



Master Thesis

im Rahmen des
Universitätslehrganges „Geographical Information Science & Systems“
(UNIGIS MSc) am Interfakultären Fachbereich für GeoInformatik (Z_GIS)
der Paris Lodron-Universität Salzburg

zum Thema

„Erreichbarkeit von Kliniken bei Tracer- Diagnose unter Anwendung eines Geschwindigkeitsprofils für Rettungswägen“

vorgelegt von

Annette Gattinger
105494, UNIGIS MSc Jahrgang 2019

Betreuer:
Dr. Christian Neuwirth

Zur Erlangung des Grades
„Master of Science – MSc“

München, 14.04.2023

Zusammenfassung

Die Masterarbeit befasst sich mit der Versorgung der Bevölkerung bei Verdacht auf Tracer-Diagnose, bemessen über die Erreichbarkeit von geeigneten Krankenhäusern. Tracer-Diagnosen sind gekennzeichnet durch den Schweregrad des Krankbildes und der Häufigkeit des Auftretens. Zudem sind es zeitkritische Erkrankungen, die eine schnelle und adäquate Behandlung in einem Krankenhaus bedürfen. Hierbei hat sich der Begriff der „golden hour“ etabliert. Vom Eingang der Notrufmeldung bis zur Behandlung in einer Notaufnahme darf nicht länger als eine Stunde vergehen, um die Wahrscheinlichkeit zu verringern, dass der Patient stirbt, bzw. starke körperliche Schädigungen zurückbleiben.

Leitziel der Masterthesis ist die Aufdeckung von regionalen Versorgungs- und Erreichbarkeitsdefiziten von medizinischen Einrichtungen in Bayern. Dabei sind unter medizinischen Einrichtungen Akutkliniken zu verstehen, die die Voraussetzungen zur Behandlung von Patienten mit Verdacht auf Tracer-Diagnosen erfüllen.

Um das Leitziel zu erreichen, wird die Zugänglichkeit von Krankenhäusern für die Bevölkerung betrachtet. Dabei sind Einzugsbereiche von Kliniken zu berechnen, wofür zunächst ein Geschwindigkeitsprofil für Rettungswägen ermittelt wird. Die abgeschätzten Geschwindigkeiten basieren auf realen Einsatzdaten von Rettungswägen dokumentiert von den Integrierten Leitstellen in Bayern. Dabei wird der aus OpenStreetMap extrahierte Straßengraph in detailliertere Straßenkategorien differenziert und über eine lineare Regression für jede Kategorie die Durchschnittsgeschwindigkeit abgeschätzt. Im Anschluss daran kommt das Geschwindigkeitsprofil zur Anwendung in Form einer Erreichbarkeitsanalyse über die Methode der Enhanced 2-Step Floating Catchment Area. Dabei werden Erreichbarkeitszonen gebildet, die die Bedeutung von Krankenhäusern als Zielklinik mit zunehmender Entfernung zum Einsatzort geringer gewichten. Der daraus resultierende Spatial Accessibility Index lässt Schlüsse zur Bewertung regionaler Erreichbarkeitsdefizite zu.

Besonders bei schwerem Schädel-Hirn-Trauma zeigen sich die größten zusammenhängenden Gebiete mit Defiziten in der Zugänglichkeit von geeigneten Krankenhäusern. Für die meisten Tracer-Diagnosen sind Regionen im Norden Schwabens und im Bayerischen Wald auffällig, auch im Main-Spessart in Unterfranken zeigen sich strukturelle Defizite.

Abstract

The master thesis deals with the health care of the population in case of suspected tracer diagnosis, measured by the accessibility of appropriate hospitals. Tracer diagnoses are characterized by the severity of the clinical picture and the frequency of occurrence. In addition, they are time-critical illnesses that require rapid and adequate treatment in a hospital. The term "golden hour" has become established in this context. No more than one hour may elapse between receipt of the emergency call and treatment in an emergency room in order to reduce the likelihood of the patient dying or being left with severe physical damage.

The aim of the master's thesis is to identify regional deficits in the supply and accessibility of medical facilities in Bavaria. In this context, medical facilities are to be understood as acute care hospitals that fulfill the requirements for the treatment of patients with suspected tracer diagnoses.

In order to achieve the key objective, the accessibility of hospitals for the population is considered. For this purpose, catchment areas of hospitals are to be calculated, for which a speed profile for ambulances is determined. The estimated speeds are based on real data of ambulances documented by the integrated control centers in Bavaria. The street graph extracted from OpenStreetMap is differentiated into more detailed street categories and a linear regression is used to estimate the average speed for each category. Subsequently, the velocity profile is applied in the form of an accessibility analysis via the Enhanced 2-Step Floating Catchment Area method. In this process, accessibility zones are formed that give less weight to the importance of hospitals as destination hospitals as the distance to the deployment site increases. The resulting Spatial Accessibility Index allows conclusions to be drawn on the evaluation of regional accessibility deficits.

Especially for severe traumatic brain injury, the largest contiguous areas show deficits in the accessibility of suitable hospitals. For most tracer diagnoses, regions in the north of Swabia and in the Bavarian Forest are conspicuous; structural deficits are also evident in Main-Spessart in Lower Franconia.

Inhaltsverzeichnis

Zusammenfassung -----	I
Abstract -----	II
Inhaltsverzeichnis -----	III
Abbildungsverzeichnis -----	V
Tabellenverzeichnis -----	VII
Abkürzungsverzeichnis -----	VIII
1 Einleitung -----	1
1.1 Stand der Forschung-----	1
1.2 Ziele der Arbeit-----	2
1.3 Aufbau und Struktur der Arbeit-----	3
2 Datengrundlagen -----	4
2.1 Standorte relevanter Krankenhäuser-----	4
2.1.1 Krankenhauslandschaft zur Behandlung von Tracer-Diagnosen-----	4
2.1.2 Beschreibung des Krankenhausdatensatzes-----	10
2.2 Einsatzdaten Integrierter Leitstellen in Bayern-----	12
2.2.1 Datenerfassung der Integrierten Leitstellen Bayern-----	12
2.2.2 Beschreibung der Einsatzdaten-----	14
2.3 Digitales Straßennetz-----	16
2.3.1 Digitales Straßennetzes aus OpenStreetMap-Daten-----	16
2.3.2 Beschreibung des Straßendatensatzes-----	20
2.4 Weitere Daten-----	27
2.4.1 Administrative Grenzen-----	27
2.4.2 Bevölkerungsdaten-----	29
3 Entwicklung eines Geschwindigkeitsprofils für Rettungswägen -----	31
3.1 Routing-Basis-----	31
3.1.1 Routing-Theorie-----	31
3.1.2 Auswahl der Routing-Engine-----	37
3.2 Operationalisierung des Straßengraphen-----	44
3.2.1 Implementierung des OSRM-Backends-----	44
3.2.2 Zuweisung der relevanten Straßenkategorien-----	47
3.2.3 Ermittlung der Einsatz-Fahrstrecken-----	49

