologische Indizes konzentrieren und ihre Korrelation und Ähnlichkeit innerhalb der Netzwerkklassen analysieren.

Die Suche nach dem optimalen Index

Bereits im Jahr 1990 forschte Randić nach den optimalen Indizes [13]. In einer Ausgabe der Croatia Chemica Acta dokumentiert er die Suche nach einem optimalen Mass, welches die Eigenschaften eines Graphen am ausführlichsten beschreibt. Die kritische Feststellung von Randić ist, dass die topologischen Indizes möglichst orthogonal sein sollten, um die Eigenschaften eines Graphen am besten wiederzugeben [13]. Aus den Ergebnissen ist ersichtlich, dass bestimmte topologische Indizes sich besser eignen, bestimmte Eigenschaften vorherzusagen, während andere weniger effektiv sind. Randić schlussfolgert, dass eine Kombination von Indizes die besten Vorhersagen ermöglicht und dass die Suche nach dem optimalen Set von Indizes ein wesentlicher Schritt bei der Vorhersage von Moleküleigenschaften ist. Es konnte bereits festgestellt werden, dass verschiedene Masse eine gewisse Redundanz aufweisen [8].

Usefulness

Usefulness-Score Der Usefulness-Score (engl. für Nützlichkeitspunktzahl) ist ein Mass dafür, wie nützlich oder hilfreich etwas ist. Er wird häufig verwendet, um Produkte, Dienstleistungen oder Informationsquellen zu bewerten und deren Wirksamkeit zu bestimmen. Es gibt zahlreiche Möglichkeiten, Nützlichkeitsbewertungen zu berechnen.

Die spezifische Methode, die eingesetzt wird, hängt dabei vom gegebenen Kontext ab, für den die Bewertung dient. Einige gängige Methoden sind Kundenbefragungen, Benutzertests und Expertenbewertungen. Beispielsweise kann ein Unternehmen Kunden befragen, um die Nützlichkeit seiner Produkte zu ermitteln, oder ein Forscher kann Benutzertests durchführen, um die Nützlichkeit einer neuen Softwareanwendung zu bewerten. Mit Usefulness-Scores kann die Entscheidungsfindung in einer Vielzahl von Kontexten quantifiziert werden. Ein Unternehmen kann etwa über Kundenfeedback und Nützlichkeitsbewertungen bestimmen, welche Produkte auf Lager gehalten werden sollen oder ob Verbesserungen an bestehenden Produkten vorzunehmen sind. In ähnlicher Weise kann eine Bibliothek Usability-Studien und Usefulness-Scores verwenden, um zu ermitteln, welche Ressourcen gekauft oder Benutzern zur Verfügung gestellt werden sollen [14].

Usefulness von topologischen Indizes In derselben Arbeit von Randić (1990) [13] werden *nützliche* Eigenschaften von topologischen Indizes angesprochen. Danail Bonchev und Oskar E. Polansky diskutieren die Anwendung von Konzepten wie Knoten- und Graphentheorie auf chemische Strukturen und zeigen, wie sie zur Beschreibung von Eigenschaften wie Reaktivität und Stabilität von Verbindungen verwendet werden können. Bonchev argumentiert, dass topologische Indizes eine effektive Möglichkeit darstellen, komplexe chemische Systeme zu beschreiben und zu verstehen sowie, dass ihre Verwendung bei der Vorhersage und Modellierung chemischer Reaktionen von grossem Nutzen sein kann [15].

Genereller Ansatz zur Usefulness Von Ma et al. [10] wurden erste Versuche unternommen, um Messwerte für verschiedene Klassen zu bewerten und deren Usefulness zu berechnen. Im dazu in der Zeitschrift Information Sciences veröffentlichten Artikel [10] wird ein topologischer Index als numerisches Mass definiert, das die topologische Struktur eines Graphen charakterisiert und dazu verwendet werden kann, um die Konnektivität oder Komplexität eines Netzwerks zu beschreiben. In ihrer Veröffentlichung legen die Wissenschaftler die Nutzung topologischer Indizes in einer Vielzahl von Bereichen dar, darunter Chemie, Biologie und soziale Netzwerke. Die Autoren diskutieren auch die Grenzen topologischer Indizes und schlagen Möglichkeiten vor, wie sie verbessert oder erweitert werden könnten, um die Komplexität realer Netzwerke besser zu erfassen. Insgesamt liegt der Nutzen topologischer Indizes in ihrer Fähigkeit, ein quantitatives Mass für die topologische Struktur eines Graphen bereitzustellen, das zur Untersuchung der Eigenschaften komplexer Systeme und zur Lösung von Problemen in vielen Bereichen verwendet werden kann.

Mathematisch wird die Usefulness eines topologischen Index als Vektor repräsentiert: Dieser besteht aus verschiedenen Meta-Indizes wie der *Structure Sensitivity*, der *Abruptness* und der *Structural Graph Measure* eines topologischen Index. Diese Meta-Indizes sind in vorhergehenden quantitativen Forschungsarbeiten definiert worden [16, 17].

$$U^W(P_6) = (W(P_6), Abr(W, P_6), SS(W, P_6), I_\lambda(P_6)) \quad (1.1)$$

Dabei ist $U^W(P_6)$ der Usefulness-Vektor des Wiener-Index für das Netzwerk P_6 und P_6 ist ein Pfadgraph mit sechs Knoten. Abr ist die Abruptness-Funktion, SS die Structure-Sensitivity-Funktion und I_λ die *Structural-Graph-Measure*.

1.1.2. Wissenschaftlicher Bezug

Mit der Thesis wird das Ziel verfolgt, einen Beitrag in der quantitativen Netzwerkanalyse zu leisten. Es geht darum, eine Anwendung zu schaffen, welcher es ermöglicht, topologische Messwerte nach ihrer Aussagekraft bzw. Usefulness zu bewerten. In der Chemieinformatik sowie der Bioinformatik werden Netzwerke zur Analyse von Molekülen und Proteinen verwendet. Dabei helfen die topologischen Indizes, die Struktur von Molekülen und Proteinen zu charakterisieren. Die Anwendung soll eine Hilfestellung für die Selektion der topologischen Indizes sein.

1.2. Zentrale Fragestellung

Die zentrale Fragestellung der Thesis ist, wie die topologischen Indizes sinnvoll miteinander verglichen werden können und wie ein nützlicher topologischer Index für die Eingabe eines Netzwerkes berechnet werden kann. Dieser Hintergrund führt zu drei verschiedene Forschungsfragen und drei

Hypothesen, mittels derer die zentrale Fragestellung beantwortet werden soll. Schliesslich wird mit dieser Arbeit die Absicht verfolgt, die Hypothesen zu überprüfen und die Forschungsfragen zu beantworten.

Die Forschungsfragen lauten:

- F1** Wie können verschiedene topologische Indizes sinnvoll miteinander verglichen werden?
- F2** Wie kann ein nützlicher topologischer Index Φ für die Eingabe eines Netzwerkes \mathcal{G} berechnet werden?
- F3** Kann durch Einsatz von Machine Learning der Vergleich von topologischen Netzwerkmesswerten optimiert werden?

Zur Beantwortung der Forschungsfragen werden die folgenden Hypothesen aufgestellt:

- H1** Topologische Indizes können miteinander verglichen werden, indem sie in einer Menge \mathcal{G} Graphen einer Netzwerkklasse \mathcal{N} gegenübergestellt und analysiert werden.
- H2** Ein nützlicher topologischer Index Φ für die Eingabe eines Netzwerkes kann gefunden werden, indem die Relevanzen der topologischen Indizes innerhalb der Netzwerkklasse \mathcal{N} definiert und berechnet werden.
- H3** Durch den Einsatz von Machine Learning kann der Prozess für das Analysieren und Untersuchen der Relevanz von Φ in \mathcal{N} verbessert und vereinfacht werden.

Im nachfolgenden Kapitel, Kapitel 2. Theorie, werden die Grundlagen für die Beantwortung der Forschungsfragen und Hypothesen erläutert.

In der Methodik und der Erarbeitung der Resultate erfolgt eine Annäherung an die Fragestellung von verschiedenen Seiten und es werden unterschiedliche Möglichkeiten diskutiert, die topologischen Indizes miteinander zu vergleichen.

2. Theorie

In der Theorie werden die notwendigen Begriffe, die topologischen Indizes sowie ein bisheriger Ansatz zur Graphenklassifikation und Usefulness beschrieben. Es werden Grundlagen der Graphentheorie sowie aktuelle Forschungsergebnisse vorgestellt.

Die Graphenklassen, welche in der Arbeit verwendet werden, sind in Kapitel [2.3](#) beschrieben.

2.1. Was ist Graphentheorie?

Die Graphentheorie ist ein Zweig der Mathematik, der mit der Untersuchung von Graphen befasst ist, wobei es sich um Strukturen handelt, die aus Knoten (eng. Nodes, auch Punkte genannt) und Kanten (eng. Edges, auch Bögen oder Linien genannt) bestehen, die die Knoten verbinden. Graphen sind ein nützliches Werkzeug zum Modellieren und Verstehen der Beziehungen zwischen verschiedenen Objekten oder Entitäten und sie haben zahlreiche Anwendungen in einer Vielzahl von Bereichen, darunter Informatik, Ingenieurwesen, Biologie und Soziologie [18, p. 1ff.].

Auf Abbildung 2.1 folgt eine Übersicht, in welcher die Arbeit verschiedenen Feldern der Lehre und Forschung angegliedert wird. Sie zeigt Schnittstellen zu diversen Bereichen und gibt den Zusammenhang der topologischen Indizes zu anderen Themen wieder.

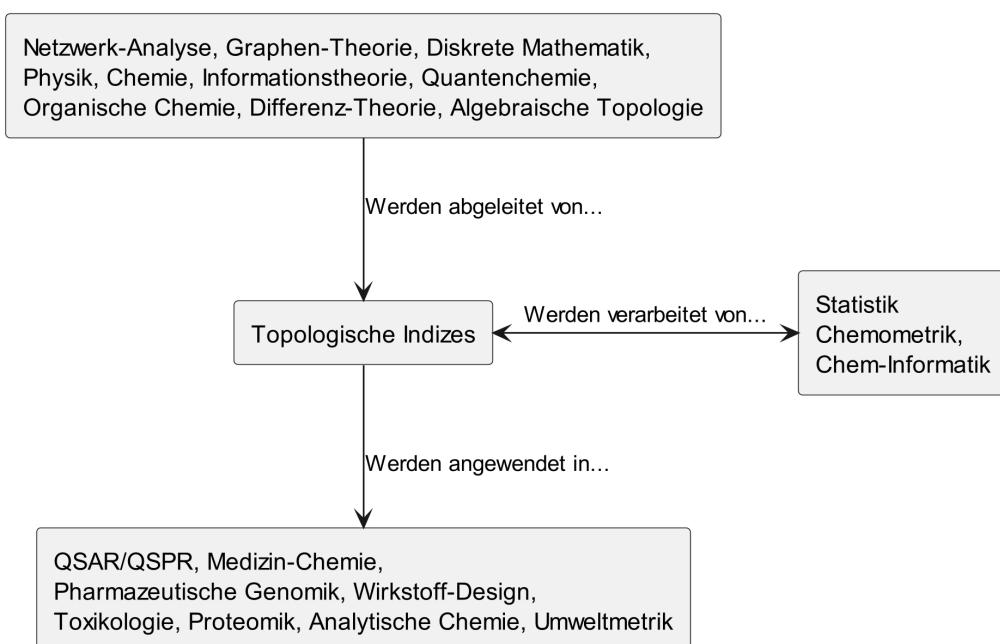


Abbildung 2.1.: Übersicht zur Anordnung des Themengebietes, übersetzt aus Todeschini und Consonni [19]

Eine der Hauptanwendungen der Graphentheorie ist die Darstellung und Analyse von Netzwerken. Beispielsweise kann ein soziales Netzwerk als Graph dargestellt werden, mit Menschen als Knoten und den Beziehungen zwischen ihnen (Freundschaften, familiäre Bindungen etc.) als Kanten. Die Analyse der Struktur des Netzwerks kann Aufschluss darüber geben, wie sich Informationen im Netzwerk verbreiten und welche Faktoren die Veränderung des Netzwerks beeinflussen können [20, p. 440ff.].

Die Graphentheorie wird auch dazu verwendet, Probleme in der Informatik und den Ingenieurwissenschaften zu lösen. Sie kann etwa genutzt werden, um den kürzesten Weg zwischen zwei Punkten in einem Verkehrsnetz zu finden [21, p. 269] oder um Aufgaben in einem Computersystem zu planen [22, p. 9]. In der Biologie kann die Graphentheorie eingesetzt werden, um die Wechselwirkungen zwischen Genen oder Proteinen in einer Zelle zu untersuchen [23, p. 3] und in der Soziologie, um die Struktur sozialer Netzwerke und deren Kommunikationsmuster zu analysieren [24, p. 1ff.].

2.2. Was ist ein Graph?

Formell wird ein Graph als $G = (V, E)$ definiert, wobei V die Menge der Knoten und E die Menge der Kanten ist. Knoten können als Punkte oder Kreise dargestellt werden und Kanten als Linien, die zwei Knoten verbinden [25, p. 45].

Ein Graph kann **ungerichtet** oder **gerichtet** sein. Bei einem ungerichteten Graphen können die Kanten in beide Richtungen durchlaufen werden. Bei einem gerichteten Graphen können die Kanten nur in eine Richtung durchlaufen werden. Eine Kante kann auch **gewichtet** oder **ungegewichtet** sein. Gewichtete Kanten haben einen numerischen Wert, der als Kosten, Länge oder ähnliches interpretiert werden kann [25, p. 46].

Auf Abbildung 2.2 folgt ein Beispiel für einen gerichteten, ungerichtet und gewichteten Graphen.

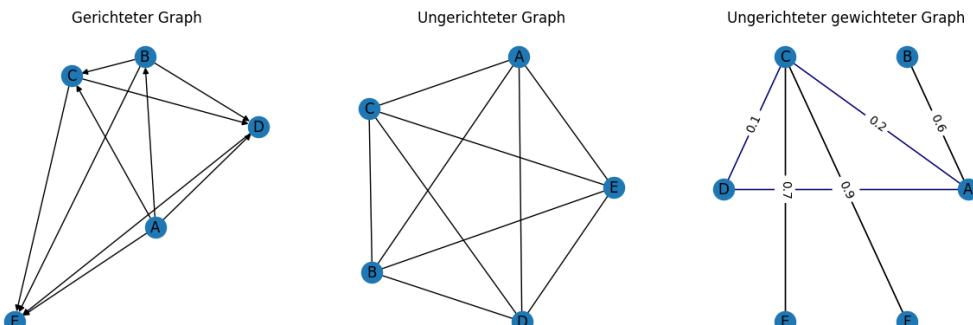


Abbildung 2.2.: Darstellung der verschiedenen Graphen

Ein Graph kann **zusammenhängend** oder **unzusammenhängend** sein. Ein zusammenhängender Graph besteht aus einer einzigen zusammenhängenden Komponente, d. h. es gibt einen Pfad von jedem Knoten zu jedem anderen Knoten. Ein unzusammenhängender Graph besteht aus mehreren Komponenten, die nicht miteinander verbunden sind.

Die **Ordnung** eines Graphen ist die Anzahl der Knoten, die in ihm enthalten sind. Die Grösse eines Graphen ist die Anzahl der Kanten, die in ihm enthalten sind. Ein Graph mit n Knoten kann höchstens $\frac{n(n-1)}{2}$ Kanten enthalten, wenn es sich um einen ungerichteten Graphen handelt [25, p. 48]. Wenn es sich um einen gerichteten Graphen handelt, kann es höchstens $n(n - 1)$ Kanten enthalten.

Ein wichtiger Begriff in der Graphentheorie ist der **Grad** eines Knotens, der die Anzahl der Kanten beschreibt, die mit diesem Knoten verbunden sind. Im Falle eines ungerichteten Graphen ist der Grad eines Knotens einfach die Anzahl seiner Nachbarn. Im Falle eines gerichteten Graphen unterscheidet man zwischen dem Eingangsgrad, der Anzahl der Kanten, die auf den Knoten zeigen, und dem Ausgangsgrad, der Anzahl der Kanten, die vom Knoten wegführen. Der Grad eines Knotens kann dazu verwendet werden, die Verbindungsstärke zwischen den Knoten zu beschreiben und somit eine Aussage über die Struktur des Graphen zu treffen [25, p. 48] [26, p. 14].

Eine Möglichkeit, die Verbindungen zwischen den Knoten eines Graphen darzustellen, ist die Verwendung von **Adjazenzmatrizen**. Eine Adjazenzmatrix ist eine quadratische Matrix, die die Verbindungen zwischen den Knoten des Graphen darstellt. Die Einträge der Matrix geben an, ob es eine Verbindung zwischen zwei Knoten gibt oder nicht. Im Falle eines ungerichteten Graphen ist die Adjazenzmatrix symmetrisch. Im Falle eines gerichteten Graphen ist die Adjazenzmatrix nicht symmetrisch. Die Verwendung von Adjazenzmatrizen kann helfen, die Struktur des Graphen zu verstehen und somit Aussagen über seine Eigenschaften und Funktionen zu treffen [25, p. 51] [26, p. 151].

Als Beispiel betrachten wir einen ungerichteten Graphen mit 4 Knoten, der wie folgt dargestellt werden kann:

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

In diesem Fall ist die Adjazenzmatrix symmetrisch, da der Graph ungerichtet ist. Die Einträge der Matrix geben an, ob es eine Verbindung zwischen zwei Knoten gibt oder nicht. Es gibt so etwa eine Verbindung zwischen Knoten 1 und Knoten 2, da der Eintrag $a_{1,2} = 1$ ist. Es gibt jedoch keine Verbindung zwischen Knoten 1 und Knoten 4, da der Eintrag $a_{1,4} = 0$ ist. Die Verwendung der Adjazenzmatrix kann helfen, die Struktur des Graphen zu visualisieren und Aussagen über seine Eigenschaften zu treffen.

Ein weiterer wichtiger Begriff in der Graphentheorie ist der **durchschnittliche Grad** des Graphen, der angibt, wie viele Kanten jeder Knoten im Durchschnitt hat [25, p. 48]. Für einen ungerichteten Graphen mit n Knoten und m Kanten wird der durchschnittliche Grad k wie folgt definiert:

$$k = \frac{2m}{n}$$

Für gerichtete Graphen unterscheidet man zwischen dem durchschnittlichen Eingangsgrad und dem durchschnittlichen Ausgangsgrad. Der durchschnittliche Grad gibt eine Aussage über die allgemeine Verbindungsstärke im Graphen und kann bei der Identifikation von Clusterstrukturen oder dem Vergleich von Graphen helfen [25, p. 48].

Ein weiteres wichtiges Konzept in der Graphentheorie ist die **Gradverteilung**. Die Gradverteilung beschreibt, wie viele Knoten in einem Graphen einen bestimmten Grad haben.

$$P(k) := \frac{\delta_k}{N}$$

Wobei $|V| := N$ und δ_k die Anzahl der Knoten mit Grad k ist [27, p. 311].

Eine häufige Gradverteilung ist die Poisson-Verteilung, die für viele zufällige Graphen gilt. Eine andere häufige Gradverteilung ist die Potenzgesetz-Verteilung, die für viele reale Netzwerke gilt, wie in sozialen Netzwerken oder in der Biologie. Die Potenzgesetz-Verteilung ist charakterisiert durch einen hohen Anteil von Knoten mit niedrigem Grad und einem kleinen Anteil von Knoten mit hohem Grad, die als Hubs bezeichnet werden. Die Gradverteilung kann helfen, die Struktur eines Graphen zu verstehen und somit Aussagen über die Funktionsweise des Graphen zu treffen [25, p. 51].

Die Zentralitätsmasse beschreiben, welche Knoten im Graphen besonders wichtig sind [27, p. 313]. Die Zentralität eines Knotens kann auf verschiedene Arten gemessen werden. Eine Möglichkeit ist die **Degree-Zentralität**, die angibt, wie viele Kanten mit einem Knoten verbunden sind. Ein Knoten mit hoher Degree-Zentralität hat viele Nachbarn und ist somit wichtiger als ein Knoten mit niedriger Degree-Zentralität. Die Degree-Zentralität $C_D(v)$ eines Knotens v in einem ungerichteten Graphen $G = (V, E)$ wird wie folgt definiert:

$$C_D(v) = k_v$$

wobei k_v die Anzahl der Nachbarn von v ist [27, p. 314].

Ein weiteres Mass für die Zentralität ist die **Betweenness-Zentralität**, die angibt, wie oft ein Knoten auf dem kürzesten Pfad zwischen anderen Knoten liegt. Ein Knoten mit hoher Betweenness-Zentralität spielt eine wichtige Rolle bei der Vermittlung von Information zwischen anderen Knoten. Die Betweenness-Zentralität $C_B(v)$ eines Knotens v in einem ungerichteten Graphen $G = (V, E)$ wird wie folgt definiert:

$$C_B(v_k) = \sum_{v_i, v_j \in V, v_i \neq v_j} \frac{\sigma_{v_i v_j}(v_k)}{\sigma_{v_i v_j}}$$

wobei $\sigma_{v_i v_j}$ die Anzahl der kürzesten Pfade von v_i nach v_j ist und $\sigma_{v_i v_j}(v_k)$ die Anzahl der kürzesten Pfade von v_i nach v_j durch v_k ist [27, p. 314].

2.3. Netzwerkklassen

Netzwerkklassen sind eine Sammlung von Netzwerken, die bestimmte Eigenschaften teilen. Diese Eigenschaften können z. B. die Zahl der Knoten, die Anzahl Kanten oder die topologischen Indizes sein. Brandstädt et al. und später Bang-Jensen et al. sowie Emmert-Streib et al. fassen populäre Graph-Modelle respektive Graphenklassen in verschiedenen Werken zusammen [27–29]. In den nachfolgenden Abschnitten folgen die bedeutendsten Netzwerkklassen nach [27]. Sie bilden die Grundlage für die Weiterarbeit in der Thesis und werden vor allem in der Datenaufbereitung und den Experimenten verwendet.

Nachfolgend werden die Eigenschaften der unterschiedlichen Netzwerkklassen „Random“, „Small World“ und „Scale-free“ miteinander verglichen. Dabei werden die Struktur, die Verteilung der existierenden Grade und die Adjazenzmatrix visualisiert und beschrieben. Zur Einführung sind reguläre, zufällige und Small-World-Graphen auf Abbildung 2.3 visualisiert.

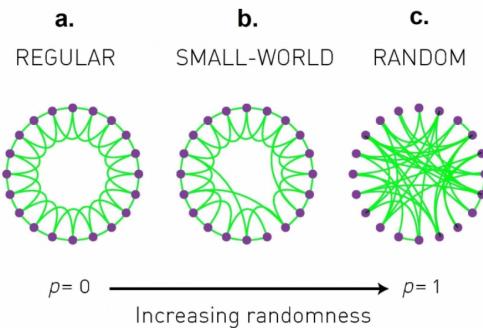


Abbildung 2.3.: Eigenschaften der unterschiedlichen Netzwerkklassen (Quelle: Barabási [25, p. 97])

2.3.1. Random

Ein Zufallsgraph (engl. random graph) ist ein Graph, der zufällig gemäss einer Wahrscheinlichkeitsverteilung über die Menge aller möglichen Graphen generiert wird. Das Studium von Zufallsgraphen ist ein Zweig der Graphentheorie und der probabilistischen Kombinatorik.

Ein gängiges Modell zum Erzeugen eines Zufallsgraphen ist das Erdős-Rényi-Modell, bei dem jede mögliche Kante unabhängig mit einer festen Wahrscheinlichkeit p aufgenommen wird [30, p. 205 ff]. Dieses Modell ist umfassend untersucht und weist besonders interessante Eigenschaften auf wie den Phasenübergang, der bei der kritischen Wahrscheinlichkeit $p = 1/n$ auftritt, wobei n die Anzahl der Scheitelpunkte im Graphen ist [31, p. 152].

Des Weiteren ist das bevorzugte Bindungsmodell (engl. preferential attachment) zu nennen, das einen Zufallsgraphen erzeugt, indem es mit einer kleinen Anzahl von Scheitelpunkten beginnt und nacheinander neue Scheitelpunkte hinzufügt, von denen jeder mit einer bestimmten Wahrscheinlichkeit proportional mit vorhandenen Scheitelpunkten verbunden ist [23, p. 2f]. Dieses Modell wird häufig verwendet, um das Wachstum von sozialen Netzwerken oder des World Wide Web (WWW) zu modellieren. Es gibt zahlreiche andere Modelle zum Generieren von Zufallsgraphen, jedes hat seine einzigartigen Eigenschaften. Einige Beispiele sind das Small-World, das Watts-Strogatz und das k -reguläre Modell [32, p. 398f.].

Das Studium von Zufallsgraphen hat zu einem tieferen Verständnis der Struktur und Eigenschaften realer Graphen sowie zur Entwicklung effizienter Algorithmen zum Generieren und Analysieren grosser Graphen geführt. Auf Abbildung 2.4 sind drei Zufallsgraphen ersichtlich; gewisse Knoten besitzen einen Grad $k = 0$, was bedeutet, dass sie isoliert sind und keine Kanten haben [25, p. 85]:

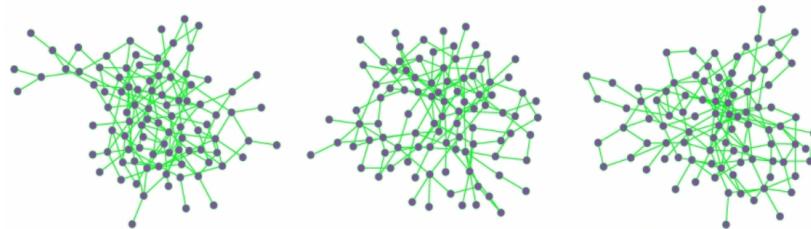


Abbildung 2.4.: Drei Random-Graphen mit $p = 0.03$ und $N = 100$ (Quelle: Barabási. [25, p. 84])

2.3.2. Small World

Ein Small-World-Graph ist eine Art komplexes Netzwerk, das sowohl eine hohe Clusterbildung als auch eine kurze durchschnittliche Pfadlänge aufweist. Das Konzept der Small-World-Graphen ist erstmals in den 1960er-Jahren von Stanley Milgram durch das „Six Degrees of Separation“-Experiment eingeführt worden. Er hat gezeigt, dass zwei beliebige Menschen auf der Erde durchschnittlich über sechs Bekanntschaften verbunden sind [33, p. 62]

Ende der 1990er-Jahre haben Duncan J. Watts und Steven H. Strogatz das Konzept der Small-World-Graphen weiterentwickelt und den Begriff „Small-World Networks“ eingeführt [20, p. 440]. Sie haben damit gezeigt, dass Small-World-Netzwerke durch eine hohe Clusterbildung gekennzeichnet sind, was bedeutet, dass Knoten in der Regel mit anderen Knoten zusammenhängen, die wiederum eng mit diesen verbunden sind, genauer gesagt durch eine kurze durchschnittliche Pfadlänge. Dies heißt, dass es unkompliziert ist, jeden Knoten im Netzwerk durch eine vergleichsweise kleine Anzahl von Schritten zu erreichen.

Watts und Strogatz (1998) definieren in ihrem Modell zum Aufbau von Small-World Networks n Knoten, wobei jeder Knoten mit m nächsten Nachbarn verbunden ist. Dies wird als reguläres Gitter bezeichnet (siehe Abbildung 2.5), wobei $n = 10$ und $m = 4$ ist [34].

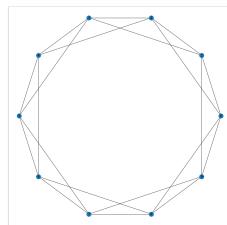


Abbildung 2.5.: Regular Lattice Graph (Quelle: Gayen [34])

Bei Betrachtung einer jeden Kante (u, v) wird mit der Wahrscheinlichkeit p zufällig ein Knoten w ausgewählt und die Kante (u, v) so verbunden, dass sie zu (u, w) wird. Für $p = 0$ behält es seine Struktur und hat einen hohen durchschnittlichen Abstand und eine hohe Clusterbildung. Für $p = 1$ wird ein Zufallsnetzwerk mit kleiner durchschnittlicher Distanz und geringer Clusterbildung gebildet 2.6, wobei $n = 10$, $m = 4$ und $p = 1$ ist. Für einen Zwischenwert von p erhält man ein ideales Small-World Network mit geringer durchschnittlicher Entfernung und hoher Clusterbildung [34].

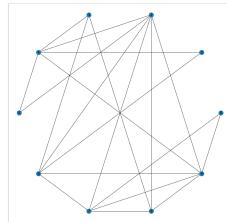


Abbildung 2.6.: Regular Lattice Graph mit $n = 10, m = 4$ und $p = 1$ (Quelle: Gayen [34])

Es hat sich herausgestellt, dass Small-World-Graphen in Systemen der realen Welt weitverbreitet sind, darunter soziale Netzwerke, Transportnetzwerke und das Internet [35, p. 4]. Sie werden auch in verschiedenen Bereichen wie Physik, Biologie und Informatik untersucht, da sie bedeutende Auswirkungen auf die Verbreitung von Informationen und die Robustheit von Netzwerken haben [36, p. 2].

2.3.3. Scale-free

In der Anfangsphase des Web, in den 1990er-Jahren, wurde davon ausgegangen, dass dieses die Eigenschaften eines zufälligen Netzwerks hat, des Netzwerktyps, der von Erdős–Rényi (1960) mathematisch charakterisiert wurde und in dem die Wahrscheinlichkeit für die Verbindung zweier Knoten als Konstante angegeben ist. In einem derartigen Netzwerk folgt die Gradverteilung der Poisson-Form [25].

Albert et al. (1999, S. 1) haben gezeigt, dass das WWW nicht diese zufällige Netzwerkstruktur hat [23]. Tatsächlich ist die Gradverteilung ein Potenzgesetz, bei dem die Wahrscheinlichkeit, dass ein Knoten den Grad k hat, proportional zu $k^{-\lambda}$ ist, wobei λ etwa 2 ist. Ein solches Netzwerk hat andere Eigenschaften, als ein zufälliges Netzwerk, da ein Potenzgesetz weniger stark abfällt als eine Poisson-Kurve. Anstatt dass praktisch alle Knoten mehr oder weniger den gleichen Grad haben, sind einige wenige Knoten äußerst stark verbunden und die überwiegende Mehrheit hat einen geringeren Grad als der Durchschnitt [25].