3.2.4	Berechnung eines Geschwindigkeitsprofils für RTW -----	54
3.2.5	Validierung des Geschwindigkeitsprofils für RTW-----	57
3.2.6	Implementierung des Geschwindigkeitsprofils für RTW-----	60
4	Erreichbarkeitsanalyse der Krankenhäuser -----	63
4.1	Erreichbarkeitsanalyse-----	63
4.1.1	Erreichbarkeitsmodell -----	63
4.1.2	Eingangsparameter des Erreichbarkeitsmodells E2SFCA-----	66
4.1.3	Berechnung der Krankenhäuserreichbarkeit nach E2SFCA -----	68
4.2	Auswertung der Krankenhäuserreichbarkeit -----	70
4.2.1	Ergebnisse zur Versorgung der Bevölkerung nach Tracer-Diagnose-----	70
4.2.2	Vergleich der Ergebnisse-----	101
5	Diskussion -----	105
5.1	Beantwortung des Forschungsziels -----	105
5.2	Bewertung der Ergebnisse -----	109
5.3	Eignung der gewählten Methoden-----	110
6	Ausblick -----	112
	Literaturverzeichnis -----	113
	Anhang -----	A
	Anhang I: car_con.lua -----	A
	Anhang II: osm2sqlite.py -----	F
	Anhang III: calc_routes.py-----	J
	Anhang IV: leastsquares.py -----	L
	Anhang V: RTW_con.lua -----	M
	Anhang VI: calc_fz.py-----	Q
	Anhang VII: reg_koeff.py -----	R
	Anhang VIII: DM_GEM_KH.py-----	R
	Anhang IX: Karte Landesentwicklungsprogramm-----	T

Abbildungsverzeichnis

Abb. 1: Gliederung der zugelassenen Kliniken-----	4
Abb. 2: Übersicht der relevanten Zeitintervalle (eigene Darstellung nach INM und Eckpunktepapier 2016)-----	6
Abb. 3: Überblick der Krankenhausverteilung nach Tracer-Diagnose -----	9
Abb. 4: Registrierung der Geometrie in der Systemtabelle geometry_columns -----	10
Abb. 5: Prozess der Datenverarbeitung im INM -----	13
Abb. 6: Registrierung der Geometrie in der Systemtabelle geometry_columns -----	14
Abb. 7: Vereinfachtes konzeptionelles OSM-Datenmodell (Minghini und Frassinelli 2019; Ramm und Topf 2010) -----	17
Abb. 8: OSM-Kartenausschnitt mit Beispielen an Key-Value-Paaren auf Map-Features -----	19
Abb. 9: Konnektivität am Beispiel vereinfachter Autobahnausfahrten -----	19
Abb. 10: Ausschnitt der administrativen Grenzen und Referenzpunkte -----	29
Abb. 11: Verteilung der Bevölkerung auf Gemeindeebene -----	30
Abb. 12: Schematischer Ablauf des Dijkstra-Algorithmus -----	33
Abb. 13: Schematischer Ablauf des A*-Algorithmus -----	35
Abb. 14: Schematisches Konzept der Contraction Hierarchies -----	37
Abb. 15: nicht erreichbare Einsatzorte in Fußgängerzone-----	53
Abb. 16: nicht erreichbare Einsatzorte in Obstplantage -----	53
Abb. 17: linearer Zusammenhang der dokumentierten und der gerouteten Fahrzeiten vor (links) und nach (rechts) der Datenbereinigung -----	55
Abb. 18: NNLS-Geschwindigkeitsmodellierung von zehn Stichproben -----	57
Abb. 19: Schematische Darstellung der 2-Step Floating Catchment Area (2SFCA) Methode -----	64
Abb. 20: Schematische Darstellung der Enhanced 2-Step Floating Catchment Area (E2SFCA) Methode -----	65
Abb. 21: Schematische Darstellung der Berechnung der Krankenhauserreichbarkeiten -----	66
Abb. 22: Spatial Accessibility Index (SPAI) bei Reanimation-----	71
Abb. 23: Verteilung der Bevölkerung nach Erreichbarkeit der nächstgelegenen Klinik bei Reanimation -----	72
Abb. 24: Fahrzeiten von Gemeinden in die nächste geeignete Klinik bei Reanimation -----	73
Abb. 25: Bevölkerung nach Entfernung zum nächstgelegenen geeigneten Krankenhaus bei Reanimation-----	74
Abb. 26: Verteilung der Fahrzeiten zum nächstgelegenen Krankenhaus bei Reanimation-----	75
Abb. 27: Spatial Accessibility Index (SPAI) bei Schlaganfall-----	76
Abb. 28: Verteilung der Bevölkerung nach Erreichbarkeit der nächstgelegenen Klinik bei Schlaganfall -----	77
Abb. 29: Fahrzeiten von Gemeinden in die nächste geeignete Klinik bei Schlaganfall -----	78
Abb. 30: Bevölkerung nach Entfernung zum nächstgelegenen geeigneten Krankenhaus bei Schlaganfall-----	79
Abb. 31: Verteilung der Fahrzeiten zum nächstgelegenen Krankenhaus bei Schlaganfall-----	79
Abb. 32: Spatial Accessibility Index (SPAI) bei schwerem Schädel-Hirn-Trauma -----	81
Abb. 33: Verteilung der Bevölkerung nach Erreichbarkeit der nächstgelegenen Klinik bei schwerem Schädel-Hirn-Trauma -----	82
Abb. 34: Fahrzeiten von Gemeinden in die nächste geeignete Klinik bei schwerem Schädel-Hirn-Trauma -----	83

Abb. 35: Bevölkerung nach Entfernung zum nächstgelegenen geeigneten Krankenhaus bei schwerem Schädel-Hirn-Trauma -----	84
Abb. 36: Verteilung der Fahrzeiten zum nächstgelegenen Krankenhaus bei schwerem Schädel-Hirn-Trauma -----	85
Abb. 37: Spatial Accessibility Index (SPAI) bei Polytrauma -----	86
Abb. 38: Verteilung der Bevölkerung nach Erreichbarkeit der nächstgelegenen Klinik bei Polytrauma -----	87
Abb. 39: Fahrzeiten von Gemeinden in die nächste geeignete Klinik bei Polytrauma -----	88
Abb. 40: Bevölkerung nach Entfernung zum nächstgelegenen geeigneten Krankenhaus bei Polytrauma -----	89
Abb. 41: Verteilung der Fahrzeiten zum nächstgelegenen Krankenhaus bei Polytrauma -----	90
Abb. 42: Spatial Accessibility Index (SPAI) bei Sepsis -----	91
Abb. 43: Verteilung der Bevölkerung nach Erreichbarkeit der nächstgelegenen Klinik bei Sepsis -----	92
Abb. 44: Fahrzeiten von Gemeinden in die nächste geeignete Klinik bei Sepsis -----	93
Abb. 45: Bevölkerung nach Entfernung zum nächstgelegenen geeigneten Krankenhaus bei Sepsis --	94
Abb. 46: Verteilung der Fahrzeiten zum nächstgelegenen Krankenhaus bei Sepsis -----	95
Abb. 47: Spatial Accessibility Index (SPAI) bei ST-Hebungsinfarkt -----	96
Abb. 48: Verteilung der Bevölkerung nach Erreichbarkeit der nächstgelegenen Klinik bei ST-Hebungsinfarkt -----	97
Abb. 49: Fahrzeiten von Gemeinden in die nächste geeignete Klinik bei ST-Hebungsinfarkt -----	98
Abb. 50: Bevölkerung nach Entfernung zum nächstgelegenen geeigneten Krankenhaus bei ST-Hebungsinfarkt -----	99
Abb. 51: Verteilung der Fahrzeiten zum nächstgelegenen Krankenhaus bei ST-Hebungsinfarkt -----	100
Abb. 52: Anzahl der Einwohner (Gesamtzahl 13.350.608) mit Erreichbarkeit von Kliniken innerhalb und über 30 Minuten nach Tracer-Eignung -----	101
Abb. 53: Erreichbarkeit von Kliniken in Zone 0 - 10 Minuten nach Tracer-Eignung -----	101
Abb. 54: Erreichbarkeit von Kliniken in Zone 10 - 20 Minuten nach Tracer-Eignung -----	102
Abb. 55: Erreichbarkeit von Kliniken in Zone 20 - 30 Minuten nach Tracer-Eignung -----	102
Abb. 56: Fahrzeit in die nächstgelegene Klinik nach Bevölkerungsperzentilen -----	103
Abb. 57: Summe des Spatial Accessibility Index über alle Tracer-Diagnosen -----	106
Abb. 58: Spatial Accessibility Index je Tracer-Diagnose -----	107
Abb. 59 Raumtypen nach bayrischem Landesentwicklungsprogramm -----	T

Tabellenverzeichnis

Tab. 1: Verteilung der Kliniken nach Regierungsbezirk (StMGP 2022, 267f)	5
Tab. 2: Kriterien-Matrix zur Versorgung von Tracer-Diagnosen bei Erwachsenen (eigene Darstellung nach Empfehlung 02/5-2019)	8
Tab. 3: Anzahl der Krankenhäuser nach Raumtyp in Bayern	8
Tab. 4: Datentypen der Krankenhaustabelle (Stand: 19.12.2022 12:00 Uhr)	11
Tab. 5: Datentypen der Einsatzdatentabelle (Beobachtungszeitraum: 01.01.2021 - 31.12.2021)	15
Tab. 6: Anzahl der OSM-Elemente in Bayern nach Regierungsbezirk zum Stand: 29.01.2023	21
Tab. 7: Ausprägungen des Wege-Elements im OSM-Datensatz für Bayern zum Stand 29.01.2023 (OpenStreetMap Wiki contributors 2023)	22
Tab. 8: Identifikation der befahrbaren Straßenkategorien (OpenStreetMap Wiki contributors 2023)	24
Tab. 9: Höchstgeschwindigkeiten aus dem Tag zone:traffic	25
Tab. 10: Höchstgeschwindigkeiten aus dem Tag zone:maxspeed	25
Tab. 11: Höchstgeschwindigkeiten aus dem Tag source:maxspeed	26
Tab. 12: Struktur und Dateninhalte der ALKIS® Verwaltungsgebiete	28
Tab. 13: Datentypen der Bevölkerungstabelle BASIS_BEV	29
Tab. 14: Eigenschaften der Routing-Engines nach (GIS-OPS UG 2020)	41
Tab. 15: Kategorienbildung aus den OSM-Wege-Elementen	47
Tab. 16: Datentypen der Tabelle EDGES	49
Tab. 17: statistische Kennwerte zur Beschreibung der gerouteten Einsatzfahrten nach Straßenkategorie	51
Tab. 18: statistische Kennwerte der Fahrzeit-Variablen und der Geschwindigkeit der Einsatzfahrten vor der Datenbereinigung	52
Tab. 19: statistische Kennwerte der Fahrzeit-Variablen der Einsatzfahrten nach der Datenbereinigung	54
Tab. 20: Gegenüberstellung der Ergebnisse der linearen Regression vor und nach der Datenbereinigung	55
Tab. 21: geschätzte Geschwindigkeiten je Straßenkategorie	56
Tab. 22: NNLS-Geschwindigkeitsmodellierung von zehn Stichproben - Einzelergebnisse in km/h	58
Tab. 23: Datentypen der Tabelle OUT_BY_FZ_V_RAW	61
Tab. 24: Gegenüberstellung unterschiedlicher Erreichbarkeitsmodelle (Li und Wang 2022)	63
Tab. 25: Referenzpunkte der regionalen Einheit	67
Tab. 26: Anzahl der Relationen je Tracer-Diagnose bei 2.229 Gemeinden	67
Tab. 27: Distanzgewichte der Erreichbarkeitszonen	68
Tab. 28: Beispieldatensätze für Krankenhäuser mit berechnetem R-Wert	69
Tab. 29: Beispieldatensätze mit berechnetem Spatial Accessibility Index nach Tracer-Diagnose	70
Tab. 30: Kumulierter Bevölkerungsanteil nach Fahrzeitzone zur nächstgelegenen Klinik	103
Tab. 31: Erreichbarkeit von Kliniken in 30 Minuten Fahrzeit	104
Tab. 32: Gegenüberstellung von Kennwerten für Stadt und Land	105

Abkürzungsverzeichnis

2SFCA	Two-Step Floating Catchment Area
AGS	Amtlicher Gemeindeschlüssel
API	Application Programming Interface
APSP	All-Pairs-Shortest-Path
BayStMI	Bayrisches Staatsministerium des Innern für Bau und Verkehr
CPU	Chest Pain Unit
CT	Computertomograph
E2SFCA	Enhanced Two-Step Floating Catchment Area
ELDIS	Electronic Dispatching and Information System
GCS	Glasgow Coma Score
GTFS	General Transit Feed Specification
HKL	Herzkatheterlabor
INM	Institut für Notfallmedizin und Medizinmanagement
LMU	Ludwig-Maximilians-Universität
MLD	Multi-Level-Dijkstra
MRT	Magnetresonanztomograph
NNLS	Non-negative least squares
ODbL	OpenData Commons Open Database License
OSM	OpenStreetMap
OSRM	Open Source Routing Machine
PBF	Protocolbuffer Binary Format
RTW	Rettungswagen
SDSP	Single-Destination-Shortest-Path
SHT	Schädel-Hirn-Trauma
SPAI	Spatial Accessibility Index
SSSP	Single-Source-Shortest-Path
STEMI	Herzinfarkt (ST segment elevation myocardial infarction)
STROKE	Schlaganfall
VS	Versorgungsstufe
XML	eXtensible Markup Language
ZAST	Zentrale Abrechnungsstelle für den Rettungsdienst Bayern
ZRF	Zweckverband für Rettungsdienst und Feuerwehralarmierung

1 Einleitung

1.1 Stand der Forschung

Notfallpatienten mit Verdacht auf eine Tracer-Diagnose, sei es Herzinfarkt, Schlaganfall oder polytraumatische Schwerverletzung, müssen schnellstmöglich mit dem Rettungsdienst in die nächste geeignete Klinik für eine lebensrettende Versorgung transportiert werden. Dabei hat sich in Studien die sogenannte „Golden Hour“, also eine Stunde von Unfall, bzw. Eintritt der Erkrankung bis zur Intervention in der geeigneten Klinik, als maximales Zeitfenster herausgestellt, um Patienten mit entsprechendem Krankheitsbild mit höherer Wahrscheinlichkeit ohne bleibende Schäden behandeln zu können (Bayeff-Filloff et al. 2014; Edlich und Wish 2004; Kleber et al. 2013; Little 2010).