2.3.4. Bäume

Ein Baum (engl. Tree) ist ein gerichteter Graph, der keine Schleifen und Kreise enthält. Das bedeutet, dass jeder Knoten genau einen Vorgänger hat und dass es einen eindeutigen Pfad von jedem Knoten zum Wurzelknoten gibt [37].

Die Verwendung von Bäumen ist eine gängige Methode zur Simulation von Hierarchien und Untersuchung von Struktureigenschaften in komplexen Netzwerken. Zum Beispiel kann ein Baum Graph verwendet werden, um die Verteilung von Informationsflüssen in einem Netzwerk zu

untersuchen oder um die Effekte von Strukturänderungen auf die Effizienz von Netzwerken zu bewerten [28].

Ein weiterer Vorteil von Baum Graphen ist, dass sie einfach zu generieren sind und wenig Rechenaufwand erfordern. Daher sind sie für Anwendungen zweckmässig, bei denen schnelle und einfache Simulationen erforderlich sind, insbesondere bei der Analyse von grossen Netzwerken. Zudem können Baum Graphen auch verwendet werden, um statistische Eigenschaften komplexer Netzwerke zu schätzen, etwa die Verteilung von Knotengrössen oder die Anzahl der Verbindungen zwischen Knoten.

Insgesamt bieten Baum Graphen eine geeignete Methode zur Analyse von Hierarchien und Strukturen in komplexen Netzwerken. Sie benötigen wenig Rechenaufwand und bieten relevante Informationen über die Eigenschaften von Netzwerken. Daher sind sie ein wesentlicher Bestandteil der Netzwerktheorie und werden oft in der Praxis verwendet [27, 37].

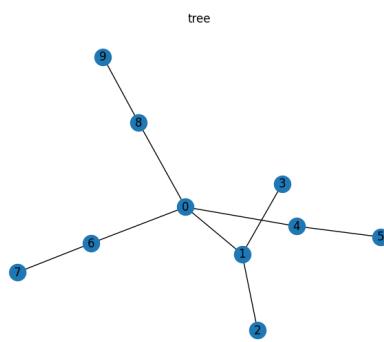


Abbildung 2.7.: Baum Graph

Sterne

Sterngraphen sind eine Klasse von ungerichteten Graphen und eine Unterklasse von Bäumen, bei denen ein Knoten (zentraler Knoten) direkt mit allen anderen Knoten (Peripherie-Knoten) verbunden ist. Ein Sterngraph mit n Knoten kann als S_n dargestellt werden. Der zentrale Knoten in einem Sterngraphen hat $n - 1$ Kanten, während jeder Peripherie-Knoten nur eine Kante hat.

Die Formel für die Anzahl der Kanten eines Sterngraphen lautet:

$$E(S_n) = n - 1 \quad (2.1)$$

Sterngraphen haben folgende Eigenschaften:

1. Sie besitzen n Knoten und $n - 1$ Kanten.

2. Sie haben keine Schleifen.
3. Ihr Durchmesser ist $\min 2, n$.
4. Sie sind zusammenhängend.

Sie sind einfach zu konstruieren und zu analysieren und werden oft verwendet, um Probleme in verschiedenen Bereichen zu lösen.

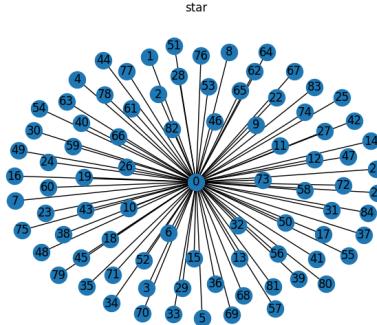


Abbildung 2.8.: Sterngraph

2.3.5. Vollständig

Vollständige Graphen sind eine bedeutsame Klasse von Graphen in der Mathematik, die eine vollständige Verbindung aller Knoten innerhalb des Graphen darstellen. Bei einem vollständigen Graphen mit n Knoten ist jeder Knoten mit jedem anderen direkt verbunden. Ein vollständiger Graph mit n Knoten kann als $G(n)$ dargestellt werden.

Die Formel für die Anzahl der Kanten eines vollständigen Graphen lautet [25, p. 53]:

$$E(G(n)) = \frac{n(n - 1)}{2} \quad (2.2)$$

Ein Complete Graph ist ein besonderer Fall des k -regulären Graphen, bei dem jeder Knoten eine gleichmässige Anzahl von Kanten hat. Er ist der maximale k -reguläre Graph für $k = n - 1$ [29].

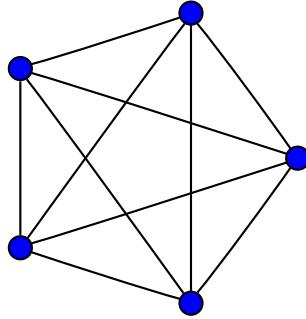


Abbildung 2.9.: K5-Graph

2.3.6. Path

Path Graphs, auch bekannt als Pfadgraphen, sind eine einfache Klasse von Graphen, die eine gerichtete Kette von Knoten darstellen. Ein Path Graph mit n Knoten kann als P_n dargestellt werden. Jeder Knoten in einem Pfadgraphen hat genau eine eingehende und eine ausgehende Kante, ausser dem ersten Knoten, der nur eine ausgehende, und dem letzten Knoten, der nur eine eingehende Kante hat.

Die Formel für die Anzahl der Kanten eines Path Graph lautet:

$$E(P_n) = n - 1 \quad (2.3)$$

Path Graphs finden häufig Anwendung in der Mathematik, insbesondere bei der Untersuchung von Problemen in der Graphtheorie und Netzwerktheorie [38]. Zum Beispiel können Path Graphs verwendet werden, um den kürzesten Pfad in einem Netzwerk zu untersuchen, und sie können auch als Bausteine für die Konstruktion von grösseren Graphen verwendet werden.

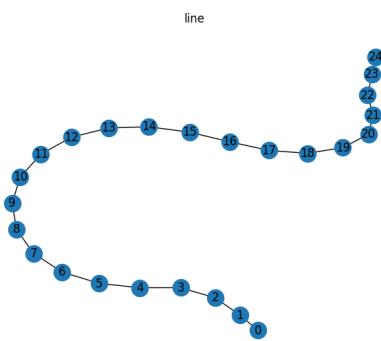


Abbildung 2.10.: Path oder Line Graph

2.4. Topologische Indizes

Topologische Indizes sind mathematische Masse, die die topologische Struktur eines Graphen charakterisieren. Sie können genutzt werden, um die Konnektivität oder Komplexität eines Netzwerks zu beschreiben, und sie haben zahlreiche Anwendungen in einer Vielzahl von Bereichen: In der Chemie dienen topologische Indizes dazu, die Eigenschaften chemischer Verbindungen wie Siede-, Schmelzpunkte und Löslichkeit vorherzusagen. Sie können auch verwendet werden, um die strukturellen und funktionellen Eigenschaften von Biomolekülen wie Proteinen und Nukleinsäuren zu untersuchen. In der Biologie kann mithilfe von topologischen Indizes, um die Protein-Ligand-Bindungsaffinität vorhergesagt werden, was für das Arzneimitteldesign bedeutsam ist. Topologische Indizes können auch zur Untersuchung der Struktur und Funktion biologischer Netzwerke eingesetzt werden, etwa Protein-Protein-Interaktionsnetzwerke und metabolische Netzwerke [39, p. 199ff]. In sozialen Netzwerken können topologische Indizes verwendet werden, um die Struktur von Netzwerken zu analysieren und zu verstehen, wie sich Informationen durch diese verbreiten.

Der topologische Index ist ein numerischer Wert oder eine Sequenz für eine bestimmte diskutierte Struktur, die tatsächlich die physikalischen, chemischen und biologischen Eigenschaften eines Graphen abbildet [40]. Einige Indizes spiegeln die Grösse des Moleküls wider, z. B. der Kohlenstoffzahlindex, andere, z. B. der Balaban-Zentralindex, charakterisieren die Form [41, p. 1].

Theorem 1 (Topologischer Index) *Ein topologischer Index ist eine numerische Invariante eines Graphen [42, p. 235].*

Topologische Indizes sind Zahlen, welche Graphen durch konstitutionelle Formeln aus mathematischen Operationen als numerische Werte repräsentieren. [43, p. 16], [43, p. 23].

2.4.1. Anwendung von topologischen Indizes

Eine der Hauptanwendungen topologischer Indizes ist die Untersuchung der Eigenschaften chemischer Verbindungen, um diese basierend auf ihren strukturellen und funktionellen Eigenschaften zu klassifizieren. Topologische Indizes werden auch zugrunde gelegt, um die Struktur und Funktion von Biomolekülen wie Proteinen und Nukleinsäuren zu untersuchen [44, p. 1015]. Topologische Indizes dienen auch dazu, wesentliche Einflussfaktoren zu identifizieren oder vorherzusagen, wie sich das Netzwerk im Laufe der Zeit verändern könnte [20, p. 400].

Topologische Indizes werden auch angewendet, um Probleme in anderen Bereichen wie Informatik, Ingenieurwesen und Wirtschaftswissenschaften zu lösen. So werden sie beispielsweise eingesetzt, um den kürzesten Weg zwischen zwei Punkten in einem Transportnetzwerk zu finden [21] und um die Ressourcenallokation in einer Lieferkette zu optimieren [45]. Insgesamt liegt der Nutzen topologischer Indizes in ihrer Fähigkeit, ein quantitatives Mass für die topologische Struktur eines Graphen bereitzustellen, das zur Untersuchung der Eigenschaften komplexer Systeme und zur Lösung von Problemen in einer Vielzahl von Bereichen verwendet werden kann.

Topologische Indizes werden u. a. von Chemikern als Werkzeug zur Beschreibung chemischer Phänomene genutzt. Die topologischen Indizes charakterisieren dabei sowohl die Grösse als auch die Form chemischer Spezies und spiegeln die Menge an Verzweigungen in Molekülen signifikant wider. Chemiker sind somit in der Lage, das chemische Verhalten einer breiten Palette chemischer Substanzen in allen drei thermodynamischen Zuständen graphenbasiert genau zu modellieren [41, p. 1].

Eine ihrer Eigenschaften, die als Einzigartigkeit oder Unterscheidungskraft bezeichnet wird, ist in der mathematischen Chemie und im strukturorientierten Arzneimitteldesign im Kontext der quantitativen Charakterisierung der Struktur von Molekülen ausführlich untersucht worden. Im Allgemeinen erhält ein Index das Attribut degeneriert, wenn er für mehr als einen Graphen denselben Wert besitzt [17, p. 1].

Weitere Wissenschaftler haben magnitudenbasierte Informationsindizes vorgeschlagen, um die Trennschärfe anderer klassischer topologischer Index für Alkanbäume und Isomere zu verbessern. Alkanbäume sind zusammenhängende und azyklische Graphen, in denen der Grad eines Knotens höchstens 4 ist. Zudem wird die Trennschärfe von informationstheoretischen Massen basierend auf Distanzen für chemische Graphen unterschieden, die einen Ring enthalten. So ist die Einzigartigkeit verschiedener informationstheoretischer und nicht informationstheoretischer Massnahmen gegeben, indem polzyklische Strukturen verwendet werden. Als Ergebnis erweisen sich der Balaban-J-Index, die Summe der lokalen Vertex-Entropien und die magnitudenbasierten Informationsindizes als einzigartig, für diese Klasse von Graphen [17, p. 1].

Neben empirischen Eigenschaften von Informationsmassen für Graphen – etwa das Bestimmen von Korrelationen zwischen den Massen – bestehen auch mathematische Probleme, z. B. der

Nachweis verschiedener Ober- und Untergrenzen von Massnahmen. Die Korrelationsfähigkeit zwischen zwei Graphmassen bezieht sich im Allgemeinen auf das Problem, ob sie strukturelle Informationen ähnlich erfassen. Außerdem wird die Klasse der Graphen-Entropiemasse, die durch Verwendung bestimmter Informationsfunktionale auf Grundlage der metrischen Eigenschaften von Graphen (z. B. der Nachbarschaft von Atomen) erhalten werden, verwendet, um Probleme in QSAR und QSPR zu lösen. Insbesondere Dehmer et al. (2012) klassifizieren die sog. Mutagenität von Molekülen unter Verwendung dieser Massnahmen und unter Nutzung überwachter Lerntechniken [17, p. 2].

Insgesamt werden in dieser Studie die Grenzen topologischer Indizes und Restriktionen bei deren Anwendung im grossen Massstab deutlich. Ein topologischer Index kann für eine bestimmte Graphenklasse eindeutig sein, aber er schlägt fehl, wenn das Mass auf eine andere Klasse angewendet wird [17, p. 9].

Topologische Indizes $TI(G)$ haben folgenden Anforderungen zu erfüllen:

Lemma 2 (Topologische Indizes und Isomorphie) Wenn $G_1 \simeq G_2$ (isomorph) sind, dann gilt $TI(G_1) = TI(G_2)$. Im Umkehrschluss gilt bei $TI(G_1) = TI(G_2)$, $G_1 \simeq G_2$ nicht, da Indizes bei nicht isomorphen Graphen auch denselben Wert ergeben können.

Dieses Lemma wird in den zwei nachstehenden Beispielen verdeutlicht.

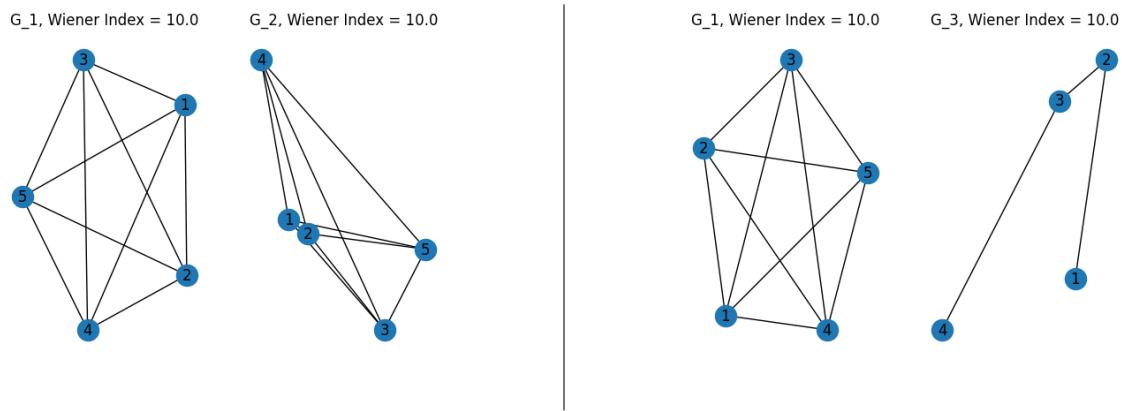


Abbildung 2.11.: Nachweis von Lemma 2. G_1 und G_2 links, sind isomorphe Graphen. G_1 und G_3 in der rechten Abbildung sind nicht isomorphe Graphen, aber haben denselben Wiener Index.

Die topologischen Indizes können nach Balaban in verschiedene Kategorien eingeteilt werden [6]. Dabei spricht er von den folgenden Klassen: gradbasiert (Adjazenzmatrix), distanzbasiert (Distanz-Matrix), zentrische Indizes und informationstheoriebasiert.

2.4.2. Gradbasiert

Sei \mathcal{G} ein molekularer Graph. Zwei Knoten von \mathcal{G} , die durch eine Kante verbunden sind, heissen „benachbart“. Sind zwei Knoten u und v benachbart, besteht die Beziehung $u \sim v$. Die Anzahl der Knoten von \mathcal{G} , die an einen gegebenen Knoten v angrenzen, ist der „Grad“ dieses Knotens und wird mit $d(v)$ bezeichnet – wenn die Eindeutigkeit gegeben ist, mit dv . Das Konzept des Grades in der Graphentheorie ist eng verwandt mit dem Konzept der Valenz in der Chemie [46, p. 351].

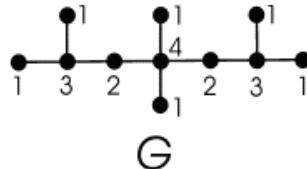


Abbildung 2.12.: Graph \mathcal{G} mit den Knotengeraden (Quelle: [46, p. 351])

Somit hat \mathcal{G} sechs Knoten vom Grad 1 (sogenannte „hängende“ Knoten, die Methylgruppen darstellen) und zwei Knoten vom Grad 2 sowie zwei vom Grad 3 und einen Knoten vom Grad 4. Aus chemischen Gründen können die molekularen Graphen von Kohlenwasserstoffen keine Knoten besitzen, deren Grad grösser als 4 ist [46, p. 351].

Randić-Index

Der Randić-Index und der Generalized Randić-Index sind Maße für die Struktur von Molekülen in der Chemie. Der Randić-Index wurde von Milan Randić entwickelt und ist ein Mass für die „Heterogenität“ oder die Unterschiedlichkeit der Atome in einem Molekül [47].

Er wird wie folgt berechnet:

$$R = \sum_{uv \in E(G)} \frac{1}{\sqrt{d_u d_v}} \quad (2.4)$$

Hierbei ist $d(i)$ die Anzahl der Nachbarn des Atoms i und die Summe läuft über alle Atome des Moleküls.

Im Jahr 1998 haben Bollobás und Erdős den generalized Randić Index definiert, bei welchem auch die Anzahl der Bindungen zwischen den Atomen berücksichtigt wird [47].

Er wird folgendermassen berechnet:

$$R_\alpha(G) = \sum_{uv \in E} d(u)d(v)^\alpha \quad (2.5)$$

Im Allgemeinen wird der Generalized Randić-Index verwendet, wenn mehr Informationen über die Struktur eines Moleküls erlangt werden sollen, insbesondere über die Anzahl und Art der Bindungen zwischen den Atomen. Der Randić Index hingegen ermöglicht nur Informationen über die Anzahl der Nachbarn eines Atoms und ist daher weniger detailliert.

Des Weiteren ist der R_α mit einem $\alpha = +1$ unter dem Namen *Second Zagreb Index* definiert [47].

Zagreb-Index

Der erste Zagreb-Index wurde 1972 von Gutman und Trinajstic definiert [48]. Er ist gleich der Summe der quadrierten Nachbarschaftsgrade aller Knoten.

$$M_1(G) = \sum_{u \in V(G)} \deg(v)^2 \quad (2.6)$$

Der zweite Zagreb-Index wird folgendermassen definiert:

$$M_2(G) = \sum_{uv \in E(G)} \deg(u)\deg(v) \quad (2.7)$$

Er ist die Summe der Produkte der Knotengrade von benachbarten Knoten [49].

Harmonischer Index

Der harmonische Index ist eine Variante des Randić-Index [50, p. 562]. Er ist als die Summe der Gewichte aller Kanten von uv von G definiert, wobei $d(u)$ den Grad eines Knotens u in G abbildet. Für einen Graphen G lautet er wie berechnet [50, p. 562]:

$$H(G) = \sum_{uv \in E(G)} \frac{2}{d(u) + d(v)} \quad (2.8)$$

Die Berechnung des harmonischen Index kann aufwendig sein, insbesondere für grosse Graphen, aber es gibt Algorithmen und Techniken, um ihn effizient zu berechnen. Der harmonische Index hat auch interessante mathematische Eigenschaften und ist ein aktives Forschungsgebiet in der Graphentheorie [51].

Atom-Bond Connectivity Index

Dieser Index beschreibt die Verbindungen zwischen den Atomen einer Verbindung und berechnet den Index auf der Basis der Anzahl der Bindungen, die jedes Atom hat. Die Verwendung von

gradbasierten Metriken zur Charakterisierung von Verbindungen ist ein bedeutender Aspekt in der chemischen Informatik, und der Atom-Bond-Connectivity (ABC) Index ist eine gängige Methode zur Bewertung der Atomkonnektivität in Verbindungen [52].

Die Formel zur Berechnung des ABC-Index lautet:

$$ABC = \sum_{uv \in E(G)} \sqrt{\frac{d_u + d_v - 2}{d_u \times d_v}} \quad (2.9)$$

Hierbei sind d_u und d_v die Grade der Knoten u und v .

2.4.3. Distanzbasiert

Wiener-Index

Der Wiener-Index ist der erste topologische Index, der von Harold Wiener, einem Chemiker, eingeführt wurde [53]. Ein fundamentaler, distanzbasierter Index ist der Wiener-Index. Er ist als Summe der Abstände zwischen allen ungeordneten Scheitelpunktpaaren des Graphen definiert und heute aufgrund des breiten Anwendungsspektrums weitgehend verbreitet. Insbesondere ist er einer der am häufigsten verwendeten topologischen Indizes in der mathematischen Chemie. Angesichts dessen korreliert es stark mit vielen physikalischen und chemischen Eigenschaften molekularer Verbindungen, deren Eigenschaften nicht nur von ihrer chemischen Formel, sondern auch von ihrer molekularen Struktur abhängen [54, p. 2]. Es werden die Summen aller Abstände (kürzeste Wege) von jedem Knoten zu jedem anderen Knoten in die Berechnung einbezogen. Der Wiener-Index ist definiert durch [4, p. 17]:

$$W = \frac{1}{2} \sum_{i,j}^N d_{ij} \quad (2.10)$$

Hosoya-Index

Der Hosoya-Index, auch molekularer topologischer Index genannt, ist ein mathematisches Mass für die topologische Komplexität eines Moleküls. Er ist definiert als die Summe der Quadrate des Grades jedes Scheitelpunkts im Molekulargraphen, wobei der Grad eines Scheitelpunkts die Anzahl der damit verbundenen Kanten ist. Der Hosoya-Index korreliert nachweislich mit verschiedenen physikalischen und chemischen Eigenschaften von Molekülen wie Siedepunkt und Löslichkeit [55, p. 397ff].

Eine der frühesten wissenschaftlichen Arbeiten zum Hosoya-Index ist 1981 von F. Hosoya und Y. Tanaka in der Zeitschrift Chemical Physics Letters veröffentlicht worden. In diesem Artikel

stellen die Autoren das Konzept des Hosoya-Index vor und demonstrieren sein Potenzial als Mass für die molekulare Komplexität. Seit seiner Einführung ist der Hosoya-Index auf dem Gebiet der chemischen und biochemischen Forschung weitverbreitet und Gegenstand zahlreicher wissenschaftlicher Arbeiten und Übersichtsartikel [55, p. 397ff].

Er wird folgendermassen berechnet [55, p. 182]:

$$Z = \sum_{k=0}^{\lfloor N/2 \rfloor} p(G, k) \quad (2.11)$$

Dabei ist $p(G, k)$ die Anzahl Möglichkeiten, in welcher k Knoten von G gewählt werden können, so dass keine zwei Knoten benachbart sind. Das in den Gauss-Klammern stehende $N/2$ ist die kleinste natürliche Zahl innerhalb $N/2$. Den grössten Hosoya-Index eines Graphen mit n Knoten erhält man von einem kompletten Graphen 2.3.5.

Szeged-Index

Der Szeged-Index ist ein Mass für den Informationsgehalt eines Graphen und wird verwendet, um die Komplexität oder Struktur des Graphen zu bewerten. Er ist von Iván Gutman 1994 eingeführt worden und generalisiert das Konzept des Wiener-Index [56].

Er lässt sich folgendermassen berechnen [56, p. 2]:

$$Sz(G) = \sum_{e \in E(G)} n_1(e|G)n_2(e|G) \quad (2.12)$$

Hier ist e eine Kante aus E im Graphen G ; e verbindet die Knoten u und v , so wird auch $e = uv$ oder $e = vu$ geschrieben. Für $e = uv \in E(G)$ sei $n_1(e|G)$ und $n_2(e|G)$ die Anzahl Knoten von G , die näher zu Knoten u als Knoten v respektive für n_2 näher bei Knoten u als Knoten v sind [56].

2.4.4. Zentrische Indizes

Die zentrischen Indizes wurden durch Balaban eingeführt und sind eine Gruppe von Indizes, die die zentrische Struktur eines Graphen beschreiben [6]. Sie verhalten sich ähnlich wie die Distanz-Indizes, da es bei der Suche nach dem zentralen Knoten auf die Distanz zwischen den Knoten ankommt. Gemäss dem Jordanschen Zentrumssatzes ist der zentrische Knoten derjenige, der die geringste Summe der Distanzen zu allen anderen Knoten hat [57, p. 32].

Die Definition des Zentrums eines Graphens G ist eine Menge von Knoten V , deren *graph eccentricity* gleich dem Graph-Radius ist [26, p. 35].

Balabans zentrischer Index B

Der zentrische Index B von Balaban wurde entwickelt, um das Zentrum von Baumgraphen zu bestimmen [58, p. 355]. B wird ähnlich wie der erste Zagreb-Index aufgrund der quadratischen Formel definiert [58, p. 355].

$$B = \sum_i \delta_i^2 \quad (2.13)$$

Dabei ist δ_i die Anzahl Knoten, welche bei jedem Schritt i entfernt werden können, ohne dass der Graph zerfällt [58, p. 355].

Estrada-Index

Der Estrada-Index EE ist ein topologischer Index, welchen Estrada im Jahr 2005 eingeführt hat [59]. Seinen Namen hat er jedoch erst 2007 von de la Peña [60] erhalten. Er misst die Partizipation von allen Knoten in allen Teilgraphen eines Graphen [59, p. 6] und wird wie folgt berechnet [59, p. 6]:

$$EE(G) = \sum_{i=1}^n e^{\lambda_i} \quad (2.14)$$

Es ist G der Graph, n die Anzahl der Knoten und λ_i sind die Eigenwerte der Adjazenzmatrix A .

2.4.5. Informationstheoriebasiert

Unter „Informationstheorie“ wird ein Zweig der Mathematik verstanden, in dem die Quantifizierung und Analyse von Informationen thematisiert wird [61, p. 82-83]. Informationstheoriebasierte Indizes stellen Masse dar, die Prinzipien der Informationstheorie verwenden, um den Informationsgehalt oder die Komplexität eines Systems quantitativ zu bewerten. Ein Beispiel für einen auf der Informationstheorie basierenden Index ist die Entropie, die ein Mass für das Ausmass der Unsicherheit oder Zufälligkeit in einem System darstellt [62, p. 767-768]. Entropie kann verwendet werden, um den Informationsgehalt einer Nachricht zu messen, indem bspw. das Ausmass der Unsicherheit quantifiziert und durch Erhalt der Nachricht reduziert wird [63, p. 3]. Newman [36, p. 2ff] diskutiert in diesem Kontext die Verwendung informationstheoretischer Indizes, einschliesslich der Entropie und gegenseitiger Informationen, um die Entwicklung von Informationen in komplexen Systemen zu untersuchen, die Zentralität von Knoten in Netzwerken zu quantifizieren sowie die Struktur und Funktion komplexer Netzwerke zu analysieren.

Balaban [6, p. 16] beschreibt zudem, dass laut einer Untersuchung die diversen informationstheoretischen Indizes eine tiefe diskriminierende Leistung aufweisen, was bedeutet, dass sie in der Lage sind, die Struktur eines Graphen zu unterscheiden, selbst wenn die Anzahl der Knoten und Kanten

überaus gross ist. Eine Ausnahme ist der Balaban-J-Index. Es ist später im Paper ein *Super-Index* eingeführt worden, welcher ein Vektor ist, der aus den verschiedenen informationstheoretischen Indizes besteht [6, p. 16]. Dieser Super-Index besteht aus dem Informations-Orbit-Index I_{ORB} , dem chromatischen Informationsindex I_{CHR} , dem Graph-Distanz-basierten Informationsindex I_D^W , dem Hosoya- und Randić-Informationsindex I_Z und I_X^E so wie dem orbitalen Informationsindex für Graphenverbindungen I_C .

Im Rahmen dieser Arbeit wird jedoch nur der chromatische Informationsindex I_{CHR} untersucht.

Chromatic Information Index

Der Chromatic Information Index (CII) I_{CHR} ist ein Mass für den Informationsgehalt eines Graphen und wird verwendet, um die Komplexität oder Struktur des Graphen zu bewerten. Er ist definiert als die Mindestanzahl von Farben, die erforderlich ist, um die Scheitelpunkte eines Diagramms richtig einzufärben, wobei Einschränkungen berücksichtigt werden, die durch die Kanten zwischen den Scheitelpunkten vorgegeben werden.

Der CII ist ein nützliches Werkzeug zum Analysieren und Vergleichen der strukturellen Eigenschaften verschiedener Arten von Graphen und findet Anwendung in Bereichen wie Datenanalyse, Netzwerkswissenschaft und Informatik. Er ist eng mit anderen Graphenfärbungsmassen wie der chromatischen Zahl, dem chromatischen Index und dem chromatischen Polynom verwandt [64, p. 93ff].

Mowshowitz definiert den I_{CHR} als *unique chromatic information content* (engl. für „einzigartiger chromatischer Informationsgehalt“) [6, p. 13], welcher die chromatische Struktur eines Graphen repräsentiert [65].

$$\hat{V} = \{V_i\}_i^h = 1(|V_i| = n_i(\hat{V})) \quad (2.15)$$

Es ist n die Anzahl der Knoten im Graphen X . Es sei \hat{V} die arbiträre chromatische Dekomposition von X , wo $h = \chi(X)$ gilt. Dann sei $I_{CHR}(X)$ der chromatische Informationsinhalt von X .

$$I_{CHR}(X) = \min V \left\{ - \sum_{i=1}^h h \frac{n_i(V)}{n} \log \frac{n_i(V)}{n} \right\} \quad (2.16)$$

2.5. Graphenklassifikation

Die Graphenklassifikation ist eine Aufgabe, über welche Graphen einer oder mehreren Klassen zugewiesen werden. Der Anfang der Graphenklassifikationstheorie liegt im Ansatz von Sobik. Er erläutert, wie die ersten Versuche zu einer Klassifikation stattgefunden haben und später, mit dem Einsatz von Machine Learning, eine effizientere Methode gefunden wurde. Zum Schluss folgt die Methode, welche auch in der Erarbeitung der Resultate verwendet wird: der Einsatz von Graph Neural Networks. Eine Übersicht über den historischen Verlauf der Graphenklassifizierung ist von Müller et al. in [66] festgehalten.

2.5.1. Klassifikation auf Basis von Isomorphismus

Bereits im Jahr 1974 hat Zelinka die Idee der Graphenklassifikation auf Basis von Isomorphismus eingeführt [67]. Nachfolgend hat Sobik diese Methode für Graphen mit verschiedenen Knoten generalisiert [68].