Ein wichtiger Bestandteil der „Golden Hour“ ist die Transportzeit vom Unfallort des Patienten in die Notaufnahme eines Krankenhauses, das die entsprechende Tracer-Diagnose behandeln kann, ohne zusätzliche sekundäre Verlegungen in weiterführende Krankenhäuser durchführen zu müssen. Die Transportzeit kann zum einen bodengebunden mittels Routing-Analyse in einem geographischen Informationssystem auf einem Straßen-Netzwerk ermittelt werden, zum anderen über die euklidische Distanz bei Einsatz von luftgestützten Rettungsmitteln wie Rettungshubschraubern (Freysse et al. 2018; Pedigo und Odoi 2010). Das dem Straßennetzwerk zu Grunde gelegte Geschwindigkeitsprofil beruht meist auf Fahrgeschwindigkeiten, gemessen und validiert anhand von realen PKW-Fahrzeiten (Haynes et al. 2006).

Dabei ist die geographische Verteilung der geeigneten Kliniken ein relevanter Faktor, um die Versorgung der Bevölkerung zu beurteilen. Über verschiedene Modelle können Krankenhaus-Einzugsbereiche berechnet werden. Diese basieren unter anderem auf der regionalen Verfügbarkeit, im Speziellen das Verhältnis zwischen der Bevölkerung und den verfügbaren medizinischen Einrichtungen als Maß der Erreichbarkeit (Khan 1992; Wang 2012), oder der geographischen Erreichbarkeit über Distanzen und Fahrzeiten, wie zum Beispiel die Floating Catchment Area Methode (Radke und Mu 2000; Luo und Wang 2003) angewendet auf das Huff-Modell (Jia et al. 2017), mit möglichen Erweiterungen um zum Beispiel einer Gewichtung der Distanzen innerhalb der berechneten Einzugsbereiche (Luo und Qi 2009; Ahlmeyer und Wittowsky 2018).

Doch nicht nur die räumliche Verteilung der Kliniken, sondern auch die der Bevölkerung ist maßgebend. Einige Studien beschäftigen sich mit der Evaluation des Zugangs zu medizinischen Einrichtungen im Zusammenhang mit der demographischen Zusammensetzung und der Verteilung der Bevölkerung im Untersuchungsgebiet entsprechend der Raumkategorien der Raumordnung (Love und Lindquist 1995; Mullen et al. 2014; Shannon et al. 1975).

In Bayern werden aktuell die Kliniken erfasst, die Notfallpatienten mit einer Tracer-Diagnose behandeln können. Bisher beruhte die Kenntnis über Klinikeignungen auf Erfahrungen in den bayrischen Integrierten Leitstellen, daher konnten keine fundierten Aussagen über die Verteilung der Kliniken und der damit verbundenen Versorgung der Bevölkerung getroffen werden. Somit basieren aktuell die Kenntnisse von strukturschwachen Regionen, hinsichtlich der medizinischen Notfallversorgung von Tracer-Diagnosen, auf Vermutungen. Da Notfallpatienten mit z.B. einem Herzinfarkt mit großer Wahrscheinlichkeit den Rettungsdienst

in Anspruch nehmen, um in die Notaufnahme einer Klinik zu gelangen, kann ein Routing auf einem Straßennetzwerk nicht mit einem klassischen PKW-Profil erfolgen. Ein Kastenwagen, wie ein Rettungswagen, weist abweichende Beschleunigungen, vor allem bei Straßen in Hanglagen, und ein anderes Verhalten in Kurven auf. Zudem haben Rettungswägen mit Sondersignal, wie Martinshorn und Blaulicht, besondere Wegerechte, wie das Befahren von Fußgängerzonen. Ein entsprechendes Profil findet sich zwar in der Literatur (Jánošíková et al. 2019), für eine Annäherung an die bayrischen Gegebenheiten ist ein empirisches Vorgehen zur Abschätzung von Geschwindigkeiten notwendig.

1.2 Ziele der Arbeit

Als Leitziel der Forschungsarbeit ist die Aufdeckung von regionalen Versorgungs- und Erreichbarkeitsdefiziten von medizinischen Einrichtungen in Bayern anzusehen, die zeitkritische Tracer-Diagnosen, wie Herzinfarkt, Schlaganfall, Sepsis, plötzlicher Kreislaufstillstand, Polytrauma und schweres Schädel-Hirn-Trauma, behandeln können. In der Forschungsarbeit sind somit Krankenhauseinzugsbereiche anhand von bodengebundener Routing-Analysen zu berechnen und, basierend darauf, die Versorgung der Bevölkerung zu beurteilen.

Die Grundlage stellt ein zu operationalisierendes, auf einem empirisch ermittelten Geschwindigkeitsprofil für Rettungswägen mit Sondersignal basierendes digitales Straßennetz dar, das in einer Routing-Analyse vom Unfallort in die Notaufnahme der Kliniken Anwendung findet. Über ein ausgewähltes Erreichbarkeitsmodell werden Einzugsgebiete berechnet, wodurch es möglich ist Aussagen über die regionale Erreichbarkeit zu treffen. Mögliche Versorgungsdefizite lassen sich, unter Einbeziehung der Bevölkerung und der regional-räumlichen Gliederung, über statistische Auswertungen ermitteln.

Das routing-fähige Straßennetz, frei erhältlich über OpenStreetMap, muss für die Analyse von Fahrzeiten von Rettungswägen angepasst werden. Dadurch, dass Rettungsfahrzeuge mit Sondersignal erweiterte Wegerechte besitzen, sind die einzelnen Straßenkategorien hinsichtlich der Relevanz der Befahrbarkeit zu bewerten und entsprechende Eigenschaften zugewiesen werden. Durch ein, im Vergleich zu einem PKW, anderes Verhalten hinsichtlich der Beschleunigung und der Kurvengängigkeit wird ein eigenes Geschwindigkeitsprofil entwickelt. Dabei werden reale Einsatzfahrten, mit bekanntem Startort und bekanntem Einsatzort zur Berechnung der Geschwindigkeiten der einzelnen Straßenkategorien herangezogen. Eine Regressionsanalyse, zur Ermittlung der Geschwindigkeiten, erfolgt in Python.

Für die Routing-Analyse zur Berechnung der Klinikeinzugsgebiete muss zunächst die Routing-Basis definiert werden. Als Startorte werden Gemeinden herangezogen, die als Punkt- und Polygoneometrie über das Institut für Notfallmedizin und Medizinmanagement (INM) des LMU-Klinikums in München zur Verfügung gestellt werden. Von derselben Einrichtung sind Informationen zur Eignung der bayrischen Kliniken, bezüglich der Behandlung von Patienten mit Tracer-Diagnosen, und der Adresse der entsprechenden Notaufnahmen erhältlich. Die Datenverwaltung erfolgt in einer SQLite-Datenbank.

Die Erreichbarkeitsanalyse, bzw. die Einzugsgebiete der Kliniken werden über die E2SFCA-Methode in einer SQLite-Datenbank berechnet. Dabei ist ein Schwellenwert zu definieren, der

die Transportzeiten vom Unfallort in die Notaufnahme als ausreichend betrachtet, damit die „Golden Hour“, also von Erkrankung bis Ankunft am Krankenhaus, eingehalten werden kann.

Zur Beurteilung der Versorgung der Bevölkerung muss die Bevölkerungsstruktur auf Ebene der Gemeinden herangezogen werden, die über das Bayrische Landesamt für Statistik (LfStat) verfügbar ist. Anschließend werden statistische Analysen durchgeführt, die die Versorgung der Bevölkerung, bezogen auf die Erreichbarkeit, berechnet aus dem gewählten Erreichbarkeitsmodell, bewertet. Hierbei werden die regionalen Unterschiede der Raumeinheiten miteinbezogen.

1.3 Aufbau und Struktur der Arbeit

Die vorliegende Arbeit gliedert sich in drei zentrale Teile. Zu Beginn stelle ich in Kapitel 2 die verwendete Datengrundlagen dar. Zunächst wird das Hintergrundwissen zur Krankenhauslandschaft in Bayern vermittelt sowie Tracer-Diagnosen mit entsprechenden Voraussetzungen zur Eignung eines Krankenhauses beschrieben. Die Entwicklung eines Geschwindigkeitsprofils basiert auf den Einsatzdaten der Integrierten Leitstellen in Bayern, die ebenfalls in diesem Kapitel beschrieben werden. Zentrale Daten stellt das digitale Straßennetz aus OpenStreetMap dar. Dabei wird auf die Basiskomponenten des Datenmodells eingegangen und befahrbare Straßenkategorien aus dem OSM-Datensatz identifiziert. Für die Verarbeitung räumlicher Daten und statistische Auswertungen werden weitere Daten wie administrative Grenzen und Bevölkerungsdaten herangezogen.

Der zweiten Teil handelt von der Entwicklung eines Geschwindigkeitsprofils für Rettungswägen. Zunächst erfolgt in Kapitel 3 eine Beschreibung des Graphenmodells mit anschließender Darstellung unterschiedlicher gängiger Routing-Algorithmen. Für die Auswahl einer Routing-Engine erfolgt eine Gegenüberstellung von drei möglichen Routing Services. Daran schließt sich die Operationalisierung des Straßengraphen an. Zunächst erfolgt die Beschreibung der Implementierung des OSRM-Backend-Dienstes, damit dieser für das Routing verwendet werden kann. Um ein Geschwindigkeitsprofil für Rettungswägen zu erhalten, müssen verschiedene Arbeitsschritte ausgeführt werden. Es wird die Zuweisung der OSM-Straßenkategorien zu neuen umfassenderen Straßenkategorien beschrieben sowie die Berechnung der mutmaßlichen Strecke der Einsatzfahrten von Rettungswägen. Daran schließt sich die Berechnung eines Geschwindigkeitsprofils an, mit entsprechender Validierung und Implementierung.

Im Anschluss daran in Kapitel 4 kommt das entwickelte Geschwindigkeitsprofil zur Anwendung in einer Erreichbarkeitsanalyse der Krankenhäuser. Dazu wird das Erreichbarkeitsmodell der Enhanced 2-Step Floating Catchment Areas vorgestellt und die Eingangsparameter ermittelt. Um Aussagen über die Erreichbarkeit von Kliniken treffen zu können werden im Weiteren die beiden Schritte zur Berechnung des Spatial Accessibility Index beschrieben, unter Anwendung von ermittelten Distanzgewichten für definierte Fahrzeitzonen. Für jede Tracer-Diagnose im Einzelnen sowie alle Tracer-Diagnosen gegenübergestellt erfolgt eine Darstellung der Ergebnisse.

Abschließend werden in Kapitel 5 die Ergebnisse aus der Erreichbarkeitsanalyse bezogen auf die Forschungsfrage diskutiert und bewertet. Es erfolgt zusätzlich ein kritischer Blick auf die verwendeten Methoden hinsichtlich ihrer Eignung für die Beantwortung der Forschungsfrage. Ein kurzer Ausblick auf eine mögliche Weiterentwicklung beschließt die Arbeit.

2 Datengrundlagen

2.1 Standorte relevanter Krankenhäuser

2.1.1 Krankenhauslandschaft zur Behandlung von Tracer-Diagnosen

A) Krankenhauslandschaft Bayern

In den letzten Jahren erfolgte ein struktureller Wandel in der Krankenhauslandschaft. Durch die Einführung der sogenannten Fallpauschale zur Vergütung von Gesundheitsleistungen, sog. „diagnosis-related groups“ (DRG), mussten viele unwirtschaftliche Kliniken schließen, bzw. umstrukturiert werden. Auf regionaler Ebene erfolgte eine Zunahme von Schwerpunktbildung und der Spezialisierung der Kliniken auf die Versorgung von Patienten in Zentren (Köther et al. 2009).

Krankenhäuser, die vollstationäre Fachabteilungen vorweisen können, unterscheiden sich nach der Art des Trägers (öffentlich, freigemeinnützig oder privat), der Anzahl der Betten, der Anzahl der Fachabteilungen sowie nach der Art der Zulassung. Die Zulassung wird in §108 des Fünften Buches Sozialgesetzbuch (SGB V) definiert, darunter finden sich Hochschulkliniken, Plankrankenhäuser und Krankenhäuser mit einem Versorgungsvertrag (Destatis 07.04.2022).



Abb. 1: Gliederung der zugelassenen Kliniken

Für die Planung einer bedarfsgerechten klinischen Versorgung der Bevölkerung wird jedes deutsche Bundesland rechtlich verpflichtet einen eigenen Krankenhausplan mit den zugelassenen Kliniken aufzustellen¹ (StMGP 2022). Darin inkludiert sind die Hochschulkliniken, also Kliniken, die neben der Akutversorgung von Patienten auch Forschung und Lehre betreiben, sowie Krankenhäuser, für welche ein besonderer Versorgungsvertrag besteht, aber keiner Versorgungsstufe zugeordnet werden, sogenannte Vertragskliniken. Der größte Teil der zugelassenen Kliniken umfassen die Plankrankenhäuser, die als Allgemeinkrankenhäuser ohne bestimmte Fachrichtung und als Fachkrankenhäuser mit Spezialisierung auf eine Fachdisziplin ausgewiesen werden. Je nach zu erfüllendem Versorgungsauftrag werden die Allgemeinkrankenhäuser in Versorgungsstufen untergliedert. Die Regelversorgung übernehmen Krankenhäuser mit akutstationärer Grundversorgung, diese werden der Versorgungsstufe I zugeordnet. Neben Diagnose und Therapie müssen Kliniken der Versorgungsstufe II auch überörtliche Schwerpunktaufgaben übernehmen, daher gelten diese

¹ Bayerisches Krankenhausgesetz (BayKrG): Art. 3 Abs. 1 (Grundsätze der Krankenhausplanung)

als Schwerpunktversorger. Als Maximalversorger der Versorgungsstufe III gelten Kliniken, die eine umfangreiche medizintechnische Ausstattung sowie ein umfassendes Behandlungsspektrum vorweisen können.

Für den weiteren Verlauf der Arbeit werden diejenigen Kliniken herangezogen, welche im aktuellen Krankenhausplan aufgeführt sind, da diese einen offiziellen Versorgungsauftrag aufweisen und entsprechend die akutmedizinische Inanspruchnahme der Bevölkerung sicherstellen. Dies umfasst insgesamt 411 zugelassene Krankenhäuser in Bayern.