Graph-Isomorphie Eine Isomorphie zwischen zwei Objekten ist eine Bijektion zwischen ihren Mengen, bei der die Beziehungen erhalten bleiben. Der Isomorphismus ist eine Äquivalenzrelation auf der Menge aller Objekte [69]. Ein Graph G ist isomorph zu einem Graphen G' , wenn es eine Abbildung $\phi : V(G) \rightarrow V(G')$ gibt, welche die Kanten von G auf G' abbildet.

$G = [V, E, f, g, W_V, W_E]$ heisst (endlicher, gerichteter, interpretierbarer) **Graph** gdw. V eine endliche Menge und $E \subseteq V \times V$ ist, W_V und W_E endliche, nichtleere Mengen und $f : V \rightarrow W_V$ und $g : E \rightarrow W_E$ Funktionen von V bzw. E in W_V bzw. W_E sind [68, p. 65].

$H = [V', E', f', g', W_V, W_E]$ heisst **Teilgraph** von G gdw. $V' \subseteq V$ und $E' \subseteq E \cap (V' \times V')$ ist und f' und g' Einschränkungen von f bzw. g auf V' bzw. E' sind. Gilt $E' = E \cap (V' \times V')$, dann heisst H **induzierter Untergraph** von G . Dabei ist V die Knotenmenge und E die Kantenmenge des Graphen G , W_V und W_E sind die Mengen möglicher Knoten- bzw. Kanteninterpretationen und f und g die entsprechenden Markierungsfunktionen [68, p. 65].

Seien $G = [V, E, f, g, W_V, W_E]$ und $H = [V', E', f', g', W_V, W_E]$ Graphen. G und H heissen isomorph ($G \cong H$) gdw. eine eineindeutige Abbildung φ von $V \cup E$ auf $V' \cup E'$ existiert mit [68,

p. 65]:

$$\begin{aligned}
 \varphi(v) &\in V' & \forall v \in V \\
 \varphi(e) &\in E' & \forall e \in E \\
 \varphi((v_1, v_2)) &= \varphi(v_1), \varphi(v_2) & \forall v_1, v_2 \in V, (v_1, v_2) \in E \\
 f(v) &= f'(\varphi(v)) & \forall v \in V \\
 g(e) &= g'(\varphi(e)) & \forall e \in E
 \end{aligned}$$

Nach der Definition der Graph-Isomorphie wendet sich Sobik dem Klassifizierungsproblem zu. Er definiert die Menge der Graphen, die es erlauben, Graphen verschiedener Klassen zu unterscheiden als „klassifizierungsrelevant“ [68, p. 68].

2.5.2. Klassifizierung durch strukturelle Masse

Harary legt dar, dass strukturelle Graphmasse, also topologische Indizes, für isomorphe Graphen gleich sind [26].

Basak et al. verwenden strukturelle Graphmasse, um die strukturellen Ähnlichkeiten von verschiedenen Molekulargraphen festzustellen [70]. Somit werden erste Ansätze zur Graphenklassifikation auf Basis von topologischen Indizes gelegt.

Eine Voraussetzung für die Klassifizierung durch topologische Indizes ist die Einzigartigkeit der einzelnen Indizes untereinander. Dehmer et al. haben bewiesen, dass ein informationstheoretischer Index basierend auf der Grad-Grad-Vereinigung eine grosse Unterscheidungskraft besitzt [17].

2.5.3. Machine Learning – Support Vector Machine

Nach 2000 sind diverse Werke erschienen, in denen die Graphenklassifikation mit Machine-Learning-Methoden gelöst worden sind. Es werden Support Vector Machines (SVM) eingesetzt, um Protein-Netze für die Krebsforschung zu klassifizieren [71, 72].

Müller et al. beschreiben den Einsatz von Graph-Kernels, welche in einer ersten Form von Kashima und Inokushi [73] 2002 vorgeschlagen wurden. Dabei bilden Random-Walks den Kernel eines Graphen. Drei Jahre später hat Borgwardt den Graph-Kernel-Ansatz ausgebaut, indem er den Kernel auf Basis des shortest paths der Knoten aufgebaut hat [74].

2.5.4. Machine Learning – Neural Networks

Graph-Transformer sind ein Graph Neural Network, welches die Eigenschaften des Graphen in einen Vektor transformiert. Graph-Representation-Learning und Graph Neural Networks sind Themen, welche in den vergangenen Jahren stark an Bedeutung gewonnen haben [75–79]. Die Idee der Transformer ist 2017 in der Sprachverarbeitung entstanden [80] und im Jahr 2021 auf Graphen übertragen worden [81]. Das Forschungsfeld der Graph-Transformer ist überaus aktuell und wird stetig weiterentwickelt.

Für die Aufgabe „Graphenklassifikation“ gibt es diverse Benchmarking-Datensätze [82, 83]. Die besten Resultate in diesem Segment erreicht der Graph-Transformer U2GNN [84].

Jeder Graph G wird durch $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \{\mathbf{h}_v^{(0)}\}_{v \in \mathcal{V}})$ repräsentiert.

Dabei ist \mathcal{V} ein Set von Knoten, \mathcal{E} ein Set von Kanten und $\mathbf{h}_v^{(0)} \in \mathbb{R}^d$ ein Vektor, welcher die Eigenschaften des Knotens v repräsentiert [84].

Als Training werden ein Set M von Graphen $\{\mathcal{G}_m\}_{m=1}^M$ und ihre zugehörigen Labels $\{y_m\}_{m=1}^M \subseteq \mathcal{Y}$ verwendet.

Die Aufgabe der Graphenklassifikation ist es, für jeden Graphen \mathcal{G}_m den Embedded Graph $e_{\mathcal{G}_m}$ zu bestimmen und sein dazugehöriges Label y_m vorauszusagen [84].

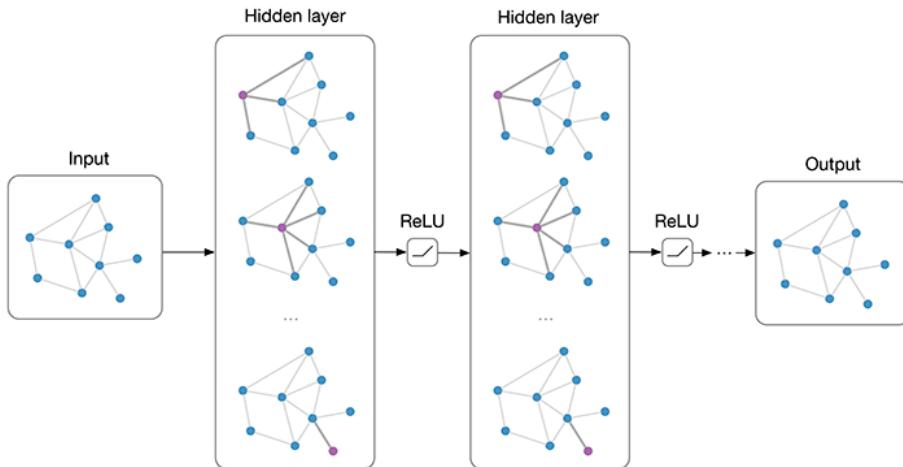


Abbildung 2.13.: Modell für die Graphenklassifizierung (Quelle: [85])

2.5.5. Unterschied von Graph-Embedding und Graph-Kernels

Graph-Kernel-basierte Ansätze zersetzen einen Graphen in eine Menge von Subgraphen, Graphlets [86] oder Weisfeiler-Lehman-Kernels genannt [87]. Diese Ansätze werden bisher verwendet, um die Ähnlichkeiten zwischen Graphen zu berechnen. Beim Zersetzen von Graphen in Graph-Kernels geht es darum, möglichst viele Informationen auf ihre zentralen Merkmale zu reduzieren. Diese Informationen werden dann in einem Vektor zusammengefasst und können mit anderen Vektoren verglichen werden.

Graph-Embedding ist ein Ansatz, welcher in den Graph Neural Networks verwendet wird. Der Ursprung geht auf William Hamilton und das Thema „Representation Learning“ von Graphen zurück [88]. Die Aggregationsfunktion ϕ ist ein Graph Neural Network, welches die Eigenschaften der Knoten in einen Vektor transformiert [89]. Mit der READOUT-Funktion ρ werden die einzelnen Vektoren dann via Sum-Pooling zusammengefasst, um Graph-Embedding zu erhalten [84].

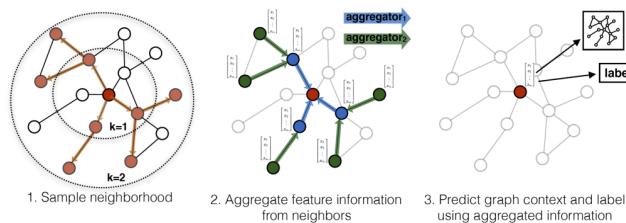


Abbildung 2.14.: Sample- und Aggregation-Ansatz visualisiert (Quelle: Hamilton [88])

3. Eigene Resultate

Die Theorie wird in Kapitel 3 verwendet, um die Resultate zu erläutern. In diesen werden weitere topologische Indizes, die im theoretischen Teil nicht vorgestellt wurden, analysiert und verglichen.

Für aufschliessende Informationen zu allen aktuellen und bedeutsamen topologischen Indizes können bestehende Literaturwerke der Graphentheorie zurate gezogen werden [17, 90–92]. Der Werdegang der Graphenklassifikation wurde aufgearbeitet und das Thema Graph-Isomorphie erläutert. Bei der Klassifikation wurden *State-of-the-Art* Methoden und deren Hintergründe vorgestellt. Als Nächstes folgen die verwendeten Daten für die Resultate sowie der Aufbau der Methodik.

Dieses Kapitel soll einen Überblick über die Resultate dieser Arbeit geben. Es werden die visualisierten Resultate sowie die Ergebnisse der Tests zusammengefasst. Die in der Einleitung 1 beschriebenen Hypothesen werden getestet und die erwarteten Resultate erarbeitet.

3.1. Erwartete Resultate

Im Idealfall wird das Resultat eine quantitative Analyse verschiedener topologischer Indizes sein und einen Beitrag zur Usefulness von topologischen Indizes leisten. Die quantitative Analyse umfasst eine explorative Datenanalyse der topologischen Indizes in verschiedenen Graph Klassen, eine statistische Analyse der Zusammenhänge zwischen den topologischen Indizes und den Graph-Klassen, so wie eine Machine Learning Analyse der Graph Klassen.

Mit der Analyse sollen die topologischen Indizes auf ihre Sinnhaftigkeit und Anwendbarkeit in verschiedenen Kontexten untersucht werden. Als Resultat soll eine Auflistung respektive Rangfolge der sinnvollsten Messwerte für die eingegebenen Graphen und topologischen Indizes ausgegeben werden. Der Algorithmus und die Formel zur Berechnung sollten möglichst einfach zugänglich und variabel sein. Der Code für die Berechnung soll Open Source sein.

Im Schlussteil der Arbeit sollen folgenden Resultate vorliegen:

- eine sinnvolle Analyse und Auflistung der topologischen Messwerte,
- eine Analyse der Berechnung des Usefulness-Scores von topologischen Indizes in verschiedenen Netzwerkklassen,
- ein Python-Modul zur Berechnung des Usefulness-Scores und
- Vorschläge von topologischen Messwerten zur Struktur eines gelieferten Netzwerks.

Um diese Resultate zu erreichen, werden folgende Schritte durchgeführt:

1. Datenaufbereitung,
2. Durchführung von Experimenten,
3. Entwicklung eines Systems, welches topologische Indizes sinnvoll vergleicht, sowie
4. Implementierung der Systeme in Python.

In der folgenden Übersicht ist dargestellt, wie die Erarbeitung der Resultate vonstatten geht:

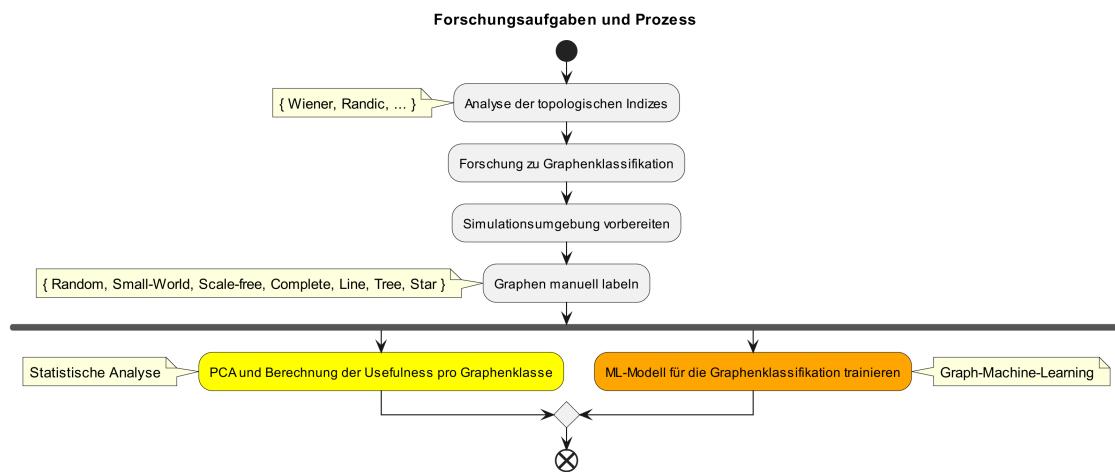


Abbildung 3.1.: Erarbeitung der Resultate

Auf Abbildung 3.1 ist dargestellt, auf welchem Weg die Resultate erarbeitet werden. Gestartet wird mit der Analyse und Dokumentation der existierenden topologischen Indizes. Der Studie zur Klassifikation von Graphen folgt das Beschaffen und Klassifizieren der Daten. Von da an ist die Arbeit in zwei parallele Arbeitsstränge geteilt. Mit einer statistischen Analyse werden die topologischen Indizes miteinander verglichen und daneben wird ein Machine-Learning-Modell entwickelt und trainiert, welches die Graphen klassifiziert. Die Resultate der beiden Arbeitsstränge werden zusammengeführt und die Hypothesen getestet.

3.2. Datenaufbereitung

3.2.1. Simulation

In der Simulationsumgebung werden diverse Netzwerke mit verschiedenen topologischen Eigenschaften erzeugt und diese gemessen. Zu Beginn werden verschiedene Netzwerke untersucht, darunter Nauty-Graphen, die aus dem Internet heruntergeladen wurden [69], sowie Netzwerke aus verschiedenen Anwendungsfällen wie biologische, soziale, chemische und Zitat-Netzwerke.

Um die Analyse zu formalisieren, liegt der Fokus der die Arbeit auf formelleren Klassen von Netzwerken. Dadurch ergeben sich zwei Vorteile für die Analyse:

- + Die Netzwerke können synthetisch erzeugt werden.
- + Die Klassen sind formal definiert und in der Literatur gut abgedeckt.

Es werden die in der Theorie zu den Netzwerkklassen 2.3 definierten Klassen verwendet. Um Graphen zu generieren, müssen deren Parameter bzw. Eigenschaften dokumentiert werden. Dadurch werden die Forschungsresultate aus dieser Arbeit reproduzierbar. Im nächsten Abschnitt werden die genauen Netzstrukturen und ihre Eigenschaften beschrieben.

3.2.2. Netze und Klassen

Es folgt das Fundament für die Daten der verwendeten Netzwerke. Diese sind in verschiedene Klassen eingeteilt.

Die Daten werden eigenständig synthetisch mit Python generiert. Es handelt sich bei allen Graphen um ungerichtete, zusammenhängende Graphen.

In dieser abgeschlossenen Liste sind die verwendeten Testdaten für die Netzwerkanalyse aufgeführt:

Tabelle 3.1.: Alle Graphen, die für die Arbeit generiert wurden und deren Anzahl Knoten.

Name	Beschreibung	Eigenschaften	Anzahl Graphen
Random	Random Erdős-Rényi-Graphen	$ V = [5..205]$, $p = 0.3$	1000
Small-World	Watts-Strogatz Small-World-Graphen	$ V = [5..205]$, $k - \text{neighbours} = \frac{ V }{2}$, $p = 0.3$	1000
Scale-free	Barabási-Albert-Graphen	$ V = [2..202]$, $m = \frac{ V }{2}$	200
Complete	k-reguläre Graphen	$ V = [2..202]$	200
Line	Pfadgraphen	$ V = [2..202]$	200
Tree	Bäume (+ Random-Trees)	$ V = [2..11] \cup [2..202]$	400
Star	Sterngraphen	$ V = [2..202]$	200

Um ein besseres Bild der generierten Netze zu ermöglichen, folgt ein visualisiertes Beispiel pro Klasse:

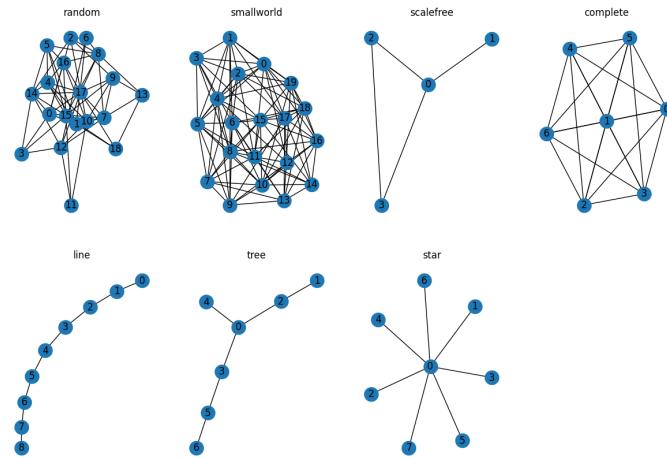


Abbildung 3.2.: Visualisierung von Graphen mit NetworkX und Matplotlib

Die Complete, Line-, Tree- und Star-Graphen unterscheiden sich schon bei der ersten Betrachtung stark. Die Random-Graphen nach Erdős-Rényi und die Small-World-Graphen können auf den ersten Blick ähnlich aussehen, differieren jedoch in der topologischen Struktur, wie bereits in der Theorie erwähnt.

3.2.3. Datenaufbereitung und Datenstruktur

Die Graphen für die Arbeit werden mithilfe von Python und NetworkX erarbeitet [93]. Das erstellte Python-Dictionary mit den jeweiligen Graphen ist nachfolgend aufgeführt:

```
1 dataset = {  
2     "random": [],  
3     "smallworld": [],  
4     "scalefree": [],  
5     "complete": [],  
6     "line": [],  
7     "tree": [],  
8     "star": []  
9 }
```

Listing 1: Datenstruktur für die Testdaten

In einem nächsten Schritt werden die erstellten Graphen ihrer Klasse zugewiesen, was auch als *Labeling* bezeichnet werden kann. Das Label der Graphen ist der Key des Dictionary. Diese Labels werden für die spätere Klassifikationsaufgabe verwendet.

3.3. Vergleich der topologischen Indizes

Die Graphen werden zunächst einer explorativen Datenanalyse unterzogen. Dabei werden erste Zusammenhänge zwischen den topologischen Indizes und den Graphenklassen untersucht.

3.3.1. Erste Erkenntnisse und Resultate

Da es sich um einen mittelgrossen Datensatz handelt und die Knotenanzahl der Graphen generell nicht über 250 liegt, dauert die Berechnung der topologischen Indizes für das ganze Datenset circa 20–30 Minuten. In der Theorie wurden die relevanten topologischen Indizes für die Analyse der Graphen beschrieben und ausgewählt. Es gibt eine Vielzahl an Messwerten, die an dieser Stelle verwendet werden könnten. Die Auswahl der topologischen Indizes ist eine subjektive Entscheidung, die auf der Erfahrung und dem Wissen des Autors basiert. Später kann die Auswahl der topologischen Indizes erweitert werden, da der Code als parametrierbares, wiederverwendbares Modul entwickelt wird. Es werden folgende topologische Indizes untersucht:

Tabelle 3.2.: Topologische Indizes unter Betrachtung

Name	Beschreibung	Implementierung
Wiener-Index	Summe aller kürzesten Pfadlängen zwischen allen Paaren von Knoten in einem Graphen	grinpy
Randic-Index	Summe der Wurzel der Gradzahlen aller Knoten in einem Graphen	grinpy
Generalized Randić-Index	Erweiterung des Randić-Index, die nicht nur die Knotengrade berücksichtigt, sondern auch andere topologische Informationen wie die Distanz zwischen Knoten, um eine umfassendere Charakterisierung der topologischen Struktur von Graphen zu ermöglichen	grinpy
Harmonic-Index	Harmonische Zentralität in Netzwerken, die als Summe der inversen harmonischen Mittelwerte aller kürzesten Pfadlängen zwischen jedem Knoten und allen anderen Knoten im Netzwerk definiert ist.	grinpy
ABC-Index	Mass für die Konnektivität von Knoten in Graphen, definiert als Summe der Produkte aus den Gradzahlen der beteiligten Knoten und der Anzahl der Kanten zwischen ihnen	grinpy
First Zagreb-Index	Summe der Quadrate der Gradzahlen aller Knoten in einem Graphen	grinpy
Second Zagreb-Index	Summe der Produkte der Gradzahlen von Paaren benachbarter Knoten	grinpy
Estrada-Index	Summe der Eigenwerte des Netzwerk-Adjazenzmatrix-Exponenten	networkx
Hosoya-Z-Index	Summe von einer Menge an Zahlen ($p(G, k)$), welche die Anzahl der Möglichkeiten angibt, wie k Bindungen aus G so ausgewählt werden, dass keine von ihnen miteinander verbunden sind	Eigenes Werk
CII	Summe der Entropien der Knotenfarben des minimalen färbenden Schemas eines Graphen	Eigenes Werk
Szeged-Index	Summe der Produkte von Graden benachbarter Knoten	Eigenes Werk

Berechnung der Indizes

Nachdem die Graphen vorbereitet und eingelesen sind, werden die Indizes berechnet. Ein Teil dieser Berechnung erfolgt mithilfe der NetworkX-Bibliothek [93], wobei die GrinPy-Bibliothek in Kombination mit NetworkX genutzt wird [94], wie zuvor erwähnt.

Um die Indizes effizient berechnen zu können, wird das Dictionary von Graphen pro Klasse durchgearbeitet, die Indizes werden berechnet und in einem Dictionary abgespeichert. Um die Variablen in einem späteren Schritt wiederverwenden zu können, wird das Dictionary in einer JSON-Datei gespeichert. Der Key der Werte ist der Index des Graphen, zum Beispiel `line_1` aus der eingelesenen Graphenliste. Die Werte des Dictionary sind die berechneten Indizes.

```

1 import grinpy as gp
2 from indices import indices
3
4 def get_topological_indices(G):
5     ''' Create a dictionary with the topological indices of a graph G. '''
6     topological_indices = {}
7     topological_indices['nodes'] = nx.number_of_nodes(G)
8     topological_indices['wiener_index'] = gp.wiener_index(G)
9     topological_indices['randic_index'] = gp.randic_index(G)
10    topological_indices['generalized_randic_index'] = gp.generalized_randic_index(G, 2)
11    topological_indices['harmonic_index'] = gp.harmonic_index(G)
12    topological_indices['atom_bond_connectivity_index'] =
13        → gp.atom_bond_connectivity_index(G)
14    topological_indices['first_zagreb_index'] = gp.first_zagreb_index(G)
15    topological_indices['second_zagreb_index'] = gp.second_zagreb_index(G)
16    topological_indices['estrada_index'] = nx.estrada_index(G)
17    topological_indices['hosoya_z_index'] = indices.hosoya_z_index(G)
18    topological_indices['chromatic_information_index'] =
19        → indices.chromatic_information_index(G)
20    topological_indices['szeged_index'] = indices.szeged_index(G)

21
22    return topological_indices

```

Listing 2: Diese Funktion berechnet die topologischen Indizes für einen Graphen.

Wie in der Tabelle der zu untersuchenden topologischen Indizes 3.1 zu sehen ist, werden einige Berechnungen aus der Bibliothek GrinPy verwendet. Andere, wie der Hosoya-Z-Index, werden in einem eigenen Indices-Modul implementiert.

Explorative Datenanalyse und erste Visualisierung

Nachdem die Indizes berechnet sind, werden die Daten visualisiert. Die Matplotlib-Bibliothek wird für die Visualisierung verwendet [95].

Um den Vergleich von zwei oder mehr Indizes sinnvoll zu gestalten, werden die Daten normalisiert und standardisiert.

Ein möglicher Vergleich von Graphen besteht darin, die Indizes innerhalb einer Klasse von Graphen einander gegenüberzustellen. Hierbei wird der Index des Graphen auf der x-Achse und der entsprechende Index-Wert auf der y-Achse dargestellt. Der Index des Graphen entspricht dem Key des zugrundeliegenden Dictionary.

Beispielsweise werden in der folgenden Analyse der Randić-Index und der generalisierte Randić-Index verglichen. Die Graphen für den Vergleich werden aus allen nicht isomorphen Graphen bis zu einer Grösse von 10 Knoten aus Nauty ausgewählt.

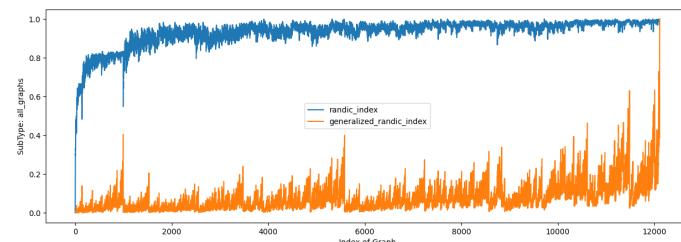


Abbildung 3.3.: Vergleich des normalisierten Randić- und des generalisierten Randić-Index von allen non isomorphic graphs (Nauty) bis 10 Knoten

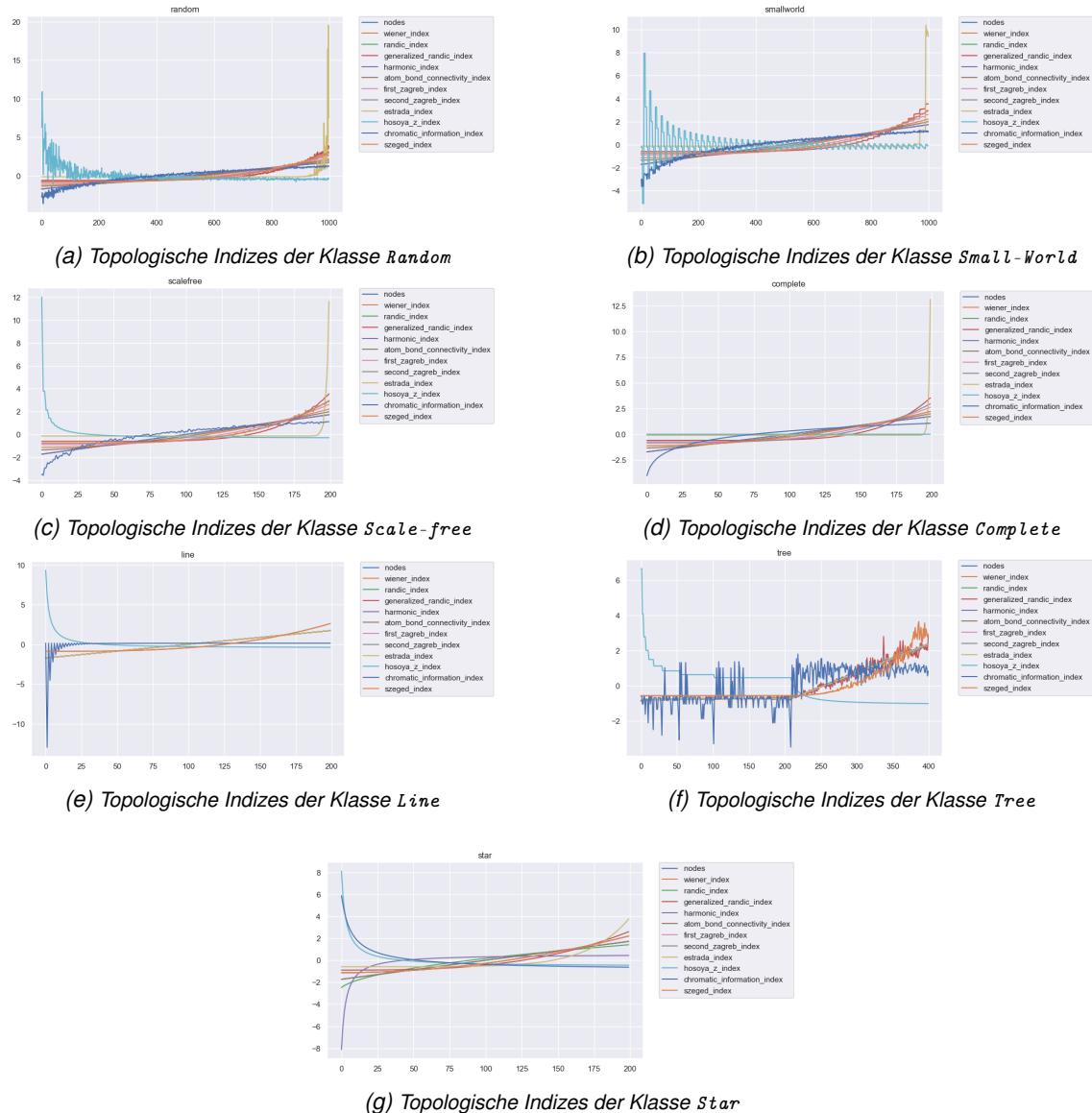
3.3.2. Vergleich der topologischen Indizes

Es werden alle topologischen Indizes, für alle Graphen berechnet, auf die Klassen aufgeteilt und normalisiert. Die Normalisierung erfolgt durch die Methode StandardScaler aus scikit-learn mit $z = (x - u)/s$. Dabei ist u der Mittelwert und s die Standardabweichung der Werte.

Danach werden die Werte der topologischen Indizes auf der y-Achse und die Anzahl Knoten der Graphen auf der x-Achse visualisiert.

3. Eigene Resultate

Vergleich der topologischen Indizes



Die Abbildungen 3.4a bis 3.4g erhalten die normalisierten topologischen Indizes der verschiedenen Graphenklassen. Die Resultate sind in Anhang A.1.1 zur besseren Lesbarkeit in grösser Form zu finden. Bereits auf den Abbildungen ist zu erkennen, dass sich die topologischen Indizes der Graphenklassen unterscheiden. Es gibt starke Differenzen zwischen den Graphenklassen Random und Small-World sowie Scale-free und Complete.