Tab. 1: Verteilung der Kliniken nach Regierungsbezirk (StMGP 2022, 267f)

Regierungs- bezirk	Klinik nach Versorgungsstufe					Hochschulklinik	gesamt
	I Anzahl	II Anzahl	III Anzahl	Fachklinik Anzahl	ohne Anzahl	Anzahl	Anzahl
Oberbayern	43	11	4	86	16	2	162
Niederbayern	19	4	-	17	2	-	42
Oberpfalz	12	4	1	9	4	1	31
Oberfranken	9	6	4	13	6	-	38
Mittelfranken	18	2	2	18	1	1	42
Unterfranken	18	5	-	14	4	1	42
Schwaben	29	3	-	17	4	1	54
Bayern	148	35	11	174	37	6	411
Anteil	36,01 %	8,52 %	2,68 %	42,34 %	9 %	1,46 %	100 %

Insgesamt gibt es sechs Hochschulkliniken in Bayern (München, Regensburg, Erlangen, Würzburg und Augsburg), jedwede mit dem Freistaat Bayern als Träger. Mit 47,2% bestimmen den größten Teil die Plankrankenhäuser mit Versorgungsstufe und 42,3% die Fachkliniken. Dies umfasst insgesamt 368 Krankenhäuser. 37 bayrische Kliniken haben einen gesonderten Versorgungsvertrag und weisen somit keine Versorgungsstufe auf.

B) Beschreibung der Tracer-Diagnosen

Im weiteren Verlauf dieser Arbeit wird sich auf die Kliniken konzentriert, die eine Behandlung von zeitkritischen Krankheitsbildern leisten können. Diese sogenannten Tracer-Diagnosen sind gekennzeichnet durch den Schweregrad des Krankheitsbildes und der Häufigkeit des Auftretens (Schlechtriemen et al. 2002). Zudem liegen entsprechende Vorgaben vor, die mit den Leitlinien der jeweiligen wissenschaftlichen Fachgesellschaften abgestimmt wurden. Insbesondere die Dauer vom Absetzen des Notrufs, nach Auftreten des lebensbedrohlichen Notfalls, bis zur Aufnahme im nächsten geeigneten Krankenhaus mit einer entsprechenden fachspezifischen Versorgungsmöglichkeit sind entscheidend für den Behandlungserfolg. Diese Prähospitalphase sollte 60 Minuten, die sogenannte „Golden Hour“, nicht übersteigen, um die Überlebenschancen von Schwerverletzten zu erhöhen, bzw. irreparable körperliche Schäden zu minimieren. Die lebensrettende klinische Therapie, wie der Beginn der Notfalloperation bei Schwerverletzten, soll laut Empfehlung nach spätestens 90 Minuten nach Notrufeingang durchgeführt werden (Bayeff-Filloff et al. 2014; Fischer et al. 2016).

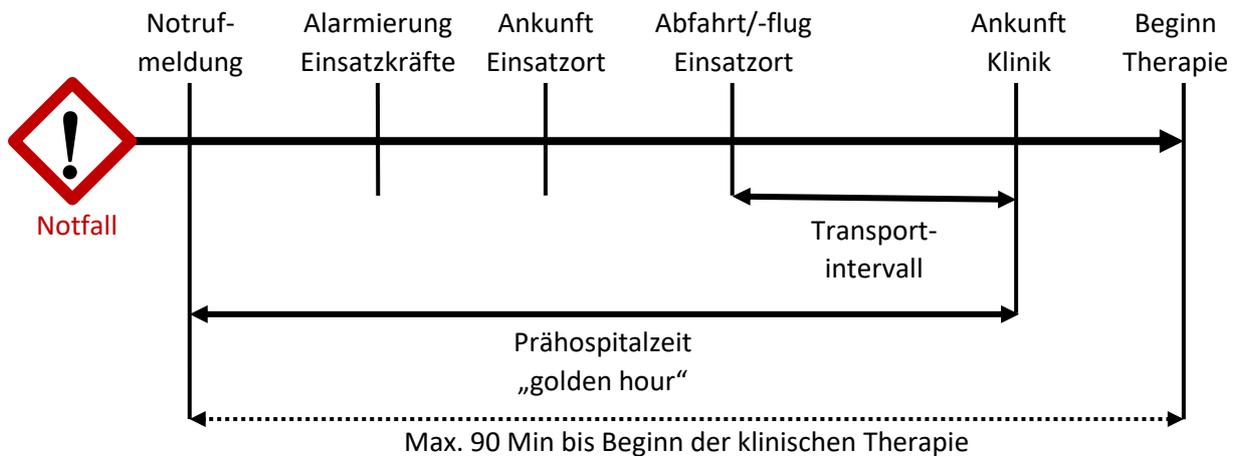
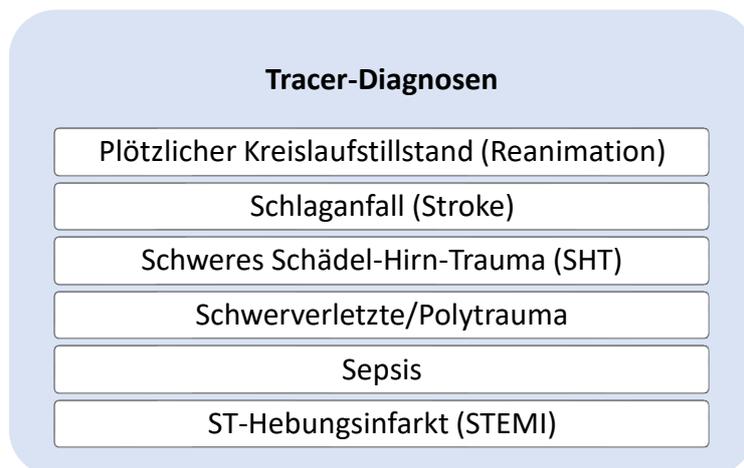


Abb. 2: Übersicht der relevanten Zeitintervalle (eigene Darstellung nach INM und Eckpunktepapier 2016)

Das Konzept der „Golden Hour“ wurde geprägt durch den amerikanischen Chirurgen R. Adams Cowley in den 1970er Jahren, der Forschung im Bereich der zivilen Traumaversorgung betrieb (Little 2010; Lerner und Moscati 2001), und ist bis heute ein feststehender Begriff bei der Definition von Zielen in der präklinischen notfallmedizinischen Versorgung. So wurde er auch im „Eckpunktepapier zur notfallmedizinischen Versorgung der Bevölkerung“ aufgegriffen (Fischer et al. 2016). Das „Eckpunktepapier 2016“ entstand aus einem Konsensus von Vertretern der entsprechenden wissenschaftlichen Fachgesellschaften und Organisationen sowie Fachexperten, und stellt die Grundlage für die notfallmedizinische Planung in Deutschland dar.

Ein wichtiger Inhaltspunkt ist die Anerkennung von notfallmedizinisch relevanten zeitkritischen Krankheitsbildern als Tracer-Diagnose, mit entsprechenden Kernaussagen und Empfehlungen auch zur Behandlung von Kindern.



Bei einem plötzlichen Kreislaufstillstand, erfolgt eine Sauerstoffunterversorgung von lebenswichtigen Organen, wie dem Gehirn, verursacht durch eine Störung der Herzfunktion und folglich der Blutzirkulation. Unverzüglich müssen Sofortmaßnahmen ergriffen werden, die die Sauerstoffversorgung aufrechterhalten, wie zum Beispiel Reanimation, bzw. Herz-Lungen-Wiederbelebung (Koster et al. 2010).

Ein Schlaganfall, auch Apoplex oder Stroke genannt, ist eine Folge eines Hirninfarkts, einer Durchblutungsstörung im Gehirn, oder einer Hirnblutung, bei der Blut im Gehirn austritt. Da es gleichzeitig zu neurologischen Ausfällen kommt wurde für den Rettungsdienst der sogenannte Face Arm Speech Test (FAST) ausgearbeitet, damit ein Schlaganfall präklinisch schnell diagnostiziert werden kann (Röther et al. 2017). Der Test befasst sich mit der Erkennung von Lähmungserscheinungen der Gesichtsmuskulatur, der Arme sowie Sprachstörungen.