Besonders auffallend sind die Werte der Tree und Small-World Klassen auf Abbildung 3.4f und 3.4b. Hier stechen besonders der Hosoya-Z-Index und der CII heraus, sie besitzen eine besonders hohe Varianz. Sie sind besonders sensitiv auf die Struktur der Graphenklassen bei Änderungen der Knotenzahl [16].

3.3.3. Korrelation der topologischen Indizes

Um die Korrelation der topologischen Indizes zu untersuchen, werden die berechneten topologischen Indizes in einer Matrix gespeichert. Dann werden die Korrelationsmatrizen der topologischen Indizes berechnet und in einer Heatmap geplottet. Für die Korrelation wird die Spearman-Korrelation [96] verwendet.

Diese weist, wie die Pearson-Korrelation, der Korrelation von zwei Variablen einen Wert zwischen -1 und 1 zu. Dabei ist -1 eine vollständige negative Korrelation, 0 bedeutet vollständige unkorrelierte Variablen und 1 eine vollständige positive Korrelation. Die Spearman-Korrelation ist Ausreisern gegenüber weniger empfindlich als die Pearson-Korrelation [97, p. 73ff].

In Anhang A.1.2 werden zusätzlich zu den Heatmaps die einzelnen Werte in einem 2D-Scatterplot visualisiert miteinander verglichen. Um die topologischen Indizes besser zu verstehen, wurde entschieden, die topologischen Indizes für eine Klasse auch einzeln zu vergleichen. Dazu folgen als Nächstes die Zusammenhänge aller topologischen Indizes untereinander.

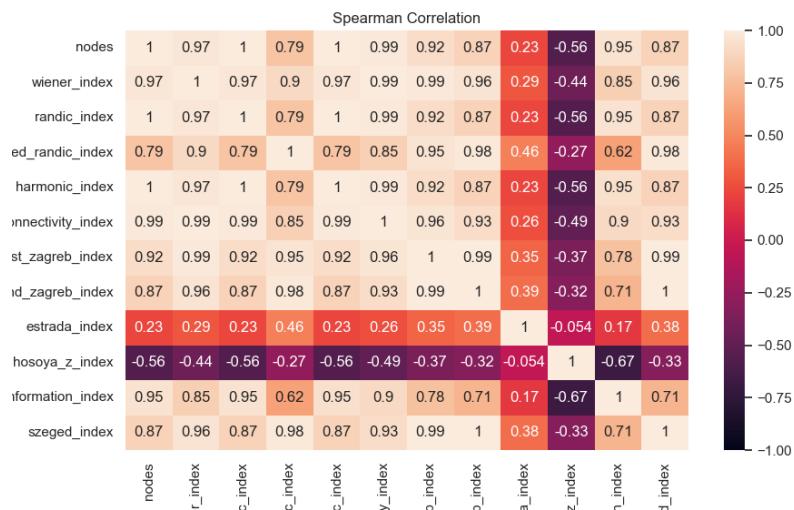


Abbildung 3.5.: Spearman-Korrelation der topologischen Indizes der Klasse Random

Anhand von Abbildung 3.5 kann festgestellt werden, dass zwei topologische Indizes einen fast unkorrelierten Zusammenhang haben. Es handelt sich dabei um die des Estrada-Index und des Hosoya-Z-Index. Ihre Werte liegen zwischen -0.67 und 0.45, wobei die meisten Werte nahe um 0 sind.

Ebenfalls auffallend ist, dass alle anderen topologischen Indizes eine positive Korrelation aufweisen. Auch wenn diese nicht überall besonders hoch ist, ist sie trotzdem generell positiv.

3. Eigene Resultate

Vergleich der topologischen Indizes

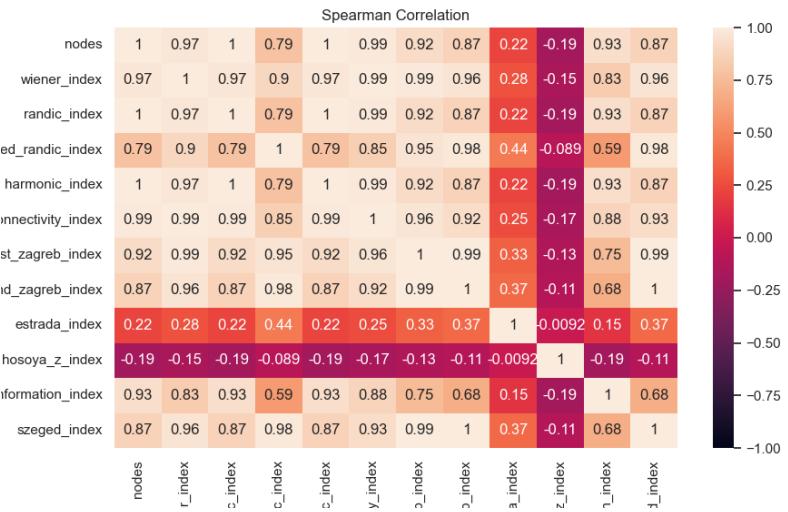


Abbildung 3.6.: Spearman-Korrelation der topologischen Indizes der Klasse Small-World

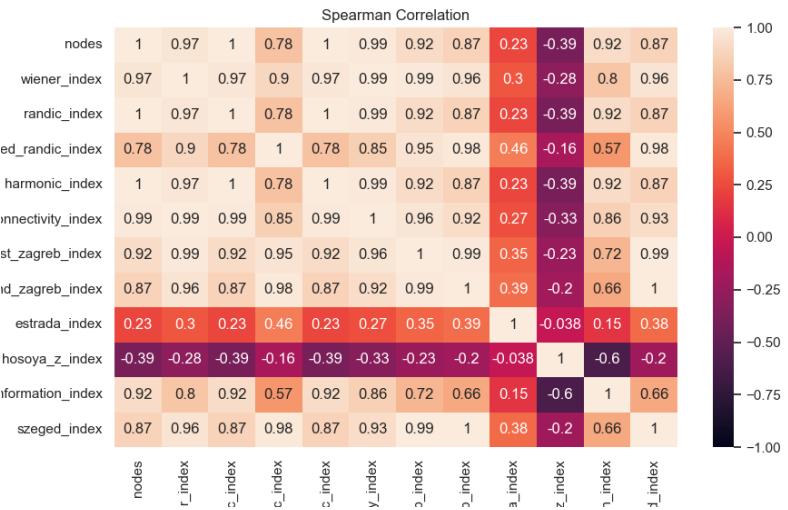


Abbildung 3.7.: Spearman-Korrelation der topologischen Indizes der Klasse Scale-free

Die Spearman-Korrelation für die Random-Graphen nach Erdős-Rényi, die Small-World-Graphen nach Watts-Strogatz und die Scale-free-Graphen nach Barabási-Albert sind auf den ersten Blick fast identisch. Ihre Heatplots liefern fast identische Ergebnisse.

Die Korrelation des Estrada- und die des Hosoya-Z-Index sind bei den Small-World- und Scale-free-Klassen noch näher bei 0 als bei den Erdős-Rényi-Graphen.

3. Eigene Resultate

Vergleich der topologischen Indizes

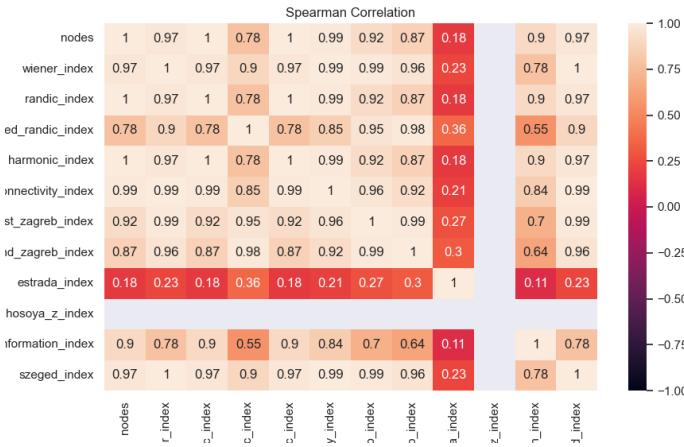


Abbildung 3.8.: Spearman-Korrelation der topologischen Indizes der Klasse Complete

Die Complete-Graphen liefern eine besondere Korrelation. Der Hosoya-Z-Index ist, wie in der Theorie 2.4.3 beschrieben, bei den Complete-Graphen immer das Maximum. Neben dem Estrada-Index, welcher ebenfalls wieder nahe bei 0 liegt, fällt der Szeged-Index auf. Dieser korreliert im Durchschnitt am stärksten mit den anderen topologischen Indizes.

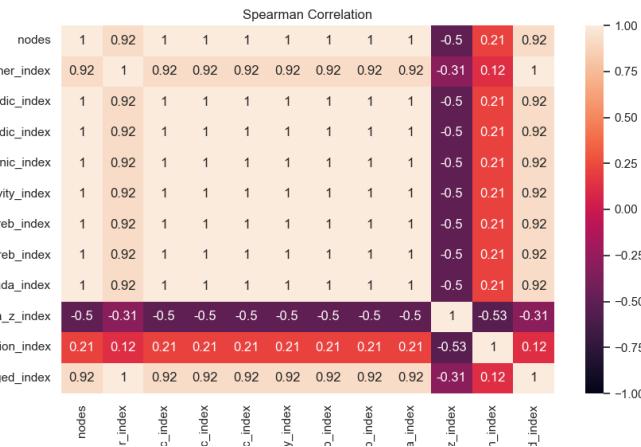


Abbildung 3.9.: Spearman-Korrelation der topologischen Indizes der Klasse Line

Die Pfadgraphen liefern eine monoton aufsteigende Korrelation. Ebenfalls in Anhang A.1.2 ist zu erkennen, dass die Werte der topologischen Indizes der Klasse Line überaus ähnlich sind. Ihr r - und p -Wert liegt bei fast allen Indizes bei 1.

3. Eigene Resultate

Vergleich der topologischen Indizes

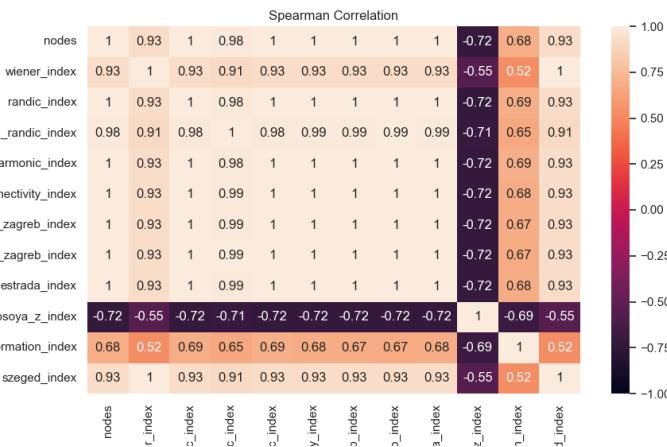


Abbildung 3.10.: Spearman-Korrelation der topologischen Indizes der Klasse Tree

Die Baumgraphen liefern eine ähnliche Korrelation wie die Pfadgraphen. Auch hier ist festzustellen, dass die Werte der topologischen Indizes der Klasse Tree überaus ähnlich sind. Neu zu erkennen ist jedoch der abfallende Trend beim CII und dem Hosoya-Z-Index. Dieser Trend ist auch in Anhang A.1.2 zu sehen. Der Wiener-Index verhält sich zum Szeged-Index in etwa gleich.

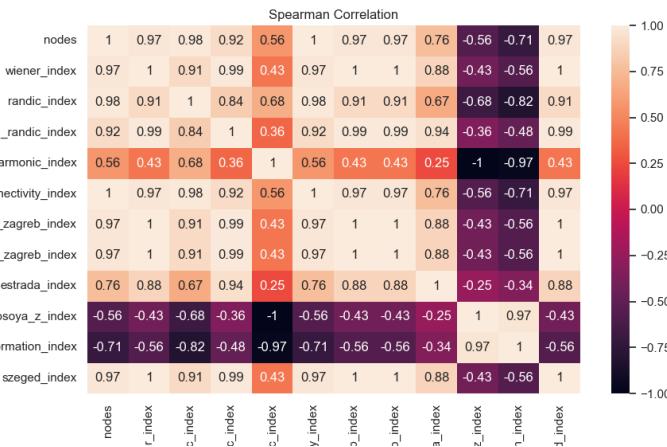


Abbildung 3.11.: Spearman-Korrelation der topologischen Indizes der Klasse Star

Die Sterngraphenklafe liefert ein neues Bild der Heatmap. Im Durchschnitt korreliert der ABC-Index am stärksten monoton aufsteigend mit den anderen topologischen Indizes. Besonders deutlich zu sehen, ist die Korrelation aller topologischen Indizes untereinander; es gibt keine Werte, welche sich durchgehend nahe bei 0 befinden.

3.3.4. Principal Component Analysis

Die Principal Component Analysis (PCA) ist eine statistische Methode, mit welcher die Dimensionalität von Daten reduziert [98] wird. Dabei werden die Daten in eine neue Basis transformiert, welche die maximale Varianz der Daten wiedergibt. Mathematisch wird dies durch die Eigenwerte und Eigenvektoren der Kovarianzmatrix der Daten beschrieben.

Beitrag der PCA zur Usefulness

Die PCA kann verwendet werden, um einen Datensatz aus interdependenten Variablen in einen neuen Satz von Variablen, den Hauptkomponenten, zu transformieren [98]. Diese Hauptkomponenten können dann analysiert werden, um Trends, Sprünge, Cluster und Ausreisser in den Daten zu beobachten. Durch die Verwendung der PCA-Methode können wir die Nützlichkeit von topologischen Indizes für verschiedene Klassen von Graphen vergleichen. Die PCA kann uns helfen, die wichtigsten Hauptkomponenten zu identifizieren, die zur Unterscheidung zwischen den verschiedenen Klassen von Graphen beitragen. Auf diese Weise können wir die topologischen Indizes identifizieren, die für die verschiedenen Klassen von Graphen am nützlichsten sind.

Durch die Anwendung der PCA können die topologischen Indizes, die die grösste Varianz innerhalb der Daten erklären, identifiziert werden und somit wichtige Informationen über die Struktur der Graphen liefern [70, p. 303].

Im Jahr 1987 haben Basak et al. 90 Indizes mittels PCA auf ihre aussagekräftigste Komponente reduziert [70]. Sie sind zum Resultat gekommen, dass die ersten zehn Komponenten 90 % der Varianz der Daten beschreiben.

Durch die Anwendung der PCA auf die topologischen Indizes können die Hauptkomponenten identifiziert werden, die die meisten Informationen enthalten und somit am meisten zur Beschreibung der Struktur des Netzwerks beitragen. Die **Usefulness** der topologischen Indizes wird in dieser Arbeit gemessen, indem die Hauptkomponenten mit den höchsten Eigenwerten identifiziert werden. Somit können die topologischen Indizes identifiziert werden, die die meisten Informationen enthalten und somit am meisten zur Beschreibung der Struktur des Netzwerks beitragen. Auf diese Weise können die wichtigsten topologischen Indizes für verschiedene Klassen von Graphen identifiziert werden.

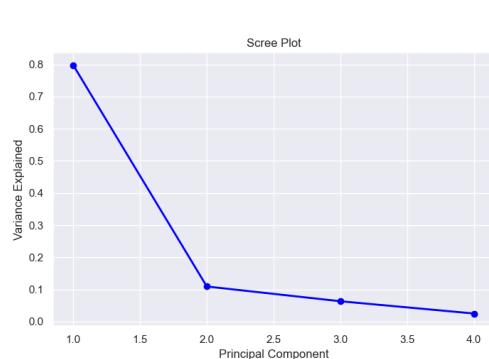
PCA in scikit-learn

Um die PCA etwas besser verstehen zu können, wird zuerst das eingegebene Datenset in einem 2D-Plot visualisiert. Die PCA-Methode in `scikit-learn` wurde per Randomized Singular Value

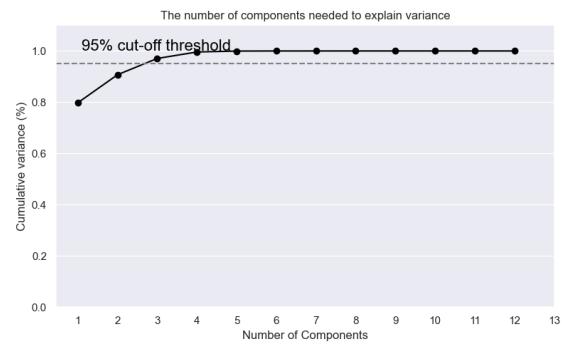
Decomposition (SVD) implementiert [99]. Diese Methode wird automatisch von der PCA-Methode in scikit-learn verwendet, wenn ein grosses Datenset ($> 500 \times 500$) eingegeben wird.

Erklärbare Varianz

Es wird untersucht, wie viele Principal Components (PCs) benötigt werden, um 95 % der Varianz der Daten zu beschreiben. Dies kann mit einem Scree-Plot dargestellt werden, der auf der x-Achse die Anzahl der PCs und auf der y-Achse die Varianz der Daten hat, welche durch die PCs beschrieben wird.

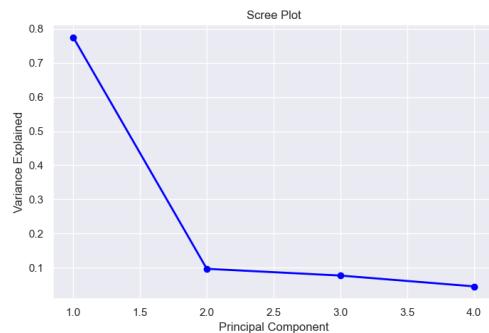


(a) Scree Plot stellt die Varianz der PC in Random dar

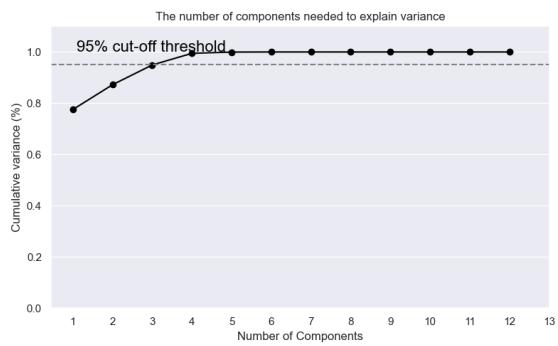


(b) Scree Plot stellt die kumulative Varianz der PC in Random dar

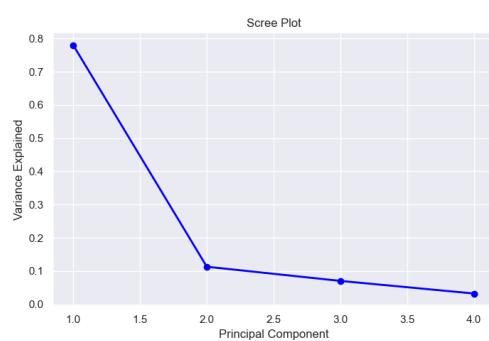
Auf allen Abbildung von 3.12a bis 3.12n ist zu erkennen, dass die Varianz der ersten Komponente überaus hoch ist. Die erklärbare Varianz von 95 % wird in allen Klassen mit maximal vier Komponenten erreicht. In den meisten Klassen reichen zwei Komponenten aus. Bei den Baum- und Sterngraphen genügt sogar eine Komponente, um 95 % der Varianz zu erklären.



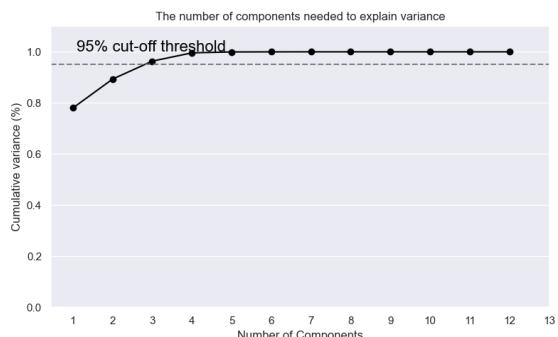
(c) Scree Plot stellt die Varianz der PC in Small-World dar



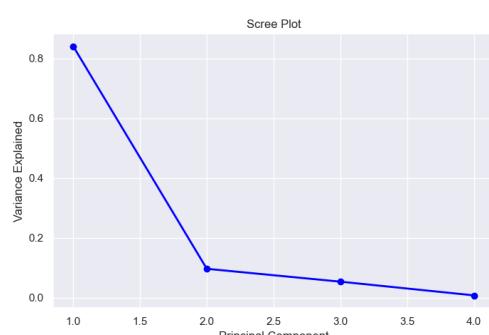
(d) Scree Plot stellt die kumulative Varianz der PC in Small-World dar



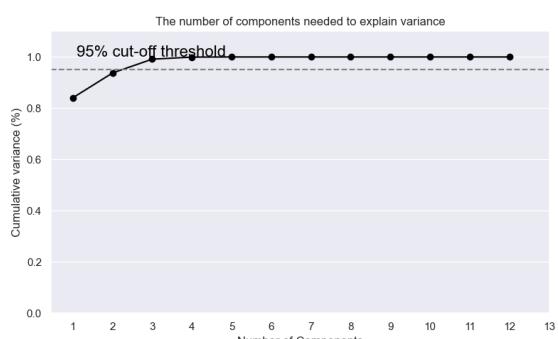
(e) Scree Plot stellt die Varianz der PC in Scale-free dar



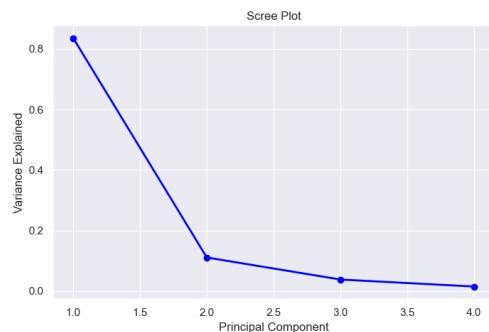
(f) Scree Plot stellt die kumulative Varianz der PC in Scale-free dar



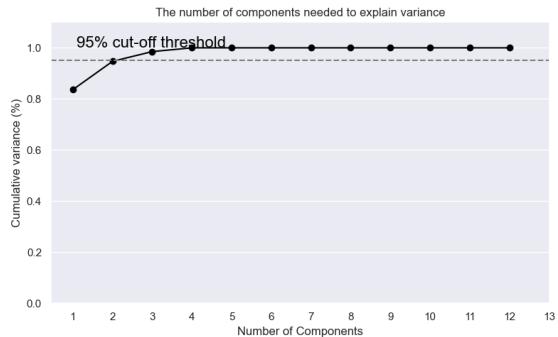
(g) Scree Plot stellt die Varianz der PC in Complete dar



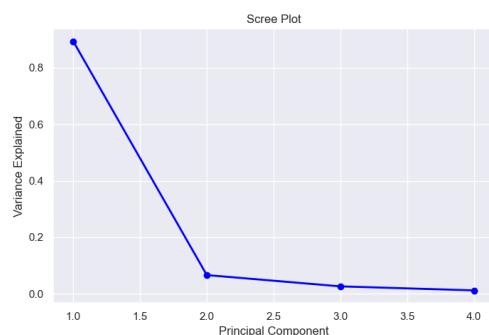
(h) Scree Plot stellt die kumulative Varianz der PC in Complete dar



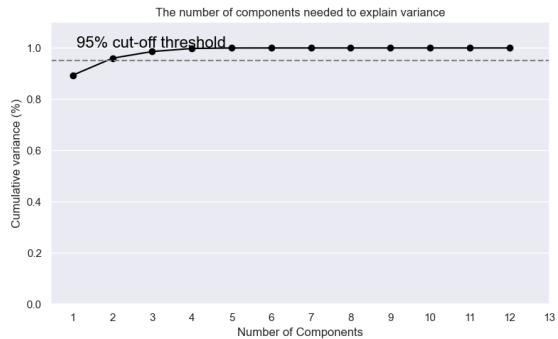
(i) Scree Plot stellt die Varianz der PC in Line dar



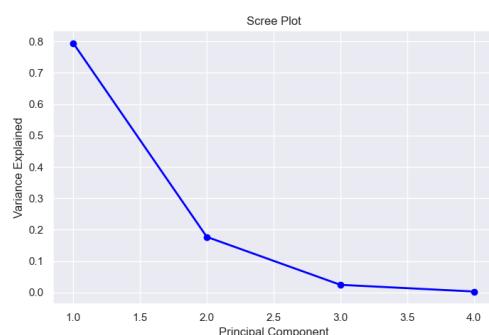
(j) Scree Plot stellt die kumulative Varianz der PC in Line dar



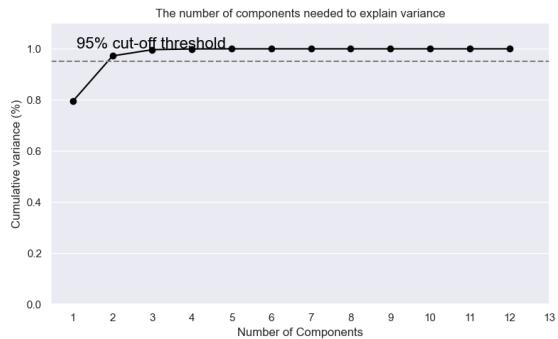
(k) Scree Plot stellt die Varianz der PC in Tree dar



(l) Scree Plot stellt die kumulative Varianz der PC in Tree dar



(m) Scree Plot stellt die Varianz der PC in star dar

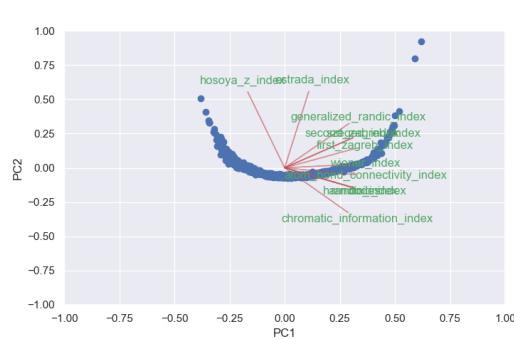


(n) Scree Plot stellt die kumulative Varianz der PC in star dar

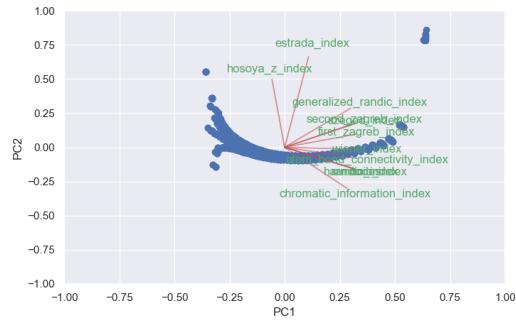
PCA-Loadings Als Nächstes folgen die Loadings der PCA-Methode. Die Eigenvektoren der PCs geben die Richtung der Komponenten an, während die Eigenwerte die Varianz der Komponenten beschreiben. Die Loadings, auch Q -Matrix genannt, sind die Gewichte der einzelnen Variablen, welche die Komponenten beschreiben [100, p. 434]. Im Wesentlichen zeigen die Loadings, wie stark jede Variable mit jeder der Hauptkomponenten korreliert. Hohe positive Loadings zeigen an, dass eine Variable stark mit einer bestimmten Hauptkomponente korreliert, während hohe negative Loadings anzeigen, dass die Variable eine starke negative Korrelation mit der Hauptkomponente aufweist. Die Loadings können verwendet werden, um zu verstehen, welche Variablen in den Daten am meisten zur Variabilität beitragen und wie die Variablen miteinander zusammenhängen [100, p. 438].

Sie werden nachfolgend in einem 2D-Plot visualisiert, wobei die x-Achse die erste Komponente und die y-Achse die zweite Komponente wiedergibt, und Folgendermassen berechnet:

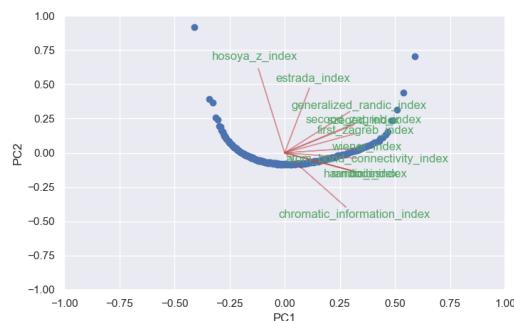
$$\text{Loadings} = \text{Eigenvektoren} \cdot \sqrt{\text{Eigenwerte}} \quad (3.1)$$



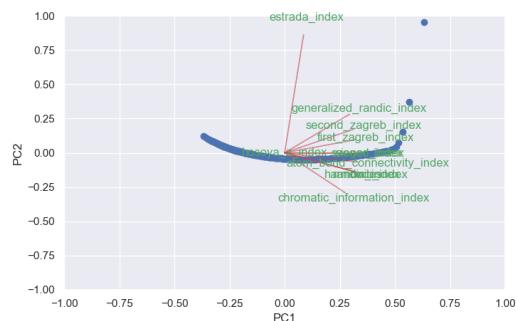
(a) Loading Plots der 2-PCA Methode der Klasse Random



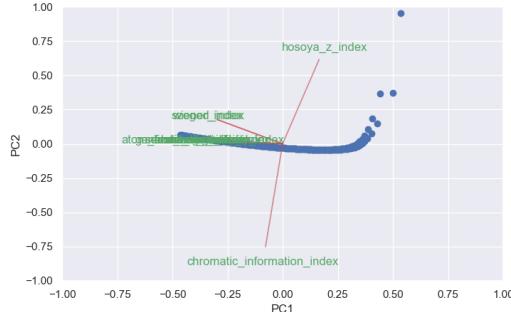
(b) Loading Plots der 2-PCA Methode der Klasse Small-World



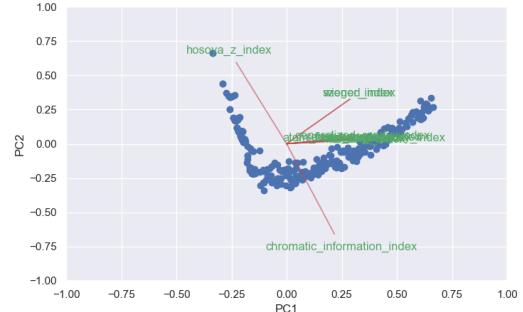
(c) Loading Plots der 2-PCA Methode der Klasse Scale-free



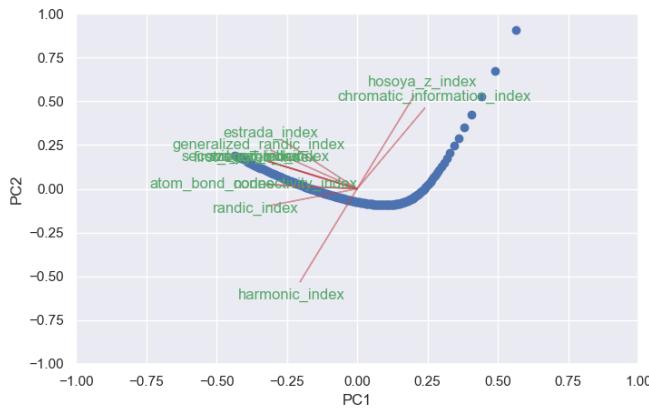
(d) Loading Plots der 2-PCA Methode der Klasse Complete



(e) Loading Plots der 2-PCA Methode der Klasse Line



(f) Loading Plots der 2-PCA Methode der Klasse Tree



(g) Loading Plots der 2-PCA Methode der Klasse Star

Die Loadings der PCs zeigen, wie die topologischen Indizes zur Komponente beitragen. Ähnlich wie bei der Spearman-Korrelation bedeuten positive Loadings, dass der Index eine positive Korrelation zur Komponente hat; negative Loadings entsprechen einer negativen Korrelation. Je höher der Wert ist, desto stärker ist die Korrelation [97]. Für uns bedeutet das, dass die Indizes mit hohen positiven Loadings die Komponente stark beeinflussen.

Die Eigenwerte λ der Komponenten speichern die Varianz der Komponenten und die Eigenvektoren \vec{v} die Richtung der Komponenten. Werden nun λ auf \vec{v} übertragen, so erhält man die PC-Loadings. Diese Loadings enthalten, da sie Varianz als auch Richtung enthalten, die Kovarianz zwischen den ursprünglichen Werten und den PCs [100, p. 438].

Einfluss der Komponenten

Zum Schluss der Korrelationsanalyse werden die einzelnen Komponenten der PCA-Methode der jeweiligen Klassen identifiziert. Mittels der den Scree-Plots lässt sich feststellen, dass bei den überwiegenden Klassen die ersten zwei Komponenten die meisten Informationen über die Daten enthalten. Deshalb werden die einzelnen Komponenten der Klasse Random bei den Hauptkomponenten zur Analyse ausgegeben.

Tabelle 3.3.: PCA-Komponenten der Erdős-Rényi-Klasse und der Einfluss der Indizes auf die Komponenten

	wiener	randic	g. randic	harmonic	abc	1st zagreb	2nd zagreb	estrada	z	cii	szeged
PC-1	0.338	0.331	0.310	0.331	0.337	0.334	0.327	0.119	-0.172	0.297	0.327
PC-2	0.007	-0.165	0.303	-0.165	-0.068	0.121	0.200	0.554	0.575	-0.342	0.198
PC-3	-0.113	-0.019	-0.075	-0.019	-0.077	-0.131	-0.123	0.784	-0.544	0.128	-0.127
PC-4	0.057	0.258	-0.379	0.258	0.166	-0.103	-0.230	0.239	0.576	0.431	-0.235

In Tabelle 3.3 ist der Einfluss der Variablen (hier topologische Indizes) auf die Komponenten der Klasse Random dargestellt. Es ist ersichtlich, dass der Wiener-Index den stärksten Einfluss auf die erste Hauptkomponente hat, der Hosoya-Z-Index auf die zweite und vierte und der Estrada-Index auf die dritte Komponente. Die abschliessende Liste aller Einflüsse der topologischen Indizes auf die Komponenten der Klassen ist in Anhang A.1.3 ersichtlich.

Die Tabelle 3.3 zeigt die Loadings der Hauptkomponenten für die Klasse Random. Anhand dieser Loadings kann die Einflüsse der verschiedenen topologischen Indizes auf die Hauptkomponenten dieser Klasse bestimmt werden. Der Index mit dem höchsten Einfluss ist schliesslich der nützlichste Index für diese Klasse. Die Herleitung dieser Aussage wird anhand der Definition 3 und der darauffolgenden Beschreibung gezeigt.

Meine Definition des *nützlichen* topologischen Index

Somit ist es nun möglich, die Hauptdefinition und Beantwortung der Forschungsfrage 1 und 2 zu finden.

Theorem 3 (nützlicher topologischer Index Φ) *Je höher der Einfluss eines topologischen Index auf die Hauptkomponenten seiner Klasse ist, desto höher ist dessen **Usefulness**.*

Für dieses Theorem muss definiert werden, was mit *Einfluss* gemeint ist. Der Einfluss eines topologischen Index auf die PCs einer Klasse lässt sich anhand der Loadings der Komponenten bestimmen. Sind die Loadings eines Index hoch positiv, so korreliert der Index positiv mit der Komponente. Die Loading-Werte der Komponenten werden in Tabelle 3.3 für die Klasse Random dargestellt.

Die Formel zur Berechnung der Usefulness eines topologischen Index auf die Hauptkomponenten seiner Klasse kann wie folgt definiert werden:

$$\Phi_i = \sum_{j=1}^m |w_{ij}| \cdot \sigma_j^2 \quad (3.2)$$

Wobei i dem Index des topologischen Indizes in dem Datenset entspricht, m der Anzahl der Hauptkomponenten und j die j -te Hauptkomponente. w_{ij} ist das Loading des topologischen Indizes i auf der j -ten Hauptkomponente. Die Werte der Loadings w_{ij} werden mit der j -ten Hauptkomponente aufsummiert und diese danach mit der erklärbaren Varianz σ^2 der j -ten Hauptkomponente multipliziert.

Die erklärbare Varianz σ^2 der Hauptkomponenten für die Klasse Random lautet:

1	----- PCA random 4 explained variance ratio -----
2	explained variance ratio
3	PC-1 0.784
4	PC-2 0.116
5	PC-3 0.069
6	PC-4 0.026

Listing 3: Erklärbare Varianz der Hauptkomponenten der Klasse Random

Nach Anwendung der Formel 3.2 werden die Daten mit der MinMax-Normalisierung $\bar{\Phi}_i = \frac{(\Phi_i - \min_x)}{(\max_x - \min_x)}$ normalisiert, wobei x das berechnete Datenset für alle Φ der Graphenklasse \mathcal{N} ist. Für den Wiener-Index und die Random-Klasse sind die Loadings in der ersten Komponente in Tabelle 3.3 gegeben. Wir sehen, dass das Loading des Wiener-Index in der ersten Komponente 0.338 beträgt.

Um die Formel 3.2 zu testen, wird der Einfluss des Wiener-Index und Randić-Index auf die Random-Graphenklasse berechnet.

$$\begin{aligned}
 \Phi_{wiener} &= |0.338| \cdot 0.784 + |0.007| \cdot 0.116 + |-0.113| \cdot 0.069 + |0.057| \cdot 0.026 \\
 &= 0.275100 \\
 \bar{\Phi}_{wiener} &= 0.747183 \\
 \Phi_{randic} &= |0.331| \cdot 0.784 + |-0.165| \cdot 0.116 + |-0.019| \cdot 0.069 + |0.258| \cdot 0.026 \\
 &= 0.286618 \\
 \bar{\Phi}_{randic} &= 0.899278
 \end{aligned} \tag{3.3}$$

Die Normalisierung der Usefulness der topologischen Indizes auf die Hauptkomponenten der Klasse Random ergibt folgende Werte:

index	usefulness score
wiener	0.747183
randic	0.899278
generalized_randic	0.985581
harmonic	0.899467
abc	0.833882
1st zagreb	0.915968
2nd zagreb	0.998606
estrada	0.000000
z	0.476098
cii	0.985445
szeged	1.000000

Listing 4: Usefulness-Score aller topologischen Indizes für die Graphenklasse Random

Das Theorem 3 besagt nun, dass der topologische Index *useful* ist, wenn sein Einfluss auf die Hauptkomponenten seiner Klasse am höchsten ist. Der Einfluss lässt sich anhand der Loadings der Komponenten bestimmen, wie in der Formel 3.2 beschrieben.

Scoring aller Klassen und Indizes

Nachdem der Einfluss der Indizes auf die Varianz der Hauptkomponenten innerhalb einer Klasse von Graphen untersucht ist, wird nun eine Reihenfolge der bedeutendsten Indizes für jede Klasse präsentiert. Dabei liegt der Fokus auf den ersten vier Hauptkomponenten, da diese in allen Fällen über 95 % der Daten erklärt werden können. Im nachfolgenden Listing sind die 3 wichtigsten Indizes für jede Klasse aufgelistet. Alle Indizes, sowie alle erklärbare Varianzen der Hauptkomponenten sind unter Anhang A.1.4 zu finden.

```

1 ----- PCA random combined usefulness score -----
2                               influence score
3 szeged                  1.000000
4 2nd zagreb              0.998606
5 generalized_randic       0.985581
6
7 ----- PCA smallworld combined usefulness score -----
8                               influence score
9 generalized_randic       1.000000
10 harmonic                0.985451
11 randic                  0.985409
12
13 ----- PCA scalefree combined usefulness score -----
14                               influence score
15 2nd zagreb              1.000000
16 szeged                  0.999622
17 cii                      0.989833
18
19 ----- PCA complete combined usefulness score -----
20                               influence score
21 generalized_randic       1.000000
22 2nd zagreb              0.990307
23 randic                  0.987024
24
25 ----- PCA line combined usefulness score -----
26                               influence score
27 szeged                  1.000000
28 wiener                  1.000000
29 generalized_randic       0.848312
30
31 ----- PCA tree combined usefulness score -----
32                               influence score
33 szeged                  1.000000
34 wiener                  0.999889
35 2nd zagreb              0.692432
36
37 ----- PCA star combined usefulness score -----
38                               influence score
39 generalized_randic       1.000000
40 estrada                 0.778188
41 szeged                  0.719018

```

Listing 5: Ausgabe der Scores der Indizes mit der höchsten Usefulness innerhalb der Graphenklassen. Bei allen Random-Graphen besitzt der Szeged-Index den höchsten Einfluss über alle Hauptkomponenten. Bei den Pfad-Graphen haben der Szeged- und Wiener-Index denselben Usefulness-Score.

Aus den Ergebnissen der Usefulness-Scores wird ersichtlich, dass bei der Graphenklaasse Star der Generalized-Randić-Index den höchsten Abstand zu den anderen Indizes hat. Hier eignet sich der Index besonders zur Verwendung und ist damit nützlicher als die anderen Masse. Der Generalized-Randić-Index hat den höchsten Einfluss auf die vier Hauptkomponenten der Klasse Star. Bei den anderen Graphenklassen sind die Scores der ersten drei Indizes sehr ähnlich.

Jedoch auch die Bäume besitzen nach den ersten beiden Indizes, dem Szeged- und Wiener-Index einen Abstand zu den anderen Indizes.

Den niedrigsten Usefulness-Score für die definierten Graphenklassen haben der CII- (Bäume, Pfad), Hosoya-Z (Small-World, Vollständig, Stern) und Estrada-Index (Scale-free, Random).

3.4. Klassifizierung der Graphen

Nach der Korrelation der Graphen folgt deren Klassifizierung. Es wird dazu ein Modell entwickelt, welches die Graphen in verschiedene Kategorien klassifiziert. Die Theorie zur Klassifizierung der Graphen wurde in Kapitel 2.5 erläutert. Nun folgen die Anwendung der hergeleiteten Theorie und deren Resultate.

3.4.1. Beschreibung des Modells

Das Modell lehnt sich an das Graph-Convolutional-network (GCN)-Modell von [85] an. Es besteht aus drei GCNConv-Schichten, einer Pooling- und einer Linear-Schicht.

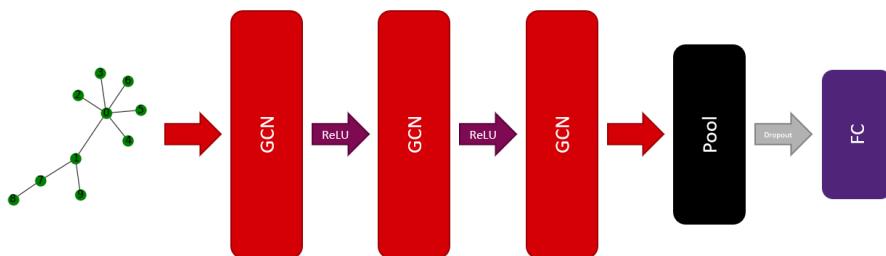


Abbildung 3.14.: Modell für die Graphenklassifizierung (Quelle: Eigene Darstellung)

Die Drei GCNConv-Schichten sind als Node-Embedding-Schichten zu verstehen.

$$X' = \hat{D}^{-1/2} \hat{A} \hat{D}^{-1/2} X \Theta \quad (3.4)$$

$\hat{A} = A + I$ ist die Adjazenzmatrix des Graphen mit eingefügten Schleifen $\hat{D}_{ii} = \sum_{j=0} \hat{A}_{ij}$ und der diagonalen Gradmatrix. Danach folgt eine *ReadOut*-Schicht. Diese existiert, um die einzelnen Node-Embeddings in ein Graph-Embedding zu überführen.

$$X_{\mathcal{G}} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} x_v^{(L)} \quad (3.5)$$

Es wird mit einer *Dropout*-Schicht [101] gearbeitet, um Overfitting zu verhindern. Zum Schluss folgt der Classifier, welcher die Graphen in die Kategorien klassifiziert.

Als Aktivierungsfunktion wird $ReLU(x) = \max(x, 0)$ verwendet. Die Optimierung der Lernrate erfolgt adaptiv mit *Adam* [102] und dem Parameter 0.01. Für die Multi-Class-Klassifizierung wird die Loss-Funktion Crossentropy verwendet.

3.4.2. Training des Modells

Zur Reproduktion und Evaluation folgt die Deklaration der genutzten Hardware und der Daten, die beim Training verwendet wurden.

Komponente	Beschreibung
Prozessor	AMD Ryzen 9 3900X (24 Cores)
Arbeitsspeicher	64 GB DDR4 3600 MHz
Grafikkarte	NVIDIA GeForce RTX 2080 Ti (CUDA 11.6)
Betriebssystem	Windows 11 Pro

Tabelle 3.4.: Technische Komponenten für das Training

Das Datenset besteht aus 3200 Graphen mit verschiedener Anzahl Knoten, siehe Tabelle 3.1. Die Knoten haben jeweils drei Features (Degree, Density und Betweenness), welche für das Node-Embedding verwendet werden. Die Multi-Class-Klassifizierung hat sieben Klassen (siehe Tabelle 3.1). Die Graphen werden in 80 % Trainings- und 20 % Testdaten aufgeteilt, was eine Trainingsmenge von 2560 Graphen und eine Testmenge von 640 Graphen ergibt. Die Test- und Trainingsdaten werden in Batches à 64 Graphen unterteilt.

Nach 62 Minuten und 32 Sekunden Training in 170 Epochen, hat das Modell eine Genauigkeit von 93 % erreicht.

```

1      Output exceeds the size limit. Open the full output data in a text editor
2      Epoch: 001, Train Acc: 0.1667, Test Acc: 0.1778
3      Epoch: 002, Train Acc: 0.2417, Test Acc: 0.2259
4      Epoch: 003, Train Acc: 0.3933, Test Acc: 0.3907
5      Epoch: 004, Train Acc: 0.5200, Test Acc: 0.5204
6      ...
7      Epoch: 167, Train Acc: 0.9816, Test Acc: 0.9750
8      Epoch: 168, Train Acc: 0.8828, Test Acc: 0.8906
9      Epoch: 169, Train Acc: 0.9238, Test Acc: 0.9313
10     Epoch: 170, Train Acc: 0.9258, Test Acc: 0.9344

```

Listing 6: Verbesserung der Genauigkeit des Modells während des Trainings

Im nächsten Bild ist die Genauigkeit über die 170 Epochen dargestellt. Es ist ersichtlich, dass die Genauigkeit vorwiegend in den ersten 25 Epochen stark zunimmt. Ab der 100. Epoche gibt es nochmals einen kleinen Anstieg von 0.5 %.

Auf Abbildung A.15 im Anhang A.15 eine Dashboard-Ansicht von Weights and Biases [103], welches die Genauigkeit und die Loss-Funktion über die Epochen darstellt. Durch das Hochladen der Trainings- und Testdaten können auch die falschen Klassifizierungen analysiert werden.

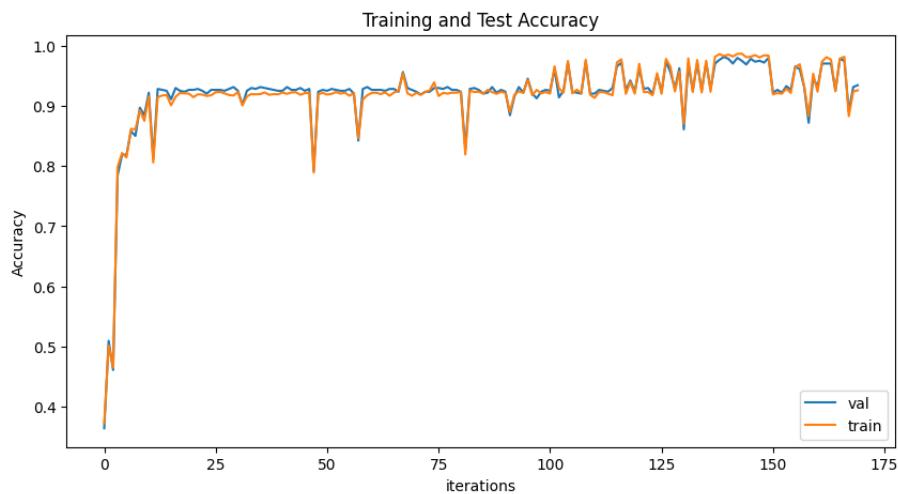


Abbildung 3.15.: Genauigkeit des Modells während des Trainings

3.5. Testen der Hypothesen

Im Rahmen der Arbeit werden drei Hypothesen formuliert. In diesem Kapitel wurden die Methoden zur Erarbeitung von Resultaten entwickelt, um diese Hypothesen zu beantworten.

Die Hypothesen lauten wie folgt:

- H1 Topologische Indizes können miteinander verglichen werden, indem sie in einer Menge \mathcal{G} Graphen einer Netzwerkklasse \mathcal{N} gegenübergestellt und analysiert werden.
- H2 Ein nützlicher topologischer Index Φ für die Eingabe eines Netzwerkes kann gefunden werden, indem die Relevanzen der topologischen Indizes innerhalb der Netzwerkklasse \mathcal{N} definiert und berechnet werden.
- H3 Durch den Einsatz von Machine Learning kann der Prozess für das Analysieren und Untersuchen der Relevanz von Φ in \mathcal{N} verbessert und vereinfacht werden.

H1: Vergleichbarkeit der topologischen Indizes

Zur Überprüfung von Hypothese H1 wurden die topologischen Indizes in einem oder mehreren Netzwerken verglichen. Hierbei wurde ein statistischer Ansatz gewählt und verschiedene Graphen mit unterschiedlichen Strukturen erzeugt. Die normalisierten Daten wurden visuell dargestellt und auf ihre Korrelation innerhalb der Klassen getestet. Es ist festzustellen, dass die topologischen Indizes miteinander verglichen werden können.

H2: Berechnung des nützlichen topologischen Index

Zur Überprüfung von Hypothese H2 wurde die PCA-Methode angewendet und die Komponenten der PCs analysiert. Aus der Analyse der Komponenten wurde ersichtlich, dass einige topologische Indizes innerhalb der PCs mehr Aussagekraft haben als andere. Dies bestätigt auch die Analyse der Vergleichbarkeit und Korrelation der topologischen Indizes. Es wurde der topologische Index mit der höchsten Relevanz für die erste und zweite PC gewählt.

H3: Machine Learning

Zur Überprüfung von Hypothese H3 wurde ein GCN trainiert. Diese hat eine Genauigkeit von über 90 % erreicht, was bedeutet, dass es 90 % der Graphen korrekt klassifiziert hat. Das GCN wurde verwendet, um die Klassen der Graphen zu bestimmen. Somit konnten die Erkenntnisse aus Hypothese H1 und H2 auf die Klassen der Graphen übertragen werden.

4. Implementierung und Tests

Ebenfalls Teil der Resultate dieser Arbeit sind die Implementierung und die Tests der entwickelten Systeme. In den erwarteten Resultaten wurde formuliert, dass die Implementierung als Bestandteil der Arbeit einfach zugänglich und ausführbar sein soll.

Nach der Erarbeitung der Resultate aus der Korrelation und der Klassifikation folgt die Entwicklung des Systems, welches die topologischen Indizes sinnvoll vergleicht.

Eine Übersicht über den Ablauf der Applikation ist in Abbildung 4.1 zu sehen.

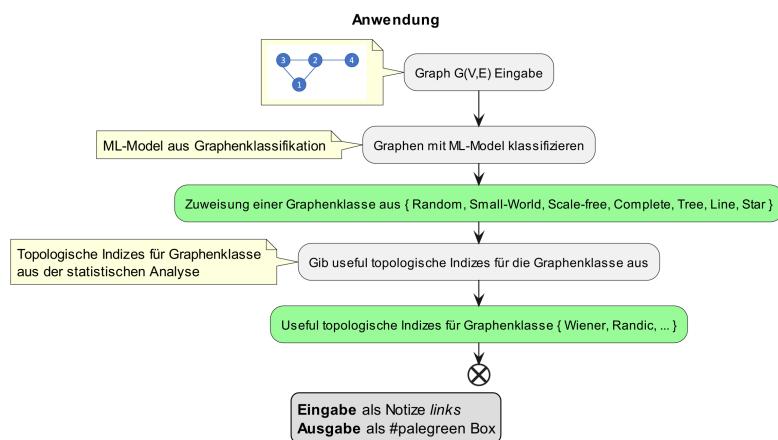


Abbildung 4.1.: Ablauf der Anwendung für die Berechnung der nützlichen topologischen Indizes für ein Netzwerk

4.1. Die Entwicklungsumgebung

4.1.1. Gründe für den Einsatz von Python

Die Programmiersprache Python wird in dieser Recherche für die Netzwerkanalyse eingesetzt, obwohl auch die Programmiersprache R eine Vielzahl an Paketen für diese Aufgabe bereitstellt.

Allerdings ist Python auf GitHub weiterverbreitet als R. Ein wesentlicher Vorteil von Python im Vergleich zu R, besteht in der Nutzung von Jupyter-Notebooks, welche die Entwicklung und Dokumentation der Netzwerkanalyse erleichtern. Durch die Kombination von Experimenten, Daten und Ergebnissen in einem Dokument können Letztere effektiv präsentiert werden. Matplotlib wird verwendet, um die Ergebnisse in Diagrammen darzustellen.

4.1.2. NetworkX

NetworkX ist die bekannteste und am meisten verwendete Bibliothek für die Netzwerkanalyse in Python [93]. Die Anwendung ist simpel und die Dokumentation weit fortgeschritten.

Ein Beispiel für das Erstellen eines einfachen Graphen mit NetworkX ist:

```
1 import networkx as nx
2
3 G = nx.Graph()
4 G.add_node(1)
5 G.add_nodes_from([2, 3])
6 G.add_edge(1, 2)
7 G.add_edges_from([(1, 2), (1, 3)])
```

Listing 7: Erstellen eines einfachen Graphen mit NetworkX

4.1.3. GrinPy

GrinPy ist eine Bibliothek für die Netzwerkanalyse in Python [94], die auf die Analyse von Graphen mit Attributen spezialisiert ist. Die Dokumentation ist noch nicht weit fortgeschritten.

GrinPy ist eine Erweiterung von NetworkX und besitzt eine Vielzahl an Funktionen, um topologische Indizes von Graphen zu berechnen. Somit können topologische Indizes von Graphen, welche durch NetworkX erstellt wurden, direkt berechnet werden. Die Graphen müssen nicht erst in ein anderes Format konvertiert werden.

4.1.4. Matplotlib

Matplotlib ist eine Bibliothek für die Visualisierung von Daten in Python [95] und wird eingesetzt, um die Ergebnisse der Netzwerkanalyse in Diagrammen darzustellen. Dies ist für die explorative Datenanalyse und das Verständnis der Ergebnisse nützlich.

Auch kann die Bibliothek verwendet werden, um die Ergebnisse in einem Dokument zu visualisieren. Die Integration in die Skripts und Jupyter-Notebooks ist unkompliziert. Ergebnisse können direkt ausgegeben und betrachtet werden. Zudem können die Bilder auch in verschiedenen Formaten zur weiteren Bearbeitung abgespeichert werden.

4.1.5. PyTorch Geometric

PyTorch Geometric [104] ist eine Graph-Machine-Learning-Bibliothek, welche eine Erweiterung von PyTorch [105] entwickelt darstellt. Sie ist entwickelt worden, um schnell und einfach Graph Neural Networks zu implementieren. Sie kann für verschiedene Anwendungen eingesetzt werden, unter anderem für **Node-Level-Predictions**, **Edge-Level-Predictions** oder in diesem Fall **Graph-Level-Predictions**.

Die Bibliothek ist optimal dokumentiert und bietet eine Vielzahl an Beispielen, welche die Implementierung von Graph Neural Networks erleichtern.

4.1.6. Pandas

Pandas ist eine Bibliothek für die Datenanalyse in Python [106]. Im Kontext der Arbeit eignet sich Pandas besonders, um die berechneten topologischen Indizes in einem Data-Frame zu speichern. Wie bereits in Code-Listing 2 beschrieben, wird als Index der Tabelle jeweils der Name (Index) des Graphen verwendet. Die Spalten des Data-Frames sind die verschiedenen Werte der topologischen Indizes.

Diverse statistische Methoden sind direkt in Pandas implementiert. Unter anderem können die Mittelwerte, Standardabweichungen und Mediane direkt berechnet werden. Aber auch die Korrelation zwischen den jeweiligen topologischen Indizes kann vereinfacht angezeigt und ausgegeben werden.

4.2. Applikation und Code

Es folgt der Code zur Entwicklung eines Python-Moduls, welches die Berechnung der nützlichen topologischen Indizes in verschiedenen Graphenklassen ermöglicht. Das Modul kann auf GitLab unter <https://git.ffhs.ch/luca.hostettler/bt-hostettler> gefunden werden.

Es ist in zwei Schritte unterteilt:

1. **Eingabe eines Graphen,**
2. **Empfehlungen der topologischen Messwerte**

4.2.1. Eingabe des Graphen

Das Programm akzeptiert als Eingabe einen NetworkX-Graphen. Aus der Datenanalyse wurde die Usefulness einzelner topologischer Indizes für verschiedene Netzwerkarten ermittelt. Dabei wurden die Netzwerke in folgende Kategorien eingeteilt: **Random**, **Small-World**, **Scale-free**, **Complete**, **Line**, **Tree** und **Star**. Die Klassifizierung des Graphen erfolgt mit dem GCN-Modell aus Abschnitt [3.4](#), welches bereits beschrieben und trainiert wurde.

```
1 # create an instance of the app
2 app = App()
3 # create a sample graph using networkx
4 graph = nx.path_graph(150)
5 # test graph classification
6 g_class = app.classify(graph)
7
8 print(f"Found graph class {app.class_keys[g_class]}")
9 # get the topological indices
10 topological_indices = app.get_topological_indices(graph)
11 # print the topological indices
12 print(topological_indices)
```

Listing 8: Code zum Einlesen des Graphen

Wie im vorhergehenden Code-Listing ersichtlich, ist der benötigte Code für die Applikation überaus kurz und einfach. Dabei ist App die Klasse, welche die Applikation implementiert. Die Methode classify ist für die Klassifikation des Graphen zuständig. Die Methode get_topological_indices ist für das Laden und Anzeigen der für den Graphen nützlichen topologischen Indizes verantwortlich.

4.2.2. Das GCN-Modell

Das Rezept für das Modell besteht aus in drei Schritten:

1. jeden Knoten mit Node-Embedding verarbeiten,
2. alle Node-Embeddings in ein gemeinsames Graph-Embedding zusammenfassen (ReadOut-Layer) und
3. die Klassifizierung auf Graph-Embedding trainieren.

```

1  class GCN(torch.nn.Module):
2      def __init__(self, hidden_channels=64, num_node_features=3, num_classes=7):
3          super(GCN, self).__init__()
4          torch.manual_seed(12345)
5          self.conv1 = GCNConv(num_node_features, hidden_channels)
6          self.conv2 = GCNConv(hidden_channels, hidden_channels)
7          self.conv3 = GCNConv(hidden_channels, hidden_channels)
8          self.lin = Linear(hidden_channels, num_classes)
9
10     def forward(self, x, edge_index, batch):
11         # 1. Obtain node embeddings
12         x = self.conv1(x, edge_index)
13         x = x.relu()
14         x = self.conv2(x, edge_index)
15         x = x.relu()
16         x = self.conv3(x, edge_index)
17
18         # 2. Readout layer
19         x = global_mean_pool(x, batch)  # [batch_size, hidden_channels]
20
21         # 3. Apply a final classifier
22         x = F.dropout(x, p=0.5, training=self.training)
23         x = self.lin(x)
24
25     return x

```

Listing 9: Der Code für das GCN-Modell

4.2.3. Code für die Klassifikation des Graphen

Das oben aufgeführte Modell 9 wird verwendet, um die Klassifikation des Graphen zu bestimmen. Diese wird in zwei Schritten durchgeführt:

1. Aufbereitung der Eingabe
2. Klassifikation des Graphen

```

1 # classify the graph
2 def classify(self, G):
3     node_labels = np.arange(G.number_of_nodes())
4     degrees = nx.degree_centrality(G)
5     betweenness = nx.betweenness_centrality(G)
6     attrs = dict(zip(node_labels, node_labels))
7     nx.set_node_attributes(G, attrs, "label")
8     nx.set_node_attributes(G, degrees, "degree")
9     nx.set_node_attributes(G, betweenness, "betweenness")
10    # convert to torch tensor
11    x = from_networkx(graph, group_nodeAttrs=["label", "betweenness", "degree"])
12    test_loader = DataLoader([x], batch_size=1, shuffle=False)
13    # run the model
14    for x in test_loader:
15        out = self.model(x, x.edge_index, x.batch)
16        # convert to numpy
17        pred = out.argmax(dim=1)  # Use the class with highest probability
18        # return the result
19        return pred

```

Listing 10: Code zur Klassifikation des Graphen

4.2.4. Empfehlungen der topologischen Messwerte

In der Konsole werden, zusammen mit der vorhergesagten Klasse des Graphen, die vorgeschlagenen topologischen Indizes ausgegeben.

```
1 # get the topological indices
2 def get_topological_indices(self, graph):
3     # classify the graph
4     y = self.classify(graph)
5     # get the topological indices
6     topological_indices = y[0]
7     # return the topological indices
8     return topological_indices
```

Listing 11: Code für den Vorschlag der topologischen Indizes

In einem späteren Schritt wird zusammen mit den vorgeschlagenen topologischen Indizes eine Erklärung geliefert.

Die Idee ist, dass die Erklärung zur *Explainability* beiträgt. Als Kontext sind zwei Punkte relevant: zum einen die topologischen Indizes, welche vorgeschlagen werden, und zum anderen die vorhergesagte Klasse des Graphen.

5. Diskussion

Die Resultate, die Implementierung und die Tests werden im Folgenden reflektiert und diskutiert. Zunächst erfolgt eine Analyse der erarbeiteten Resultate, anschliessend wird der Ansatz von Ma et al. kritisch betrachtet und mögliche Verbesserungen werden vorgeschlagen. Danach wird ein persönliches Fazit zur Arbeit gezogen, indem deren Verlauf bewertet wird. Schliesslich wird ein Ausblick gegeben, es werden potenzielle zukünftige Schritte und alternative Vorgehensweisen erörtert.

5.1. Analyse

Da die Erarbeitung der Graphen sowie die Klassifikation essenzielle Teile der Arbeit sind, werden diese ebenfalls diskutiert. Dann werden die Resultate der PCA-Methode analysiert, welche für die Analyse der topologischen Masse verwendet wurden. Auch werden die Resultate der Klassifikation untersucht.

Die Methodik der Arbeit wurde mit Fokus auf der statistischen Analyse und dem Einsatz von Machine Learning durchgeführt.

Bereits in Kapitel 3 wurden die Hypothesen vorgestellt und Tests zu diesen durchgeführt und diskutiert. In diesem Kapitel liegt der Schwerpunkt auf der Beantwortung der Forschungsfragen sowie der Diskussion der Resultate.

5.1.1. Beantwortung der Forschungsfragen

Forschungsfrage 1

F1: Wie können verschiedene topologische Indizes sinnvoll miteinander verglichen werden?

Diese Frage lässt sich durch die Analyse der Resultate zu den topologischen Indizes beantworten. Die Korrelation der topologischen Indizes zu messen, war ein wesentlicher Schritt, um die topologischen Indizes miteinander zu vergleichen. Stark korrelierende topologische Indizes wurden identifiziert, welche dann in der Analyse und den Resultaten zu den topologischen Indizes visualisiert und erklärt wurden.

Es kam dabei heraus, dass die topologischen Indizes je nach Graphenklasse unterschiedliche Werte aufweisen. Muster konnten bereits in der explorativen Datenanalyse gefunden werden. Hier wurden die Werte der topologischen Indizes für die Graphenklassen in Abhängigkeit der Anzahl Knoten visualisiert. Die Resultate sind in 3.3 zu sehen. Für die Random-Graphenklasse ist auf Abbildung A.1 zu erkennen, wie der Wiener-Index bei geringer Knotenzahl stark variiert, während er aber bei mehr Knoten fast konstant bleibt. Bei den Small-World-Graphen ist auf Abbildung A.2 der Hosoya-Z-Index besonders spannend, er weist ein sprunghaftes Verhalten auf.

Bereits in der explorativen Datenanalyse konnten ähnliche Kurven, respektive Werte für die topologischen Indizes, für die Graphenklassen gefunden werden. Als Nächstes folgte die Korrelationsanalyse der topologischen Indizes, welche in Kapitel 3 vorgestellt wurde. Mithilfe der Spearman-Korrelation wurden die topologischen Indizes innerhalb der Graphenklassen \mathcal{N} miteinander verglichen. Die Heatplots auf Abbildung 3.5 - 3.11 zeigten, dass bereits einige Indizes untereinander perfekt monoton aufsteigend und absteigen miteinander korrelieren. Die paarweisen

Korrelationen sind auf den Abbildungen A.8 - A.14 im Anhang A.1.2 zu sehen. Besonders eindrücklich ist die Korrelation aller Indizes der Graphenklasse `line` welche ausser dem CII-, Hosoya-Z- und dem Szeged-Index alle monoton aufsteigen korrelieren. Dies ist auf Abbildung A.5 besonders gut zu sehen.

Forschungsfrage 2

F2: Wie kann ein nützlicher topologischer Index Φ für die Eingabe eines Netzwerkes g berechnet werden?

Diese Frage wurde durch die Definition des **nützlichen topologischen Index** Φ beantwortet. Die Entscheidung fiel auf die Definition 3 in Kapitel 3.

Nach der Aufarbeitung der Literatur und Theorie, sowie der Korrelationsanalyse der topologischen Indizes folgte die PCA-Methode für die Analyse der Einflüsse der topologischen Indizes auf die Hauptkomponenten. Mittels der PC-Methode wurden die 7-dimensionalen topologischen Indizes auf vier Dimensionen reduziert. Diese vier Dimensionen sind für über 95 % der Varianz zuständig 3.12a - 3.12n.

Die Loading-Plots wurden zur vereinfachten Darstellung für zwei Hauptkomponenten erstellt. Auf diesen wird ersichtlich, dass einige topologische Indizes stark auf die Hauptkomponenten wirken. Beispielsweise ist auf Abbildung 3.13d der Einfluss des Estrada-Index auf die 2. Hauptkomponente der Graphenklasse `complete` stark ausgeprägt. Die Analyse führt schlussendlich zur Aussage, dass der Einfluss der topologischen Indizes auf die Hauptkomponenten einen Beitrag zur Definition des **nützlichen topologischen Index** Φ leistet.

Die aufgestellte Formel 3.2 und die Berechnungen in 3.3 lässt dazu führen, einen **nützlichen topologischen Index** Φ zu definieren.

Dieser Begriff der **Usefulness** ist ein wichtiger Bestandteil der Arbeit. Er ist jedoch nicht vollkommen definiert. Da es sich um ein komplexes Problem handelt, ist es schwierig, eine vollständige Definition zu finden. Die Arbeit soll damit aber einen Beitrag zum Problem leisten.

Forschungsfrage 3

F3: Kann durch Einsatz von Machine Learning das Vergleichen von topologischen Netzwerkmesswerten optimiert werden?

Die Antwort auf diese Frage ist ein klares Ja. Durch den Einsatz der Graphenklassifikation mit einem GCN konnte ein starker Mehrwert erzielt werden. Das neuronale Netz hat sich als gutes Werkzeug für die Klassifikation der Graphen erwiesen. Es ist festzustellen, dass ein grosses

Potenzial für weitere Anwendungen besteht. Die Klassifikation der Graphen mit einem GCN hat hervorragend funktioniert. Das Netz hat eine hohe Genauigkeit erreicht und die Graphenklassen gut voneinander getrennt. Die Resultate sind in Kapitel 3 zu sehen. Auf Abbildung 3.15 ist zu sehen, dass in bereits wenigen Epochen konnte eine hohe Genauigkeit erreicht werden. Es wäre denkbar, dass in der Arbeit von Ma et al. **Meta-Indizes** wie die *Structure Sensitivity*, die *Uniqueness* oder die *Abruptness* als Features verwendet werden.

5.1.2. Datenaufbereitung

Es wurde mit Graphen aus sieben Klassen gearbeitet: Erdös-Rényi Random, Small-World, Scale-free, Tree, Complete, Path und Star. Diese Graphen wurden mithilfe der Python-Bibliothek NetworkX [93] erzeugt. Aufgrund von früh im Studium gesammelter Erfahrung mit der Erzeugung von Graphen war es kein Problem, diese durchzuführen.

Bereits bei der Generation der Graphen wurde der Code modular und wiederverwendbar parametrisiert, was in der explorativen Phase eine schnelle Erzeugung von Graphen mit verschiedenen Parametern ermöglichte.

Die Ermittlung der topologischen Indizes wurde in einem separaten Skript vorgenommen, welches die Graphen aus dem vorherigen Schritt einliest und die topologischen Indizes berechnet. Es wurde mit einer Datenstruktur gearbeitet, welche die Graphen und die zugehörigen topologischen Indizes speichert.

5.1.3. Statistische Analyse

Zu Beginn wurde die explorative Datenanalyse durchgeführt, bei der erste Erkenntnisse gesammelt wurden. Nach dem Normalisieren der Werte der topologischen Indizes und der bereits vorhanden sauberen Datenstruktur wurden die Daten visualisiert. In 3.3 ist zu sehen, wie sich der topologische Messwert einer Klasse zur Anzahl Knoten verhält.

Nach der explorativen Datenanalyse folgte die statistische Analyse.

Es bestand bereits eine Vermutung zur Korrelation der topologischen Indizes, diese wurde mit der Spearman-Korrelation verifiziert. Hier ging es nicht mehr nur um die Korrelation der Anzahl Knoten zu den topologischen Indizes, sondern um die der topologischen Indizes untereinander. Aus den Korrelationsplots (3.5-3.11) ist ersichtlich, dass einige topologische Indizes miteinander korrelieren. Wie bereits in einigen vorherigen Werken erwähnt, ist die Suche nach topologischen Indizes mit einer hohen Eindeutigkeit ein anhaltender Forschungsgegenstand [17, 107].

Es wurde die PCA-Methode genutzt, um eine Reihe von relevanten topologischen Massen aus der Literatur nach ihrem Einfluss auf die Hauptkomponenten innerhalb der Klasse zu untersuchen.

5.1.4. Klassifikation

Mit einem GCN [85] wurde ein Modell trainiert, um einen Graphen in eine der sieben Klassen zu klassifizieren. Da die Graphen bei der Erzeugung direkt nach ihrer Klasse erstellt wurden, waren die gekennzeichneten Daten für das Training des neuronalen Netzes vorhanden. Es wurde die Python-Bibliothek PyTorch [105] mit der Erweiterung PyTorch Geometric [104] für die Implementierung des neuronalen Netzes verwendet.

Das Training des neuronalen Netzes wurde in einem separaten Skript durchgeführt, welches die Graphen aus dem vorherigen Schritt einliest und das Modell trainiert. Insgesamt dauerte das Training mit 170 Epochen 62 Minuten auf einem Desktop-PC mit einem „24 Core AMD Ryzen 9 3900X 4.3GHz“-Prozessor und einer „NVIDIA GeForce RTX 2080 Ti“-Grafikkarte 3.4. Zu Beginn war die Genauigkeit des Modells äusserst niedrig. Ein Grund dafür ist, dass bei den Small-World- und Random-Graphen zu ähnliche Parameter verwendet wurden. Dies führte zu einer besonders hohen Ähnlichkeit der Graphen, was das Modell verwirrte.

Wie bereits in 3.4 beschrieben, wurden die Hyper-Parameter des neuronalen Netzes angepasst, um die Genauigkeit schliesslich auf über 92 % zu erhöhen. Dies ist ein hoher Wert. Die Tests haben gezeigt, dass das Modell ausgezeichnet in der Lage ist, die Graphen zu klassifizieren.

5.1.5. Vorschlag zum Ansatz von Ma et al.

Aktuell werden die Meta-Indizes für die Berechnung der Usefulness von topologischen Massen verwendet. Eine Erweiterung des Feature-Vektors, welcher je nach Klasse des Graphen anders aufgebaut ist, würde nach dieser Arbeit Sinn ergeben. Die Meta-Indizes versuchen das Problem der unterschiedlichen Graphen zu lösen, indem sie die topologischen Masse in einen gemeinsamen Nenner bringen. Doch auch bei den Meta-Indizes gibt es je nach Klasse unterschiedliche Prioritäten zur Berechnung der Usefulness.

Als Beispiel ist ein Transportations-Graph zu nennen. Die meisten Graphen sind nicht dicht und haben viele Knoten mit einer geringen Anzahl Nachbarn. In der Literatur werden ihre Adjazenzmatrizen *sparse* genannt. Dir vorliegenden Arbeit liegt die Definition aus [108] zugrunde, welche auch am NIST (National Institute of Standards and Technology) verwendet wird.

Scott et al. [109] haben 2006 einen Network-Robustness-Index (NRI) vorgeschlagen. Die Problematik beim NRI ist, dass er primär für Transportsysteme geeignet ist. In Bezug diese Arbeit und die gewünschten Resultate in der quantitativen Graphenanalyse besteht das Interesse an generellen Graphen.

Während der Simulation und des Aufbaus der Datenstruktur zur weiteren Bearbeitung ist aufgefallen, dass die topologischen Indizes je nach Komplexität des Graphen und des topologischen

Index eine grosse Rolle bei der Berechnung spielen. Deshalb empfiehlt der Autor der vorliegenden Arbeit, die Komplexität der topologischen Indizes ebenfalls als Meta-Index zu bewerten.

Man betrachte insbesondere den Wiener-Index, welcher über

$$W = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n d_{ij} \quad (5.1)$$

berechnet wird und bei dem d_{ij} die kürzeste Distanz zwischen Knoten i und Knoten j ist. Dies ist bei einem Graphen mit n Knoten überaus aufwendig und ergibt eine Laufzeitkomplexität $W(G)$ von $O(n^3 \log n)$, n ist die Anzahl Knoten von G .

Somit ist aus der explorativen Datenanalyse ersichtlich, dass die Rechenzeit bei Hypercubes der 6. und höheren Dimensionen hoch ist.

5.2. Fazit

Diese Arbeit soll mit der Herleitung und Definition der Usefulness eines topologischen Indizes einen neuen Ansatz für die quantitativen Graphenanalysen liefern. Die Erkenntnisse aus der Arbeit können in die Forschung rund um das Thema **Usefulness** von topologischen Indizes einfließen. Mit der Korrelation zwischen den topologischen Indizes innerhalb einer Graphenklasse konnte bestehende Schwierigkeiten und Ähnlichkeiten aufgezeigt werden.

Mit meinen Resultaten, dem Ansatz der Klassifikation von Graphen und der Analyse der Einflüsse der Indizes auf die Hauptkomponenten der PCA-Methode, wurde ein neuer Ansatz für die quantitativen Graphenanalysen gefunden. Neben Ansatz von Ma et al. und diesem Ansatz benötigt es noch weitere Resultate, um die Usefulness sinnvoll zu definieren und einen Standard zu schaffen. Die Usefulness für topologische Indizes ist ein überaus komplexes Thema, quantitativ ist es kompliziert zu definieren.

Die Arbeit war ausserordentlich interessant und hat viel Neues gezeigt. Wir haben viel über Graphen und topologische Masse gelernt und konnten uns mit der Literatur auseinandersetzen.

Die Aufarbeitung der Literatur zeigt, dass es zahlreiche topologische Masse gibt. Die Klassifikation von Graphen geht weit zurück und wurde mit vielen verschiedenen Methoden versucht. Gerade in der heutigen Zeit ist es aufgrund der neuronalen Netze faszinierend, die Klassifizierung von Graphen zu untersuchen.

Die Wahl der topologischen Masse fiel auf diverse ältere topologische Indizes. Diese sind in Kapitel 2 [2.4](#) beschrieben. Die Hauptliteratur für die Erarbeitung der topologischen Indizes war [6]. Es wäre interessant, modernere informationstheoretische topologische Indizes zu untersuchen.

5.3. Ausblick

Für zukünftige Werke in dem Bereich wäre es sinnvoll, den Ansatz der Klassifizierung zu wählen. Diese kann helfen, die Usefulness zu bewerten.

In weiteren Studien zur Analyse der **Usefulness** von topologischen Indizes wäre es interessant, moderne topologische Indizes und aktuelle Graphen-Neuronale Netze zu untersuchen. Des Weiteren könnten andere Graphenklassen untersucht und mehr Graphen als Trainingsdaten verwendet werden. Mit dem Code dieser Arbeit könnte ein solches Experiment weiter ausgebaut werden, sie bietet ein gutes Fundament für Machine-Learning getriebene Analyse von Graphen.

Ein ähnlicher Ansatz für die Graph-Embeddings beim GCN könnte das Berechnen der Usefulness unterstützen. Hier könnte neben der Clustering-Methode aus [10] ebenfalls ein GCN verwendet werden. Die Meta-Indizes könnten als Node-Features genutzt werden.

Zusammenfassend kann gesagt werden, dass diese Bachelor-Thesis wichtige Erkenntnisse über die Verwendung von topologischen Indizes in der Graphentheorie geliefert hat. Durch die Untersuchung verschiedener Indizes für eine Vielzahl von Graphen konnten nützliche Einblicke gewonnen werden, welche Indizes für bestimmte Graphenklassen am besten geeignet sind.

Die entwickelte Anwendung, die den einflussreichsten topologischen Index für eine bestimmte Klasse von Graphen bestimmt, hat das Potenzial, ein wertvolles Werkzeug für Forscher und Praktiker zu sein, die mit Graphdaten arbeiten. Sie könnte in verschiedenen Bereichen wie der Molekular-, Bioinformatik, Sozial- und Synthetischen Graphen eingesetzt werden.

Die Ergebnisse dieser Arbeit können dazu beitragen, Forschung und Praxis in der Graphenanalyse zu beeinflussen und somit auch mögliche zukünftige Entwicklungen in diesem Bereich vorantreiben. Es bleibt zu hoffen, dass die gewonnenen Erkenntnisse zu neuen Forschungsansätzen führen.

A. Anhang

A.1. Ergänzende Darstellungen zu den Resultaten

A.1.1. Vergleich der topologischen Indizes pro Klasse

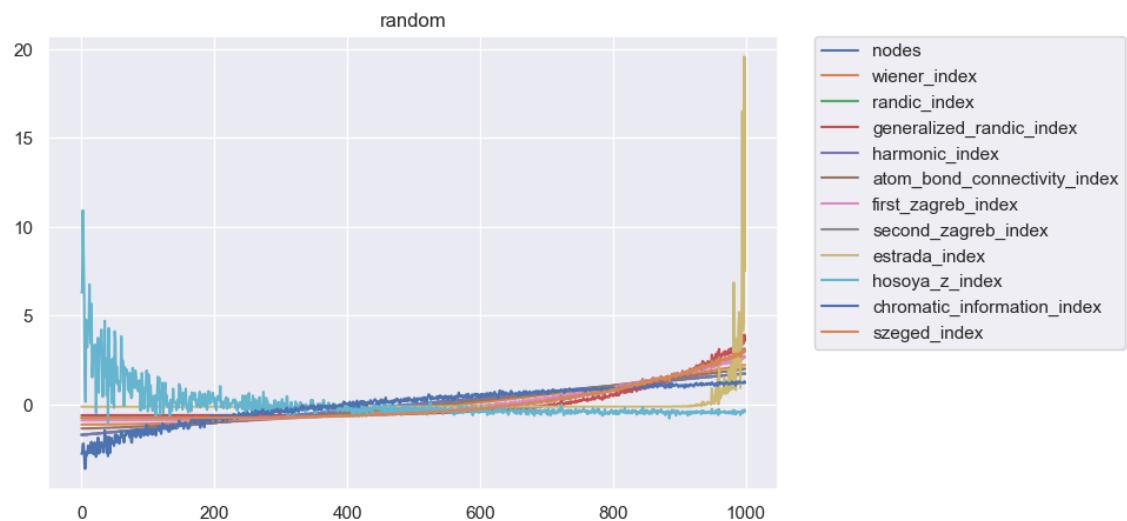
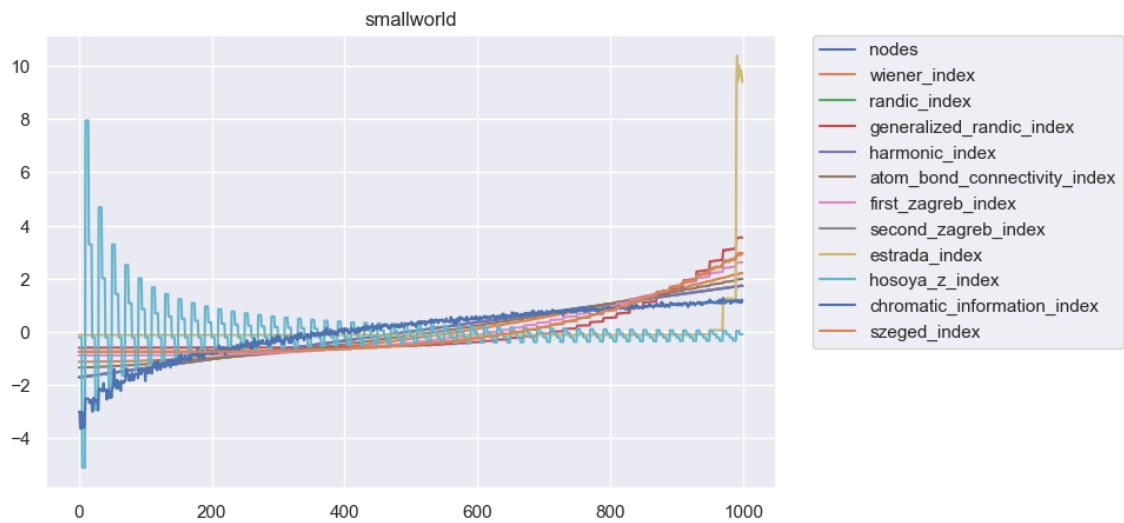
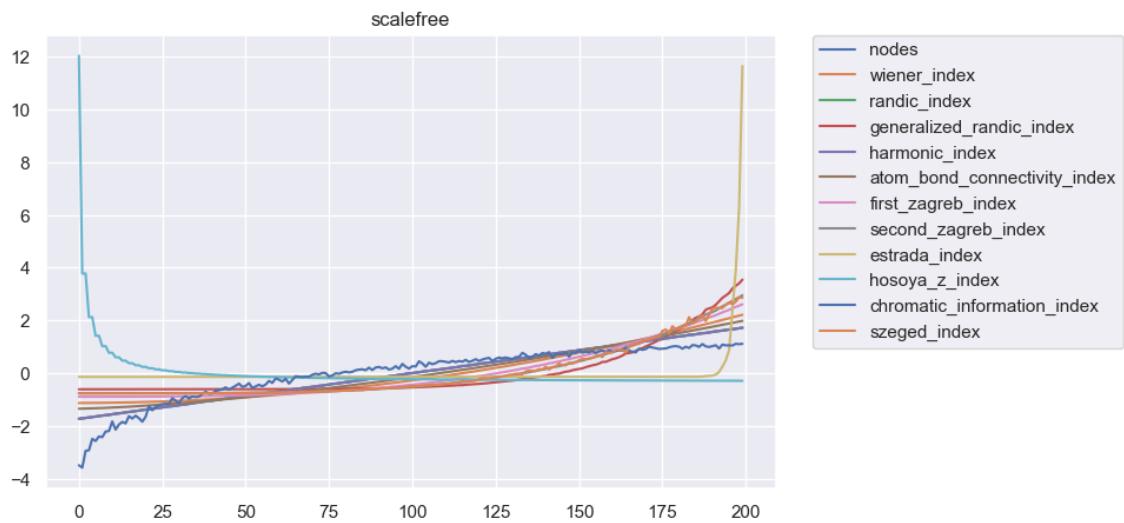
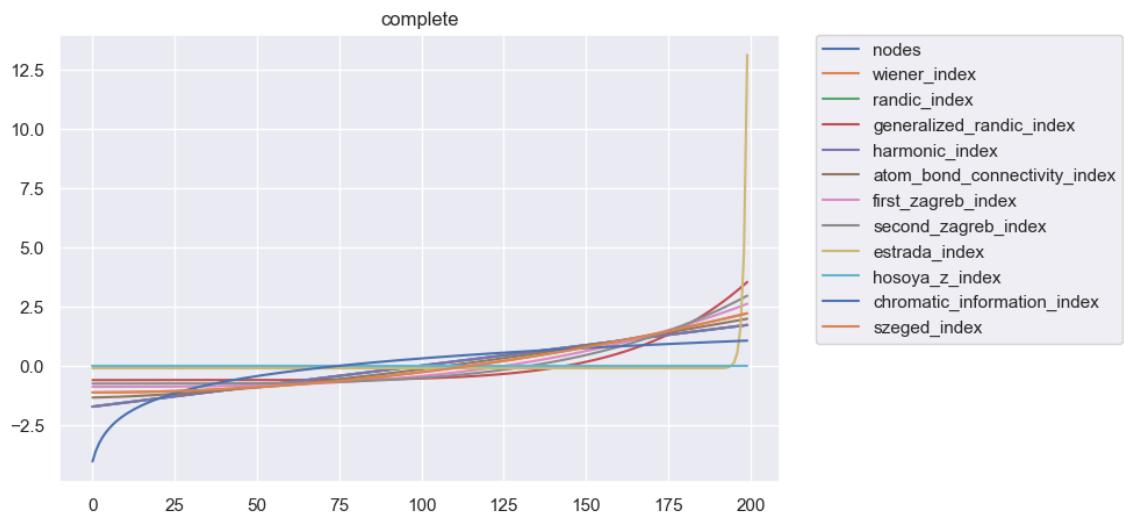
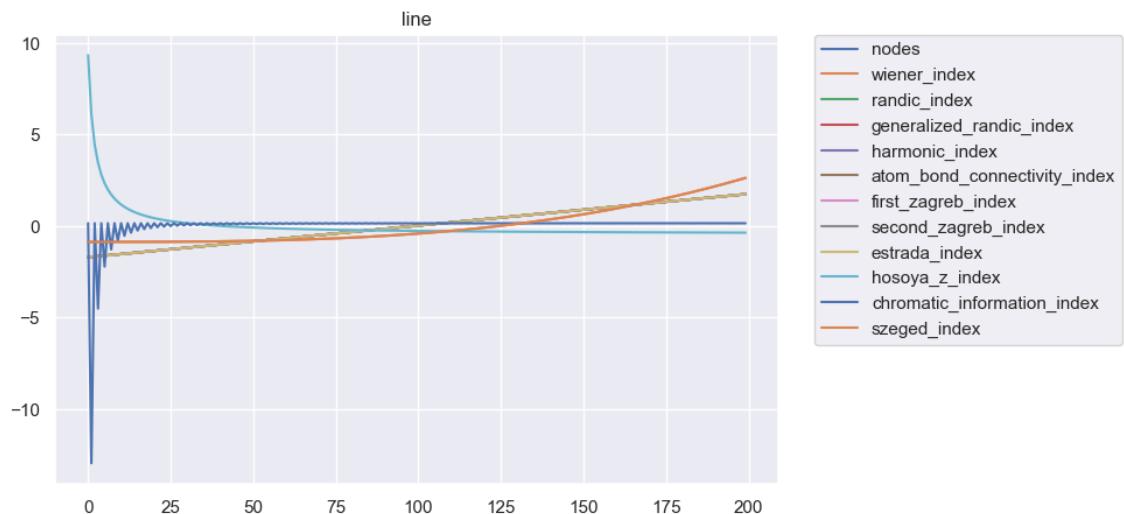


Abbildung A.1.: Topologische Indizes der Klasse random

Abbildung A.2.: Topologische Indizes der Klasse *smallworld*Abbildung A.3.: Topologische Indizes der Klasse *scalefree*

Abbildung A.4.: Topologische Indizes der Klasse *complete*Abbildung A.5.: Topologische Indizes der Klasse *line*

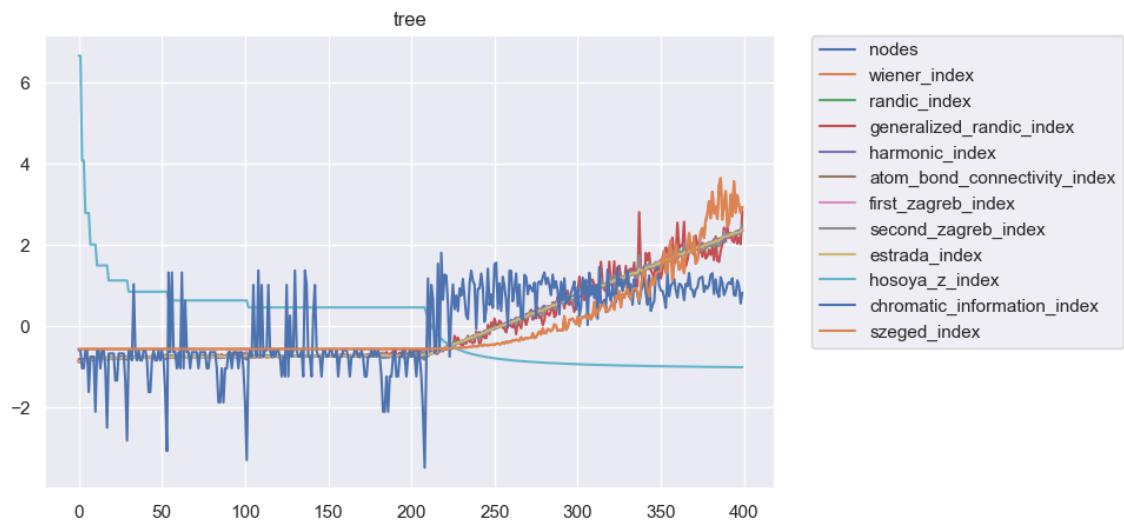


Abbildung A.6.: Topologische Indizes der Klasse tree

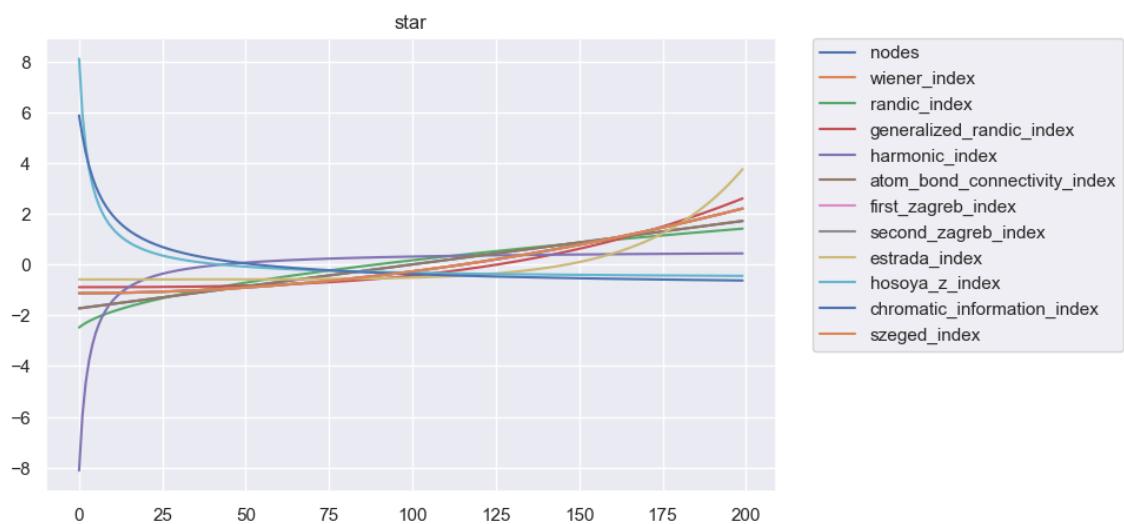


Abbildung A.7.: Topologische Indizes der Klasse star

A.1.2. Einzel-Vergleich der topologischen Indizes



Abbildung A.8.: Einzelne Vergleiche der topologischen Indizes der Klasse random

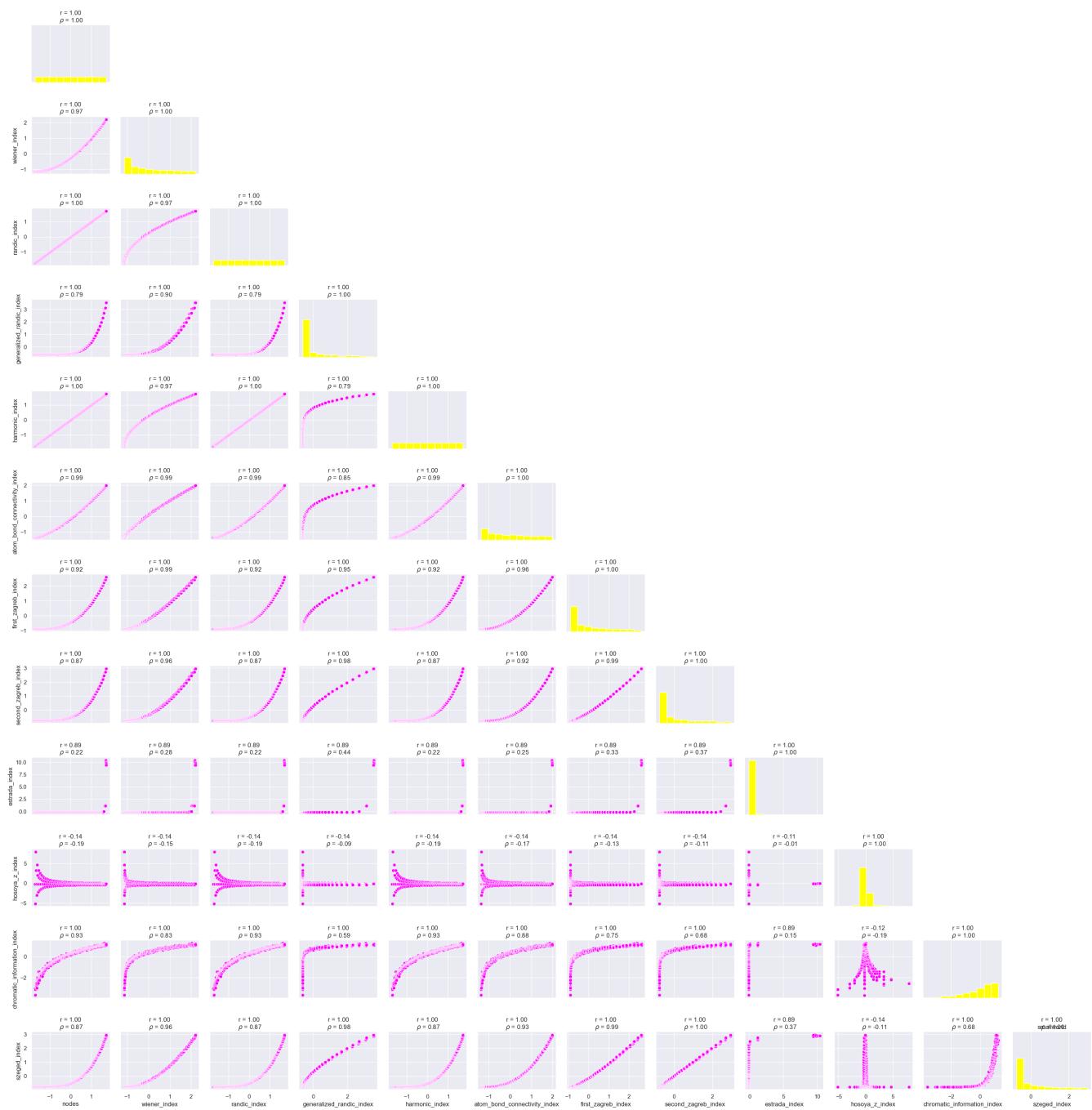
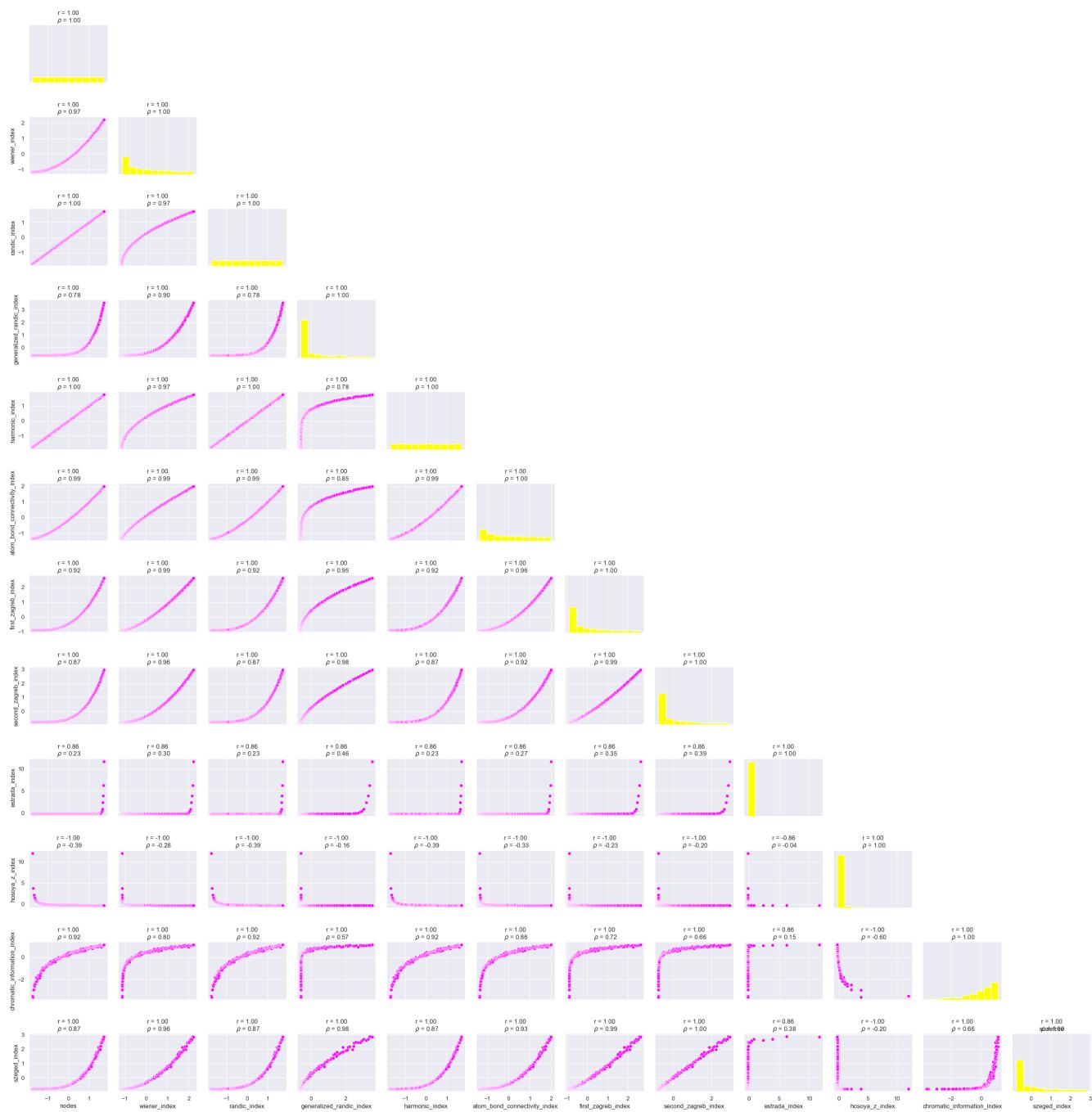


Abbildung A.9.: Einzelne Vergleiche der topologischen Indizes der Klasse smallworld

Abbildung A.10.: Einzelne Vergleiche der topologischen Indizes der Klasse *scalefree*

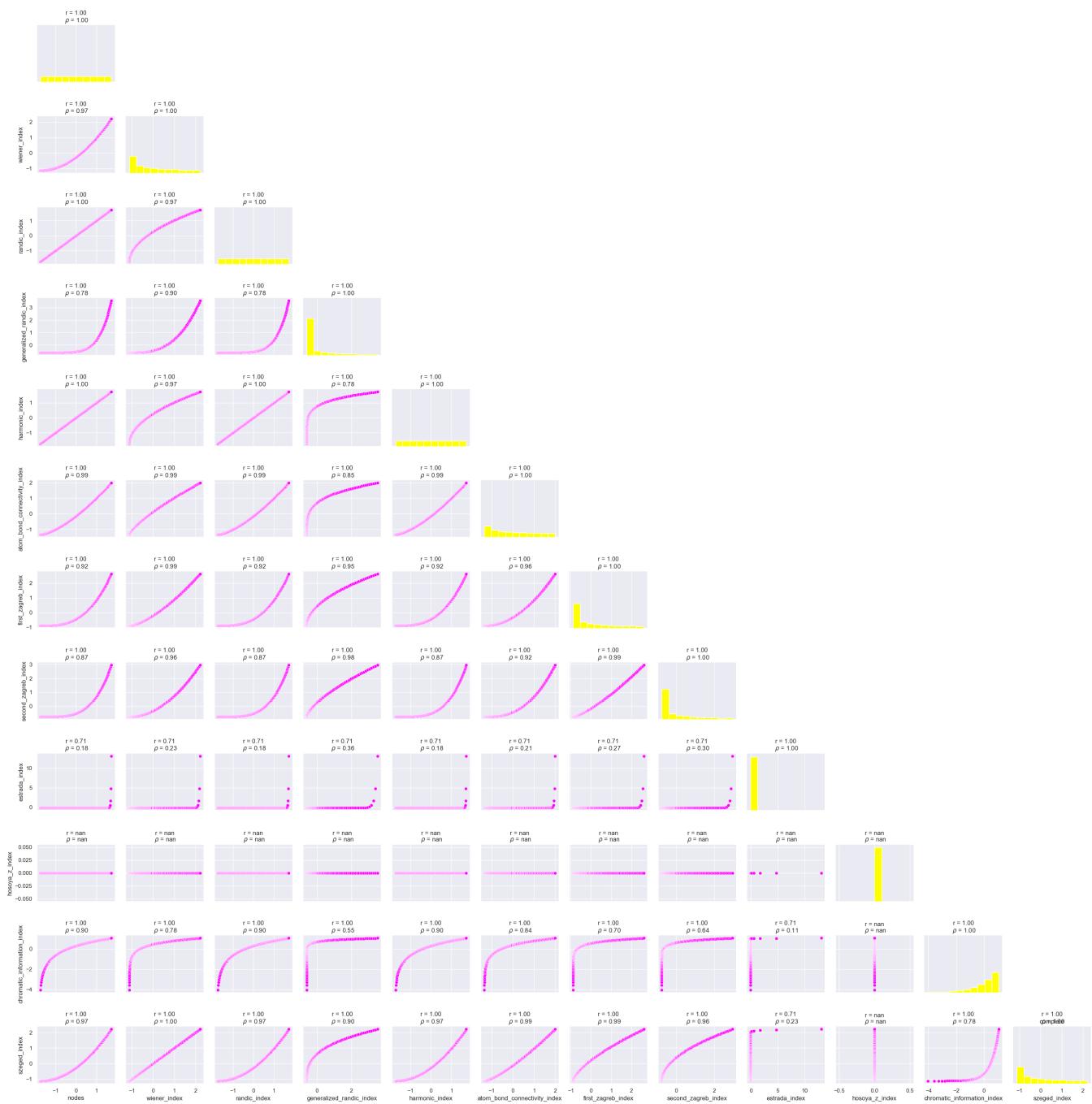
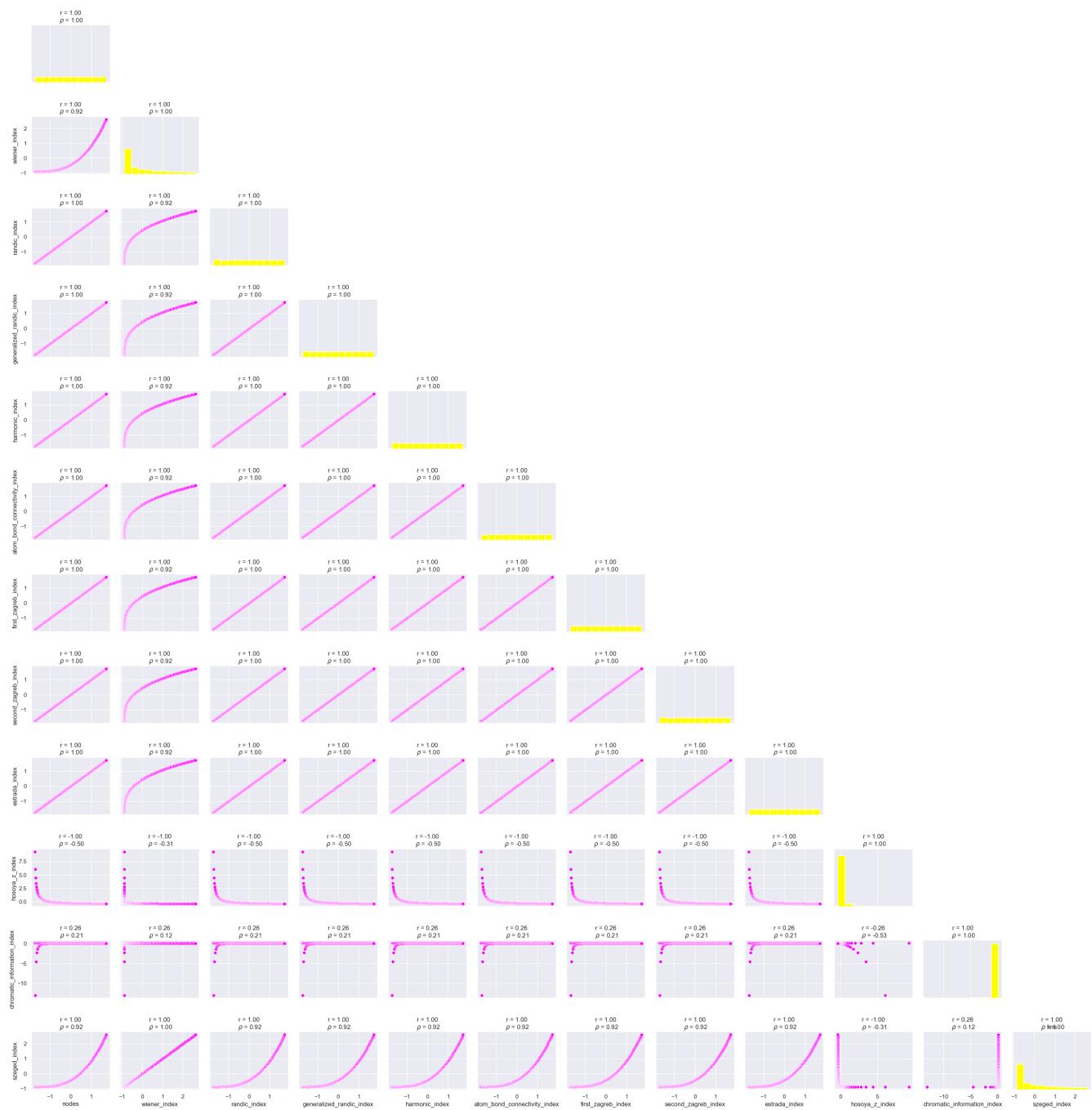


Abbildung A.11.: Einzelne Vergleiche der topologischen Indizes der Klasse complete

Abbildung A.12.: Einzelne Vergleiche der topologischen Indizes der Klasse *line*

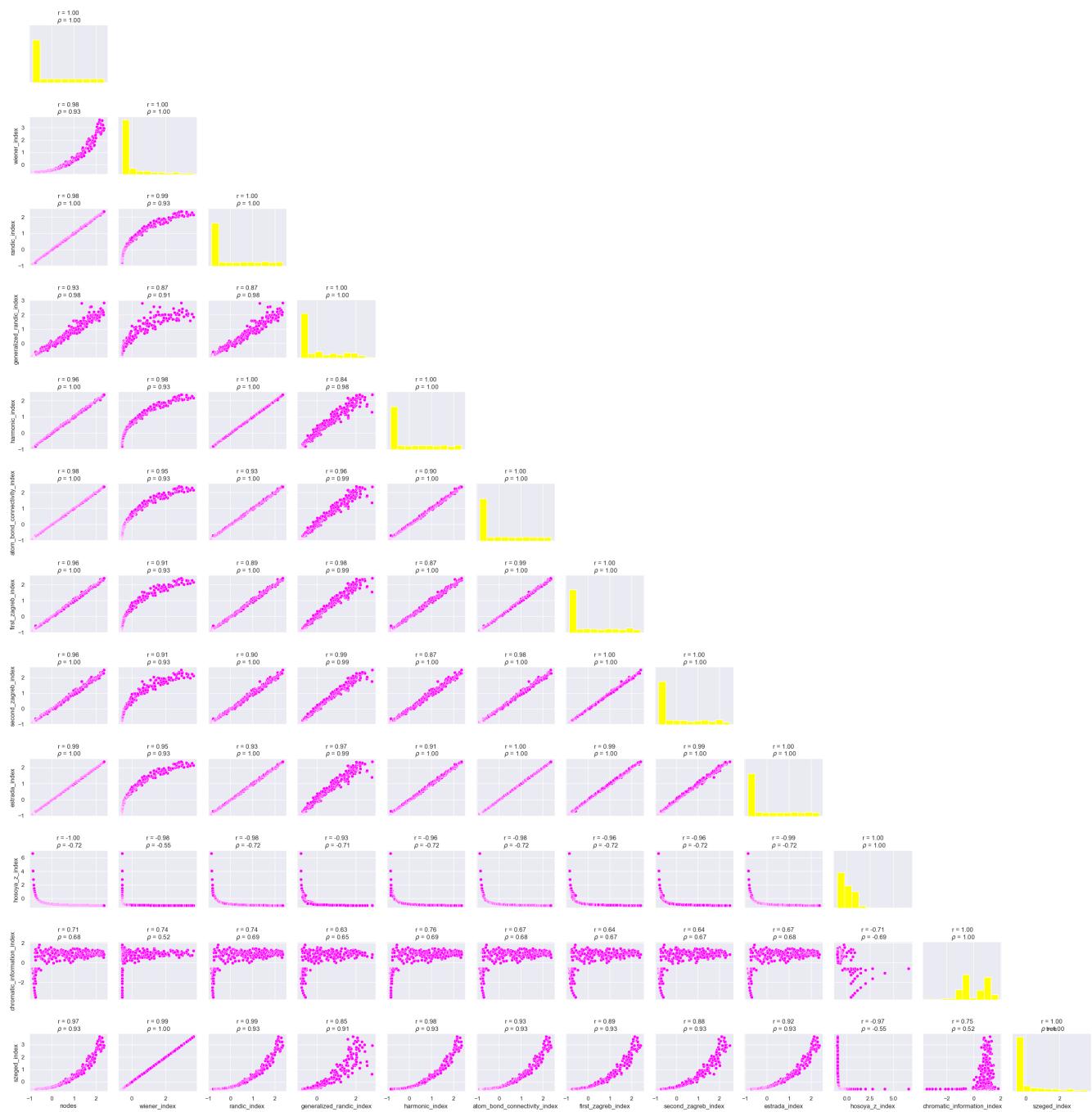


Abbildung A.13.: Einzelne Vergleiche der topologischen Indizes der Klasse tree

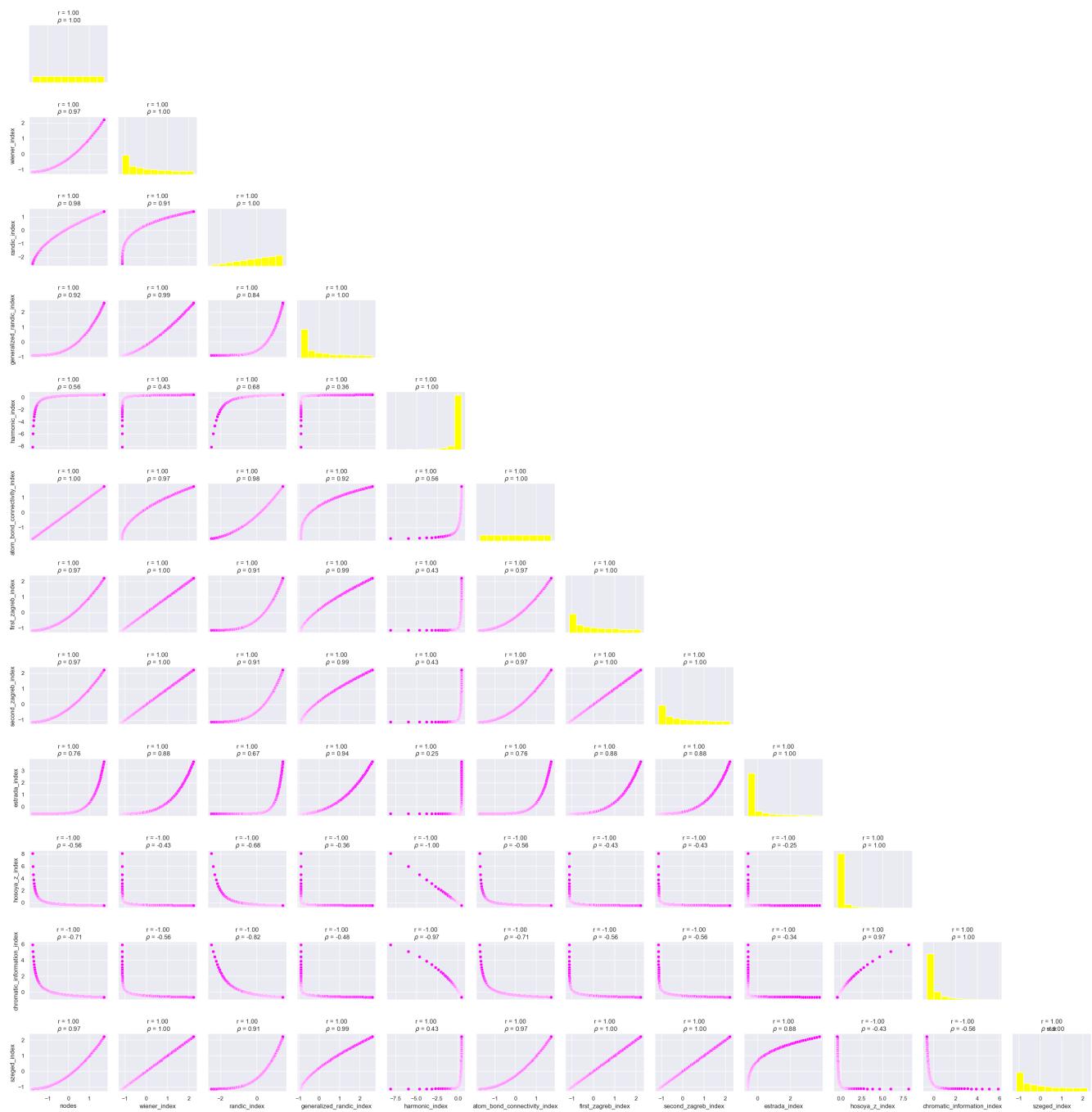


Abbildung A.14.: Einzelne Vergleiche der topologischen Indizes der Klasse star

A.1.3. Einfluss aller topologischen Indizes auf die Hauptkomponenten

83

```

1 topological_indices_all_graphs key: random, shape: 1000
2 ----- PCA random 4 components -----
3      wiener    randic   generalized_randic   harmonic    abc  1st zagreb  2nd zagreb  estrada      z      cii      szeged
4 PC-1  0.338    0.331            0.310    0.331  0.337      0.334      0.327  0.119 -0.172  0.297  0.327
5 PC-2  0.007   -0.165            0.303   -0.165 -0.068      0.121      0.200  0.554  0.575 -0.342  0.198
6 PC-3 -0.113   -0.019           -0.075   -0.019 -0.077      -0.131     -0.123  0.784 -0.544  0.128 -0.127
7 PC-4  0.057    0.258           -0.379    0.258  0.166      -0.103     -0.230  0.239  0.576  0.431 -0.235
8
9 topological_indices_all_graphs key: smallworld, shape: 1000
10 ----- PCA smallworld 4 components -----
11      wiener    randic   generalized_randic   harmonic    abc  1st zagreb  2nd zagreb  estrada      z      cii      szeged
12 PC-1  0.344    0.334            0.317    0.334  0.342      0.341      0.334  0.119 -0.059  0.290  0.334
13 PC-2 -0.034   -0.172            0.253   -0.172 -0.096      0.071      0.147  0.649  0.556 -0.298  0.143
14 PC-3  0.056    0.088           -0.082    0.088  0.076      0.020     -0.018 -0.537  0.813  0.134 -0.016
15 PC-4 -0.042   0.212            -0.351    0.212  0.072      -0.190     -0.278  0.517  0.159  0.546 -0.278
16
17 topological_indices_all_graphs key: scalefree, shape: 200
18 ----- PCA scalefree 4 components -----
19      wiener    randic   generalized_randic   harmonic    abc  1st zagreb  2nd zagreb  estrada      z      cii      szeged
20 PC-1  0.342    0.333            0.314    0.333  0.340      0.339      0.331  0.123 -0.123  0.287  0.332
21 PC-2  0.019   -0.151            0.279   -0.151 -0.055      0.122      0.192  0.457  0.640 -0.411  0.184
22 PC-3 -0.109   -0.079           -0.021   -0.079 -0.105      -0.090     -0.056  0.815 -0.526  0.086 -0.064
23 PC-4  0.040    0.275           -0.388    0.275  0.154      -0.138     -0.260  0.323  0.511  0.395 -0.256
24
25 topological_indices_all_graphs key: complete, shape: 200
26 ----- PCA complete 4 components -----
27      wiener    randic   generalized_randic   harmonic    abc  1st zagreb  2nd zagreb  estrada      z      cii      szeged
28 PC-1  0.346    0.338            0.313    0.338  0.345      0.341      0.332  0.094  0.0  0.284  0.346
29 PC-2 -0.026   -0.153            0.251   -0.153 -0.085      0.070      0.144  0.879 -0.0  -0.291 -0.026
30 PC-3 -0.073   0.201            -0.410    0.201  0.045      -0.233     -0.326  0.459 -0.0  0.605 -0.073
31 PC-4  0.233    0.202           -0.550    0.202  0.271      0.036     -0.188  0.088  0.0 -0.624  0.233
32
33

```

```

1      topological_indices_all_graphs key: line, shape: 200
2 ----- PCA line 4 components -----
3      wiener randic generalized_randic harmonic    abc 1st zagreb 2nd zagreb estrada      z      cii szeged
4 PC-1 -0.311 -0.331          -0.331 -0.331 -0.331      -0.331 -0.331 -0.331 0.174 -0.083 -0.311
5 PC-2  0.179  0.024          0.025  0.025  0.024      0.024  0.024  0.024 0.611 -0.748  0.179
6 PC-3  0.239 -0.037          -0.036 -0.037 -0.038      -0.037 -0.037 -0.037 0.670  0.653  0.239
7 PC-4 -0.560  0.176          0.177  0.177  0.175      0.176  0.176  0.176 0.384  0.086 -0.560
8
9 topological_indices_all_graphs key: tree, shape: 400
10 ----- PCA tree 4 components -----
11      wiener randic generalized_randic harmonic    abc 1st zagreb 2nd zagreb estrada      z      cii szeged
12 PC-1  0.299  0.320          0.315  0.319  0.320      0.319  0.319  0.320 -0.240  0.227  0.299
13 PC-2  0.328  0.034          0.054  0.031  0.040      0.047  0.047  0.040 0.588 -0.653  0.328
14 PC-3  0.094 -0.006          -0.053 -0.001 -0.019      -0.032 -0.032 -0.018 0.691  0.706  0.095
15 PC-4 -0.533  0.146          0.354  0.141  0.153      0.185  0.216  0.162 0.341 -0.136 -0.533
16
17 topological_indices_all_graphs key: star, shape: 200
18 ----- PCA star 4 components -----
19      wiener randic generalized_randic harmonic    abc 1st zagreb 2nd zagreb estrada      z      cii szeged
20 PC-1 -0.332 -0.329          -0.321 -0.215 -0.336      -0.332 -0.332 -0.282 0.215  0.253 -0.332
21 PC-2  0.160 -0.094          0.222 -0.526  0.027      0.160  0.160  0.281 0.526  0.455  0.160
22 PC-3  0.077  0.412          -0.184 -0.235  0.347      0.079  0.077 -0.734 0.235 -0.047  0.077
23 PC-4 -0.197  0.434          -0.214 -0.269  0.147      -0.196 -0.197  0.485 0.269 -0.458 -0.197
24

```

Listing 12: Ausgabe aller Komponenten der vier Hauptkomponenten der verschiedenen Netzwerk-Klassen

A.1.4. Erklärbare Varianzen und Usefulness-Scores aller Indizes

```

1 topological_indices_all_graphs key: random, shape: 1000
2 ----- PCA random 4 explained variance ratio -----
3     explained variance ratio
4 PC-1          0.784
5 PC-2          0.116
6 PC-3          0.069
7 PC-4          0.026
8 ----- PCA random combined usefulness score -----
9     usefulness score
10 wiener        0.747183
11 randic        0.899278
12 generalized_randic 0.985581
13 harmonic       0.899467
14 abc            0.833882
15 1st zagreb    0.915968
16 2nd zagreb    0.998606
17 estrada        0.000000
18 z               0.476098
19 cii            0.985445
20 szeged         1.000000
21
22 topological_indices_all_graphs key: smallworld, shape: 1000
23 ----- PCA smallworld 4 explained variance ratio -----
24     explained variance ratio
25 PC-1          0.760
26 PC-2          0.103
27 PC-3          0.084
28 PC-4          0.048
29 ----- PCA smallworld combined usefulness score -----
30     usefulness score
31 wiener        0.831004
32 randic        0.985409
33 generalized_randic 1.000000
34 harmonic       0.985451
35 abc            0.898649
36 1st zagreb    0.882126
37 2nd zagreb    0.940025
38 estrada        0.432969
39 z               0.000000
40 cii            0.977841
41 szeged         0.936464
42
43 topological_indices_all_graphs key: scalefree, shape: 200
44 ----- PCA scalefree 4 explained variance ratio -----
45     explained variance ratio
46 PC-1          0.765
47 PC-2          0.121
48 PC-3          0.076

```

```

49 PC-4          0.033
50 ----- PCA scalefree combined usefulness score -----
51           usefulness score
52 wiener        0.766124
53 randic         0.985386
54 generalized_randic 0.984622
55 harmonic       0.985271
56 abc            0.861029
57 1st zagreb    0.937208
58 2nd zagreb    1.000000
59 estrada        0.000000
60 z              0.092939
61 cii            0.989833
62 szeged         0.999622
63
64 topological_indices_all_graphs key: complete, shape: 200
65 ----- PCA complete 4 explained variance ratio -----
66           explained variance ratio
67 PC-1           0.830
68 PC-2           0.104
69 PC-3           0.057
70 PC-4           0.008
71 ----- PCA complete combined usefulness score -----
72           usefulness score
73 wiener         0.942523
74 randic         0.987024
75 generalized_randic 1.000000
76 harmonic       0.987024
77 abc            0.955097
78 1st zagreb    0.967715
79 2nd zagreb    0.990307
80 estrada        0.627966
81 z              0.000000
82 cii            0.974817
83 szeged         0.942523
84
85 topological_indices_all_graphs key: line, shape: 200
86 ----- PCA line 4 explained variance ratio -----
87           explained variance ratio
88 PC-1           0.822
89 PC-2           0.121
90 PC-3           0.041
91 PC-4           0.016
92 ----- PCA line combined usefulness score -----
93           usefulness score
94 wiener         1.000000
95 randic         0.848306
96 generalized_randic 0.848312
97 harmonic       0.848308
98 abc            0.848286
99 1st zagreb    0.848303

```

```

100 2nd zagreb          0.848307
101 estrada             0.848304
102 z                   0.584325
103 cii                0.000000
104 szeged              1.000000
105
106 topological_indices_all_graphs key: tree, shape: 400
107 ----- PCA tree 4 explained variance ratio -----
108     explained variance ratio
109 PC-1                 0.885
110 PC-2                 0.072
111 PC-3                 0.028
112 PC-4                 0.013
113 ----- PCA tree combined usefulness score -----
114     usefulness score
115 wiener               0.999889
116 randic               0.617134
117 generalized_randic   0.661762
118 harmonic              0.598685
119 abc                  0.649556
120 1st zagreb           0.686914
121 2nd zagreb           0.692432
122 estrada              0.651624
123 z                     0.305157
124 cii                  0.000000
125 szeged              1.000000
126
127 topological_indices_all_graphs key: star, shape: 200
128 ----- PCA star 4 explained variance ratio -----
129     explained variance ratio
130 PC-1                 0.779
131 PC-2                 0.193
132 PC-3                 0.024
133 PC-4                 0.003
134 ----- PCA star combined usefulness score -----
135     usefulness score
136 wiener               7.190176e-01
137 randic               4.641530e-01
138 generalized_randic   1.000000e+00
139 harmonic              2.433945e-15
140 abc                  7.673370e-03
141 1st zagreb           7.173088e-01
142 2nd zagreb           7.190176e-01
143 estrada              7.781882e-01
144 z                     0.000000e+00
145 cii                  5.466071e-01
146 szeged              7.190176e-01
147
148

```

Listing 13: Ausgabe aller erklärbaren Varianzen und Usefulness-Scores aller Indizes

A.1.5. Weights and Biases Dashboard für die Visualisierung der Ergebnisse

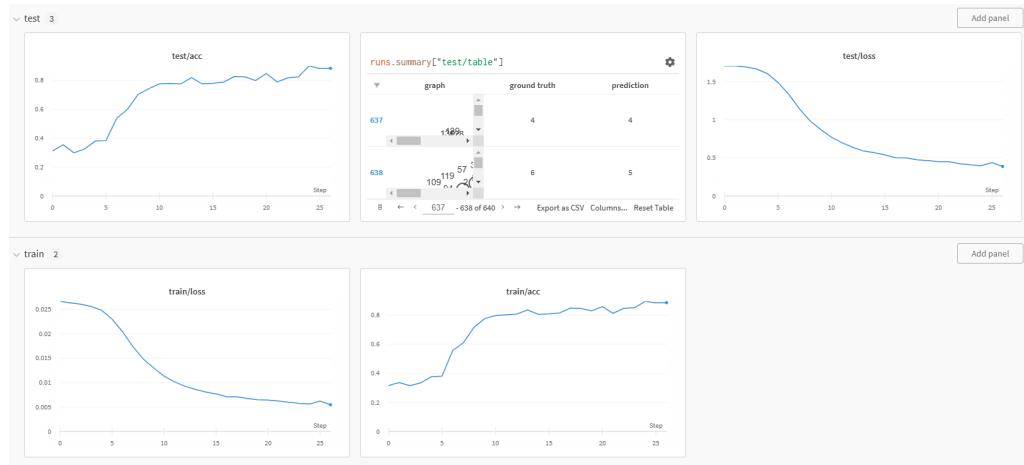


Abbildung A.15.: Weights and Biases Dashboard für die Visualisierung der Ergebnisse

Abbildungsverzeichnis

2.1. Übersicht zur Anordnung des Themengebietes, übersetzt aus Todeschini und Consonni [19]	6
2.2. Darstellung der verschiedenen Graphen	7
2.3. Eigenschaften der unterschiedlichen Netzwerkklassen (Quelle: Barabási [25, p. 97])	10
2.4. Drei Random-Graphen mit $p = 0.03$ und $N = 100$ (Quelle: Barabási. [25, p. 84])	11
2.5. Regular Lattice Graph (Quelle: Gayen [34])	12
2.6. Regular Lattice Graph mit $n = 10, m = 4$ und $p = 1$ (Quelle: Gayen [34])	13
2.7. Baum Graph	14
2.8. Sterngraph	15
2.9. K5-Graph	16
2.10. Path oder Line Graph	16
2.11. Nachweis von Lemma 2. G_1 und G_2 links, sind isomorphe Graphen. G_1 und G_3 in der rechten Abbildung sind nicht isomorphe Graphen, aber haben denselben Wiener Index	19
2.12. Graph \mathcal{G} mit den Knotengeraden (Quelle: [46, p. 351])	20
2.13. Modell für die Graphenklassifizierung (Quelle: [85])	28
2.14. Sample- und Aggregation-Ansatz visualisiert (Quelle: Hamilton [88])	29
3.1. Erarbeitung der Resultate	32
3.2. Visualisierung von Graphen mit NetworkX und Matplotlib	34
3.3. Vergleich des normalisierten Randić- und des generalisierten Randić-Index von allen non isomorphic graphs (Nauty) bis 10 Knoten	38
3.5. Spearman-Korrelation der topologischen Indizes der Klasse Random	40
3.6. Spearman-Korrelation der topologischen Indizes der Klasse Small-World	41
3.7. Spearman-Korrelation der topologischen Indizes der Klasse Scale-free	41
3.8. Spearman-Korrelation der topologischen Indizes der Klasse Complete	42
3.9. Spearman-Korrelation der topologischen Indizes der Klasse Line	42
3.10. Spearman-Korrelation der topologischen Indizes der Klasse Tree	43
3.11. Spearman-Korrelation der topologischen Indizes der Klasse Star	43
3.14. Modell für die Graphenklassifizierung (Quelle: Eigene Darstellung)	54
3.15. Genauigkeit des Modells während des Trainings	56

4.1. Ablauf der Anwendung für die Berechnung der nützlichen topologischen Indizes für ein Netzwerk	58
A.1. Topologische Indizes der Klasse <code>random</code>	72
A.2. Topologische Indizes der Klasse <code>smallworld</code>	73
A.3. Topologische Indizes der Klasse <code>scalefree</code>	73
A.4. Topologische Indizes der Klasse <code>complete</code>	74
A.5. Topologische Indizes der Klasse <code>line</code>	74
A.6. Topologische Indizes der Klasse <code>tree</code>	75
A.7. Topologische Indizes der Klasse <code>star</code>	75
A.8. Einzelne Vergleiche der topologischen Indizes der Klasse <code>random</code>	76
A.9. Einzelne Vergleiche der topologischen Indizes der Klasse <code>smallworld</code>	77
A.10. Einzelne Vergleiche der topologischen Indizes der Klasse <code>scalefree</code>	78
A.11. Einzelne Vergleiche der topologischen Indizes der Klasse <code>complete</code>	79
A.12. Einzelne Vergleiche der topologischen Indizes der Klasse <code>line</code>	80
A.13. Einzelne Vergleiche der topologischen Indizes der Klasse <code>tree</code>	81
A.14. Einzelne Vergleiche der topologischen Indizes der Klasse <code>star</code>	82
A.15. Weights and Biases Dashboard für die Visualisierung der Ergebnisse	88

Tabellenverzeichnis

3.1. Alle Graphen, die für die Arbeit generiert wurden und deren Anzahl Knoten.	34
3.2. Topologische Indizes unter Betrachtung	36
3.3. PCA-Komponenten der Erdős-Rényi-Klasse und der Einfluss der Indizes auf die Komponenten	50
3.4. Technische Komponenten für das Training	55

Definitionsverzeichnis

1.	Theorem (Topologischer Index)	17
2.	Lemma (Topologische Indizes und Isomorphie)	19
3.	Theorem (nützlicher topologischer Index Φ)	50

List of Listings

1.	Datenstruktur für die Testdaten	35
2.	Diese Funktion berechnet die topologischen Indizes für einen Graphen.	37
3.	Erklärbare Varianz der Hauptkomponenten der Klasse Random	51
4.	Usefulness-Score aller topologischen Indizes für die Graphenklasse Random	52
5.	Ausgabe der Scores der Indizes mit der höchsten Usefulness innerhalb der Graphenklassen. Bei allen Random-Graphen besitzt der Szeged-Index den höchsten Einfluss über alle Hauptkomponenten. Bei den Pfad-Graphen haben der Szeged- und Wiener-Index denselben Usefulness-Score.	53
6.	Verbesserung der Genauigkeit des Modells während des Trainings	56
7.	Erstellen eines einfachen Graphen mit NetworkX	59
8.	Code zum Einlesen des Graphen	61
9.	Der Code für das GCN-Modell	62
10.	Code zur Klassifikation des Graphen	63
11.	Code für den Vorschlag der topologischen Indizes	64
12.	Ausgabe aller Komponenten der vier Hauptkomponenten der verschiedenen Netzwerk-Klassen	84
13.	Ausgabe aller erklärbaren Varianzen und Usefulness-Scores aller Indizes	87

Literaturverzeichnis

- [1] V. P. Kozyrev, "Graph theory," *Journal of Soviet Mathematics*, vol. 2, no. 5, pp. 489–519, 1974.
- [2] A. Bavelas, L. Beauguitte, and M. Maisonneuve, "Alex bavelas, 1948, a mathematical model for group structures. version bilingue et commentée." Publisher: HAL.
- [3] L. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, pp. 35–41, 1977.
- [4] H. Wiener, "Structural determination of paraffin boiling points," *Journal of the American Chemical Society*, vol. 69, no. 1, pp. 17–20, 1947. Publisher: American Chemical Society.
- [5] M. Randic, "Characterization of molecular branching," *Journal of the American Chemical Society*, vol. 97, no. 23, pp. 6609–6615, 1975. Publisher: American Chemical Society.
- [6] A. Balaban, I. Motoc, D. Bonchev, and O. Mekenyan, "1983 topological indices for structure-activity correlations." Journal Abbreviation: Topics in Current Chemistry Publication Title: Topics in Current Chemistry Volume: 114.
- [7] A. T. Balaban, "Highly discriminating distance-based topological index," *Chemical Physics Letters*, vol. 89, no. 5, pp. 399–404, 1982.
- [8] V. Kraus, M. Dehmer, and F. Emmert-Streib, "Probabilistic inequalities for evaluating structural network measures," *Information Sciences*, vol. 288, pp. 220–245, 2014.
- [9] C. Gomes, "Combination of topological indices in network analysis: A computational approach," 2019. Systems Biology, Modelling, Network Analysis, Dynamical Analysis, Machine Learning.
- [10] Y. Ma, M. Dehmer, U.-M. Künzi, S. Tripathi, M. Ghorbani, J. Tao, and F. Emmert-Streib, "The usefulness of topological indices," *Information Sciences*, vol. 606, pp. 143–151, 2022.
- [11] S. C. Basak, S. Nikolić, N. Trinajstić, D. Amić, and D. Bešlo, "QSPR modeling: Graph connectivity indices versus line graph connectivity indices," *Journal of Chemical Information and Computer Sciences*, vol. 40, no. 4, pp. 927–933, 2000.

- [12] C. Raychaudhury, S. K. Ray, J. J. Ghosh, A. B. Roy, and S. C. Basak, "Discrimination of isomeric structures using information theoretic topological indices," *Journal of Computational Chemistry*, vol. 5, no. 6, pp. 581–588, 1984.
- [13] M. Randic, "Croatica chemica acta, vol. 64 no. 1, 1991.."
- [14] M. Fiszman, D. Demner-Fushman, H. Kilicoglu, and T. C. Rindflesch, "Automatic summarization of MEDLINE citations for evidence-based medical treatment: A topic-oriented evaluation," *Journal of Biomedical Informatics*, vol. 42, pp. 801–813, Oct. 2009.
- [15] D. Bonchev and O. Polansky, "On the Topological Complexity of Chemical Systems," Jan. 1987.
- [16] B. Furtula, I. Gutman, and M. Dehmer, "On structure-sensitivity of degree-based topological indices," *Applied Mathematics and Computation*, vol. 219, no. 17, pp. 8973–8978, 2013.
- [17] M. Dehmer, M. Grabner, and K. Varmuza, "Information indices with high discriminative power for graphs," *PLoS ONE*, vol. 7, no. 2, p. e31214, 2012.
- [18] B. Bollobás, *Modern Graph Theory*, vol. 184 of *Graduate Texts in Mathematics*. New York, NY: Springer, 1998.
- [19] R. Todeschini and V. Consonni, *Handbook of Molecular Descriptors*. Methods and Principles in Medicinal Chemistry, Wiley, 1 ed., Sept. 2000.
- [20] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998. Bandiera_abtest: a Cg_type: Nature Research Journals Number: 6684 Primary_atype: Research Publisher: Nature Publishing Group.
- [21] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, Dec. 1959.
- [22] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. USA: W. H. Freeman & Co., 1990.
- [23] R. Albert, H. Jeong, and A.-L. Barabási, "Diameter of the world-wide web," *Nature*, vol. 401, no. 6749, pp. 130–131, 1999. Number: 6749 Publisher: Nature Publishing Group.
- [24] J. A. Barnes, "Graph Theory and Social Networks: A Technical Comment on Connectedness and Connectivity," *Sociology*, vol. 3, pp. 215–232, May 1969. Publisher: SAGE Publications Ltd.
- [25] A.-L. Barabasi, *Network Science*. Cambridge University Press, 1st edition ed., 2016.
- [26] F. Harary, *Graph Theory*. CRC Press, 1994.

- [27] F. Emmert-Streib, S. Moutari, and M. Dehmer, *Mathematical Foundations of Data Science Using R*. De Gruyter, 2020.
- [28] A. Brandstädt, V. B. Le, and J. P. Spinrad, *Graph Classes: A Survey*. Discrete Mathematics and Applications, Society for Industrial and Applied Mathematics, Jan. 1999.
- [29] J. Bang-Jensen and G. Gutin, “Basic Terminology, Notation and Results,” in *Classes of Directed Graphs* (J. Bang-Jensen and G. Gutin, eds.), Springer Monographs in Mathematics, pp. 1–34, Cham: Springer International Publishing, 2018.
- [30] G. Grimmett, “Random Processes on Graphs and Lattices,” *Cambridge University Press*, 2010.
- [31] B. Bollobás and B. Béla, *Random Graphs*. Cambridge University Press, Aug. 2001. Google-Books-ID: o9WecWgilzYC.
- [32] M. Newman, “Networks - An Introduction,” 2010.
- [33] J. Travers and S. Milgram, “An Experimental Study of the Small World Problem,” *Sociometry*, vol. 32, no. 4, pp. 425–443, 1969. Publisher: [American Sociological Association, Sage Publications, Inc.].
- [34] A. Gayen, “Small World Model - Using Python Networkx,” Jan. 2020. Section: Python.
- [35] A.-L. Barabási and R. Albert, “Emergence of Scaling in Random Networks,” *Science*, vol. 286, pp. 509–512, Oct. 1999. Publisher: American Association for the Advancement of Science.
- [36] M. E. J. Newman, “The Structure and Function of Complex Networks,” 2003.
- [37] P. Cayley, “On the Analytical Forms Called Trees,” *American Journal of Mathematics*, vol. 4, no. 1, pp. 266–268, 1881. Publisher: Johns Hopkins University Press.
- [38] E. A. Bender and S. Gill Williamson, *Lists, Decisions and Graphs - With an Introduction to Probability*. FreeTechBooks, Dec. 2010.
- [39] A. T. Balaban, “Topological indices based on topological distances in molecular graphs,” *Pure and Applied Chemistry*, vol. 55, pp. 199–206, Jan. 1983. Publisher: De Gruyter.
- [40] S. Manzoor, M. K. Siddiqui, and S. Ahmad, “On entropy measures of molecular graphs using topological indices,” *Arabian Journal of Chemistry*, vol. 13, no. 8, pp. 6285–6298, 2020.
- [41] D. H. Rouvray, “The modeling of chemical phenomena using topological indices,” *Journal of Computational Chemistry*, vol. 8, no. 4, pp. 470–480, 1987. _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.540080427>.
- [42] D. Plavšić, S. Nikolić, N. Trinajstić, and Z. Mihalić, “On the harary index for the characterization of chemical graphs,” *Journal of Mathematical Chemistry*, vol. 12, no. 1, pp. 235–250, 1993.

- [43] J. Devillers and A. T. Balaban, *Topological Indices and Related Descriptors in QSAR and QSPR*. CRC Press, 2000. Google-Books-ID: H2S1DwAAQBAJ.
- [44] H. González-Díaz, S. Vilar, L. Santana, and E. Uriarte, “Medicinal chemistry and bioinformatics—current trends in drugs discovery with networks topological indices,” *Current Topics in Medicinal Chemistry*, vol. 7, no. 10, pp. 1015–1029, 2007.
- [45] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, June 2013.
- [46] I. Gutman, “Degree-Based Topological Indices,” *Croatica Chemica Acta*, vol. 86, no. 4, pp. 351–361, 2013.
- [47] X. Li and Y. Shi, “A survey on the Randic index,” *Match (Mulheim an der Ruhr, Germany)*, vol. 59, pp. 127–156, Jan. 2008.
- [48] K. Xu, “The zagreb indices of graphs with a given clique number,” *Applied Mathematics Letters*, vol. 24, no. 6, pp. 1026–1030, 2011.
- [49] K. C. Das, K. Xu, and J. Nam, “Zagreb indices of graphs,” *Frontiers of Mathematics in China*, vol. 10, no. 3, pp. 567–582, 2015.
- [50] L. Zhong, “The harmonic index for graphs,” *Applied Mathematics Letters*, vol. 25, no. 3, pp. 561–566, 2012.
- [51] J. Li and W. C. Shiu, “The harmonic index of a graph,” *Rocky Mountain Journal of Mathematics*, vol. 44, pp. 1607–1620, 2014.
- [52] E. Estrada, L. Torres, L. Rodriguez, and I. Gutman, “An atom-bond connectivity index: modelling the enthalpy of formation of alkanes,” *Journal of Chemical Sciences*, 1998. Publisher: NISCAIR-CSIR, India.
- [53] G. Fath-Tabar and A. Ashrafi, “The Hyper-Wiener Polynomial of Graphs,” *Iranian Journal of Mathematical Sciences and Informatics*, vol. 6, Nov. 2011.
- [54] M. Cavaleri, D. D’Angeli, and A. Donno, “A group representation approach to balance of gain graphs,” *Journal of Algebraic Combinatorics*, vol. 54, pp. 265–293, Aug. 2021. arXiv:2001.08490 [math].
- [55] H. Hosoya, “Topological Index as a Sorting Device for Coding Chemical Structures,” *Journal of Chemical Documentation*, vol. 12, pp. 181–183, Aug. 1972. Publisher: American Chemical Society.
- [56] I. Gutman and A. Dobrynin, “The szeged index - a success story,” *Graph Theory Notes New York*, vol. 34, pp. 37–44, 1998.

- [57] J. L. Gross, J. Yellen, and M. Anderson, *Graph Theory and Its Applications*. Chapman and Hall/CRC, 3 ed., 2018.
- [58] A. T. Balaban, “Chemical graphs,” *Theoretica chimica acta*, vol. 53, no. 4, pp. 355–375, 1979.
- [59] E. Estrada and J. A. Rodríguez-Velázquez, “Subgraph centrality in complex networks,” *Physical Review E*, vol. 71, no. 5, p. 056103, 2005. Publisher: American Physical Society.
- [60] J. A. de la Peña, I. Gutman, and J. Rada, “Estimating the estrada index,” *Linear Algebra and its Applications*, vol. 427, no. 1, pp. 70–76, 2007.
- [61] M. Dehmer, “Information processing in complex networks: Graph entropy and information functionals,” *Applied Mathematics and Computation*, vol. 201, no. 1, pp. 82–94, 2008.
- [62] M. Dehmer, “Information theory of networks,” *Symmetry*, vol. 3, no. 4, pp. 767–779, 2011.
- [63] J. Iceland, “The Multigroup Entropy Index (Also Known as Theil’s H or the Information Theory Index),” *Retrieved July*, vol. 31, Jan. 2004.
- [64] D. Bonchev, “Information theoretic indices for characterization of chemical structures. by danail bonchev wiley, new york, 1983,” *International Journal of Quantum Chemistry*, vol. 27, no. 1, pp. 103–103, 1985. _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/qua.560270109>.
- [65] A. Mowshowitz, “Entropy and the complexity of graphs: IV. entropy measures and graphical structure,” *The bulletin of mathematical biophysics*, vol. 30, no. 4, pp. 533–546, 1968.
- [66] L. A. J. Mueller, M. Dehmer, and F. Emmert-Streib, “Comparing biological networks: A survey on graph classifying techniques,” in *Systems Biology* (A. Prokop and B. Csukás, eds.), pp. 43–63, Springer Netherlands, 2013.
- [67] B. Zelinka, “On a certain distance between isomorphism classes of graphs,” *Časopis pro pěstování matematiky*, vol. 100, no. 4, pp. 371–373, 1975. Publisher: Mathematical Institute of the Czechoslovak Academy of Sciences.
- [68] F. Sobik, “Graphmetriken und klassifikation strukturierter objekte,” *ZKI-Informationen, Akad. Wiss. DDR*, vol. 2, no. 82, pp. 63–122, 1982.
- [69] B. D. McKay and A. Piperno, “Practical graph isomorphism, II,” *Journal of Symbolic Computation*, vol. 60, pp. 94–112, Jan. 2014.
- [70] S. C. Basak, V. R. Magnuson, G. J. Niemi, R. R. Regal, and G. D. Veith, “Topological indices: their nature, mutual relatedness, and applications,” *Mathematical Modelling*, vol. 8, pp. 300–305, 1987.

- [71] M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, M. Ares, and D. Haussler, “Knowledge-based analysis of microarray gene expression data by using support vector machines,” *Proceedings of the National Academy of Sciences*, vol. 97, no. 1, pp. 262–267, 2000. Publisher: Proceedings of the National Academy of Sciences.
- [72] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene selection for cancer classification using support vector machines,” *Machine Learning*, vol. 46, no. 1, pp. 389–422, 2002.
- [73] H. Kashima and A. Inokuchi, “Kernels for graph classification,” in *Kernels for graph classification*, 2002.
- [74] K. Borgwardt and H. Kriegel, “Shortest-path kernels on graphs,” in *Fifth IEEE International Conference on Data Mining (ICDM’05)*, pp. 74–81, IEEE, 2005.
- [75] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, “An End-to-End Deep Learning Architecture for Graph Classification,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, Apr. 2018.
- [76] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, “Deep Graph Infomax,” Dec. 2018. arXiv:1809.10341 [cs, math, stat].
- [77] G. Chen, P. Chen, C.-Y. Hsieh, C.-K. Lee, B. Liao, R. Liao, W. Liu, J. Qiu, Q. Sun, J. Tang, R. Zemel, and S. Zhang, “Alchemy: A Quantum Chemistry Dataset for Benchmarking AI Models,” June 2019. arXiv:1906.09427 [cs, stat].
- [78] W. L. Hamilton, “Graph Representation Learning,” *Mc Gill University*, 2020.
- [79] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu, “Do Transformers Really Perform Bad for Graph Representation?,” Nov. 2021. arXiv:2106.05234 [cs].
- [80] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention Is All You Need,” Dec. 2017. arXiv:1706.03762 [cs].
- [81] V. P. Dwivedi and X. Bresson, “A Generalization of Transformer Networks to Graphs,” Jan. 2021. arXiv:2012.09699 [cs].
- [82] Huggingface, “Papers with Code - Machine Learning Datasets,” 2017.
- [83] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, “Open Graph Benchmark: Datasets for Machine Learning on Graphs,” Feb. 2021. arXiv:2005.00687 [cs, stat].
- [84] D. Q. Nguyen, T. D. Nguyen, and D. Phung, “Universal Graph Transformer Self-Attention Networks,” Mar. 2022. arXiv:1909.11855 [cs, stat] version: 9.
- [85] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks.”

- [86] N. Shervashidze, S. V. N. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt, “Efficient graphlet kernels for large graph comparison,” in *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, pp. 488–495, PMLR, Apr. 2009. ISSN: 1938-7228.
- [87] N. Shervashidze, “Weisfeiler-Lehman Graph Kernels,” *Journal of Machine Learning Research*, 2011.
- [88] W. L. Hamilton, R. Ying, and J. Leskovec, “Inductive Representation Learning on Large Graphs,” Sept. 2018. arXiv:1706.02216 [cs, stat].
- [89] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, “How Powerful are Graph Neural Networks?”, Feb. 2019. arXiv:1810.00826 [cs, stat].
- [90] V. Ravi and K. Desikan, “On computation of the reduced reverse degree and neighbourhood degree sum-based topological indices for metal-organic frameworks,” *Main Group Metal Chemistry*, vol. 45, pp. 92–99, Jan. 2022. Publisher: De Gruyter.
- [91] S. Amin, M. A. Rehman, A. Naseem, I. Khan, N. Alshammary, and N. N. Hamadneh, “Analysis of complex networks via some novel topological indices,” *Mathematical Problems in Engineering*, vol. 2022, p. e9040532, 2022. Publisher: Hindawi.
- [92] P. Sarkar, N. De, and A. Pal, “On Some Neighbourhood Degree-Based Multiplicative Topological Indices and Their Applications,” *Polycyclic Aromatic Compounds*, vol. 42, pp. 1–16, Dec. 2021.
- [93] A. Hagberg, P. Swart, and D. S Chult, “Exploring network structure, dynamics, and function using networkx.”
- [94] D. Amos and R. Davila, “GrinPy — Documentation,” Nov. 2022.
- [95] J. D. Hunter, “Matplotlib: A 2d graphics environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
- [96] C. Spearman, “The proof and measurement of association between two things,” *The American Journal of Psychology*, vol. 15, no. 1, pp. 72–101, 1904. Publisher: University of Illinois Press.
- [97] M. Bärtl, *Statistik Schritt für Schritt: Das Lehrbuch vom Autor des YouTube-Kanals Kurzes Tutorium Statistik*. Independently published, 2017.
- [98] I. T. Jolliffe, “Principal Component Analysis and Factor Analysis,” in *Principal Component Analysis*, pp. 115–128, New York, NY: Springer New York, 1986. Series Title: Springer Series in Statistics.

- [99] N. Halko, P. G. Martinsson, and J. A. Tropp, “Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions,” *SIAM Review*, vol. 53, pp. 217–288, Jan. 2011. Publisher: Society for Industrial and Applied Mathematics.
- [100] H. Abdi and L. J. Williams, “Principal component analysis: Principal component analysis,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.
- [101] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.
- [102] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” Jan. 2017. arXiv:1412.6980 [cs].
- [103] L. Biewald, “Experiment tracking with weights and biases,” 2020. Software available from wandb.com.
- [104] M. Fey and J. E. Lenssen, “Fast graph representation learning with PyTorch Geometric,” in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.
- [105] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.
- [106] Wes McKinney, “Data Structures for Statistical Computing in Python,” in *Proceedings of the 9th Python in Science Conference* (Stéfan van der Walt and Jarrod Millman, eds.), pp. 56 – 61, 2010.
- [107] M. V. Diudea, A. Ilić, K. Varmuza, and M. Dehmer, “Network analysis using a novel highly discriminating topological index,” *Complexity*, vol. 16, no. 6, pp. 32–39, 2011.
- [108] P. E. Black, “Bparse graph” in Dictionary of Algorithms and Data Structures [online], Paul E. Black, ed., 2 December 2019. Available from: <https://www.nist.gov/dads/HTML/sparsegraph.html> (accessed 04 January 2023).
- [109] D. M. Scott, D. C. Novak, L. Aultman-Hall, and F. Guo, “Network Robustness Index: A new method for identifying critical links and evaluating the performance of transportation networks,” *Journal of Transport Geography*, vol. 14, pp. 215–227, May 2006.

Selbstständigkeitserklärung

Ich erkläre hiermit, dass ich diese Thesis selbstständig verfasst und keine anderen als die angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäss aus Quellen entnommen wurden, habe ich als solche kenntlich gemacht. Ich versichere zudem, dass ich bisher noch keine wissenschaftliche Arbeit mit gleichem oder ähnlichem Inhalt an der Fernfachhochschule Schweiz oder an einer anderen Hochschule eingereicht habe. Mir ist bekannt, dass andernfalls die Fernfachhochschule Schweiz zum Entzug des aufgrund dieser Thesis verliehenen Titels berechtigt ist.

Zürich, 15. März 2023



Ort, Datum, Unterschrift