Als ein schweres Schädel-Hirn-Trauma wird diagnostiziert bei einem Glasgow Coma Score (GCS) von weniger als neun Punkten (Teasdale und Jennett 1974). Dabei wird das Verhalten der Augen, der verbalen und motorischen Reaktion bewertet. Liegt der Scoring-Wert bei unter neun, so liegt per Definition eine schwere Bewusstseinsstörung als Folge einer Kopfverletzung vor.

Patienten mit Mehrfachverletzungen haben eine erhöhte Mortalitätsrate. Gemäß der Berliner Polytrauma-Definition müssen mindestens drei signifikante Verletzungen in zwei unterschiedlichen anatomischen Regionen identifiziert werden (Pape et al. 2014). Zusätzlich muss mindestens eine der folgenden unabhängigen physiologischen Variablen vorliegen: ein zu niedriger Blutdruck, ein Glasgow Coma Score von weniger als 8, ein zu niedriger pH-Wert im Blut, eine verminderte Blutgerinnung und der Patient hat ein Alter von mindestens 70 Jahren.

Die Sepsis ist eine lebensbedrohliche Organdysfunktion, der eine Infektion vorausgeht. Die Immunantwort, also die körpereigene Abwehrreaktion, schädigt dabei Gewebe und Organe, wodurch ein Organversagen oder ein septischer Kreislaufschock ausgelöst werden kann (Bracht et al. 2019).

Ein ST-Hebungsinfarkt, bzw. Herzinfarkt oder Myokardinfarkt, wird verursacht durch das Absterben von Herzmuskelzellen, als Folge einer Durchblutungsstörung, wie zum Beispiel Blutgerinnsel. Besonders charakteristisch ist die Hebung der ST-Strecke, sichtbar bei Anfertigung eines EKGs, daher auch die gängige Abkürzung STEMI für ST-elevation myocardial infarction (Schmailzl et al. 2003).

C) Definition der Relevanz nach Diagnose

Einheitliche rettungsdienstliche Handlungsempfehlungen werden in Bayern durch den Rettungsdienstausschuss aufgestellt, der durch Art. 10 des Bayrischen Rettungsdienstgesetzes legitimiert ist. Arbeitsgruppen erarbeiten verschiedene Themen und stehen dem Rettungsdienstausschuss beratend bei. Nach dem Beschlussantrag 013/2017 beschäftigte sich insbesondere die Arbeitsgruppe „AG 5 – Patientenverteilung & Behandlungskapazitäten“ mit den Voraussetzungen, die Kliniken erfüllen müssen um als geeignet zur Behandlung von Tracer-Patienten angesehen zu werden. Mit dem Dokument „Strukturelle Voraussetzungen von Akutkliniken zur Behandlung von Tracer-Diagnosen“ legte die AG 5 die Empfehlung 02/5 – 2019 dem Rettungsdienstausschuss vor, woraufhin diese zur Ausführung angenommen wurde (Rettungsdienstausschuss Bayern 13.03.2019). Das Institut für Notfallmedizin entwickelte daraufhin ein Online-Eingabetool zur Erfassung der entsprechenden Daten für die Kliniken in Bayern. Aus den dadurch erhobenen Informationen kann die Eignung der Kliniken für die jeweiligen Tracer-Diagnosen festgestellt werden. Nachfolgende Kriterien-Matrix zeigt die strukturellen Einrichtungen, die ausschlaggebend für die Eignungsbeurteilung sind.

Tab. 2: Kriterien-Matrix zur Versorgung von Tracer-Diagnosen bei Erwachsenen (eigene Darstellung nach Empfehlung 02/5-2019)²

Fachbereich	Fachgebiet	SHT (GCS <9)	Polytrauma (A und B)	STEMI	STROKE	Sepsis	Reanimation
Diagnostik/ Geräte/ Therapien	CPU			X			
	CT (24/7)	X		(X)	X	X	X
	HKL (24/7) (pPCI)			X			X
	Intensivtherapie/-station			X		X	X
	MRT				(X)		
	Schnelllabor (24/7)			X		X	
	Schockraum	X	X				
	Teleradiologische Anbindung						(X)
	Temperaturmanagement/ Hypothermie						X
Chirurgie	Allgemeine Chirurgie					X	
	Möglichkeit zur Sanierung des Infekt-Fokus					X	
	Neurochirurgie, bzw. neurotraumatologische Intervention	X					
Innere Medizin	Innere Medizin				X		
Neurologie	Fachneurologischer Dienst						(X)
	zertifizierte Stroke-Unit (auch TeleStroke)				X		
Sonstige Einrichtung	Herzinfarktnetzwerk			X			
	Notaufnahme (24/7)					X	
	Traumazentrum lokal		X				
	Traumazentrum regional/überregional	X	X				

Insgesamt werden für Bayern 148 Krankenhäuser als geeignet betrachtet. Diese verteilen sich auf folgende Raumkategorien:

Tab. 3: Anzahl der Krankenhäuser nach Raumtyp in Bayern

Raumtyp	Anzahl Krankenhäuser
Allgemeiner ländlicher Raum	66
Ländlicher Raum mit Verdichtungsansätzen	23
Verdichtungsraum	59

Daraus ergibt sich für jede Tracer-Diagnose folgende Verteilung der Kliniken in Bayern:

² Legende: X – obligatorisch; (X) - optional

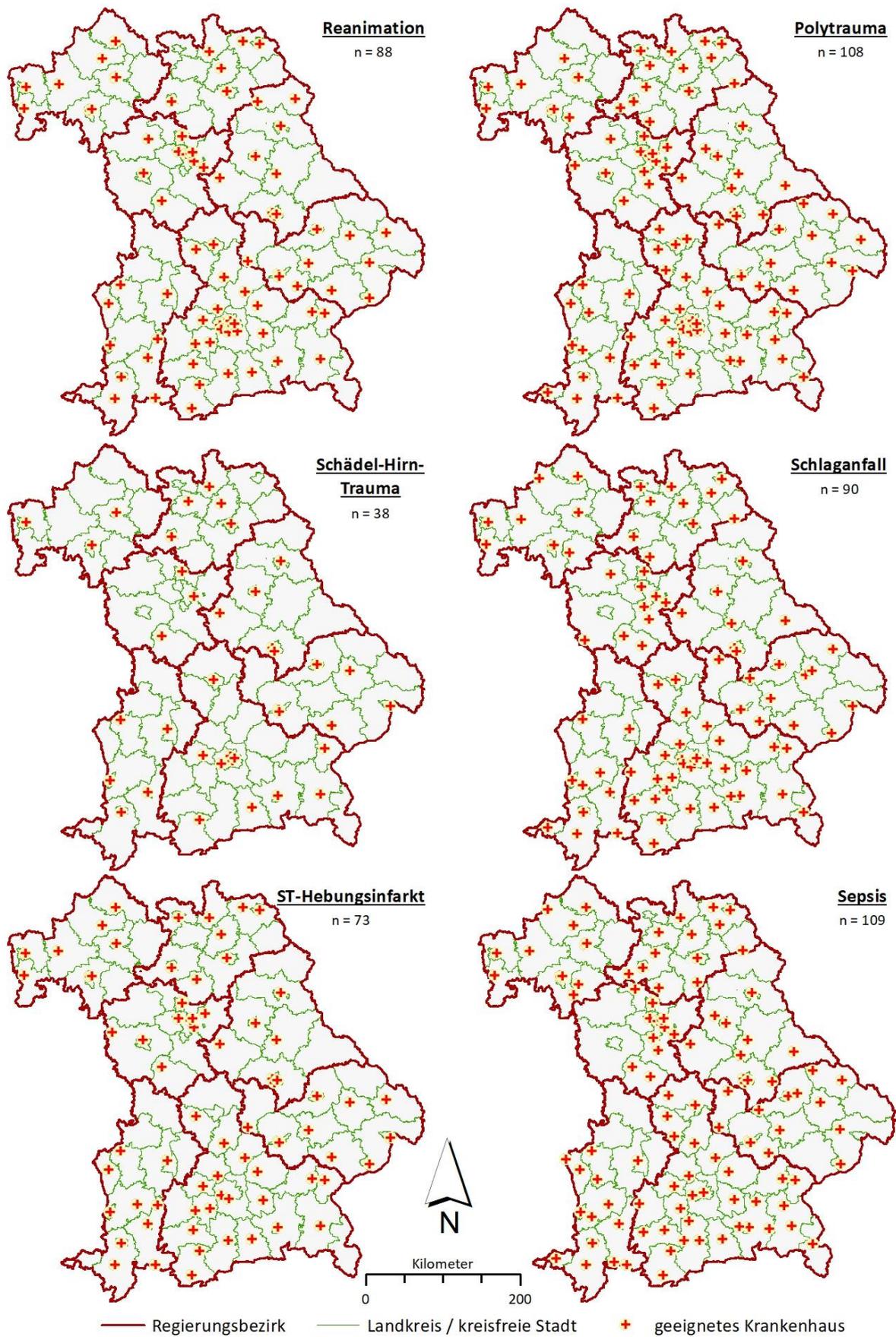


Abb. 3: Überblick der Krankenhausverteilung nach Tracer-Diagnose

2.1.2 Beschreibung des Krankenhausdatensatzes

Die Krankenhausdaten werden kontinuierlich gepflegt und für jedes relevante Fachgebiet wird die Verfügbarkeit minutengenau erhoben. Somit kann für einen gewünschten Stichtag und Zeitpunkt der entsprechende Stand abgerufen werden. Da tagsüber alle Fachgebiete einsatzbereit sind, werden die entsprechenden Versorgungsmöglichkeiten um 12:00 Uhr ausgegeben. Insgesamt wird für 148 Kliniken die Behandlungsmöglichkeit von einer oder mehreren Tracer-Diagnosen konstatiert.

Das INM ist eine interdisziplinäre klinische Einrichtung im LMU Klinikum München. Es erfüllt die Aufgaben der Notfallmedizin und des Medizinmanagements in Forschung, Lehre und Krankenversorgung. Die Schwerpunkte reichen von gutachterlichen Struktur- und Bedarfsanalysen einzelner Versorgungssektoren, z.B. im Bereich des Rettungswesens, bis hin zur Entwicklung von Simulationsverfahren von Prozessen und sektorenübergreifenden Trendanalysen unter Berücksichtigung raumbezogener Daten. Die vom INM bereitgestellten Krankenhausdaten liegen als Tabelle im TXT-Format vor. Zur weiteren Verarbeitung wird diese in eine SQLite-Datenbank (Zeichenkodierung ISO-8859-2 Latin-2 Central European aufgrund der Umlaute und „ß“) importiert.

In der Basistabelle sind die X- und Y-Koordinaten der einzelnen Krankenhausstandorte enthalten. Damit die räumlichen Informationen genutzt werden können, müssen die Koordinaten in einem Array-Feld als Punktgeometrie gespeichert werden. Die Geometrie-Spalte wird über die Funktion *AddGeometryColumn* mit dem Geometrietyp *POINT* und dem SRID 4326 (geodätisches Referenzsystem WGS84) erstellt.

	f_table_name	f_geometry_column	geometry_type	coord_dimension	srid	spatial_index_enabled
1	kh_tracer	geometry	1	2	4326	1

Abb. 4: Registrierung der Geometrie in der Systemtabelle *geometry_columns*

Insgesamt finden sich 148 Krankenhausdatensätze in der Tabelle, daher ist das Anlegen eines Spatial Index nicht notwendig, dennoch erfolgt die Implementierung eines räumlichen Index. Eine vollständige Tabellendurchsuchung (Full Table Scan) bei Datenabfragen weist bei einer geringen Anzahl an Datensätzen auch ohne Index eine gute Performance auf. In SQLite basiert die räumliche Indizierung auf dem R*-Tree-Algorithmus, eine Erweiterung des R-Tree Prinzips. Dieser ist hierarchisch strukturiert, bei dem jeder Knoten des Baums sich aus einer Gruppe an mehrdimensionalen Daten zusammensetzt. Der Raum, in welchem die Daten liegen, werden durch umschreibende Rechtecksgeometrien organisiert. Als End-Knoten, sogenannte Leaf-Nodes des Index-Baums fungieren die minimal umschreibenden Rechtecke (Minimum Bounding Rectangle), die über Pointer auf die enthaltenen Geometrien verweisen. Darüberliegende Knoten sind räumliche Cluster, aus zusammengefassten umschließenden Rechtecken, die wiederum als minimal umschließende Rechtecke definiert sind. Charakteristisch für die R*-Tree-Struktur ist, dass die Suchräume disjunkt sind, also eine Überlappung der umschließenden Rechtecke, anders als in der R-Tree-Struktur, ausgeschlossen ist. Dadurch wird die Anzahl der möglichen Suchpfade verringert. Zudem ist die Größe der umschlossenen Fläche sowie deren Begrenzung reduziert, wodurch die Speicherkapazitäten besser ausgenutzt werden und die Struktur robuster wird (Beckmann et al. 1990).

Tab. 4: Datentypen der Krankenhaustabelle (Stand: 19.12.2022 12:00 Uhr)

Spaltenname	Datentyp	PrimaryKey	Beschreibung
ID_KH_INM	text	X	Fortlaufende eindeutige Kennzeichnung der einzelnen Datensätze
KH_NAME	text		Kurzname für das Krankenhaus
ANSCHRIFTNAME	text		Vollständiger Name des Krankenhauses
V_STUFE	text		Versorgungsstufe
VERTRAGSSTATUS	text		Vertragsstatus nach Krankenhausplan
TD_REA	integer		Eignung für Tracerdiagnose Reanimation
TD_POLYT	integer		Eignung für Tracerdiagnose Polytrauma
TD_SHT	integer		Eignung für Tracerdiagnose Schädel-Hirn-Trauma
TD_STROKE	integer		Eignung für Tracerdiagnose Herzinfarkt
TD_STEMI	integer		Eignung für Tracerdiagnose ST-Hebungsinfarkt
TD_SEPSIS	integer		Eignung für Tracerdiagnose Sepsis
KOO_LON	integer		X-Koordinate des Krankenhausstandorts
KOO_LAT	integer		Y-Koordinate des Krankenhausstandorts
GEOMETRY	POINT		Geometrische Entität

Die Standorte der Krankenhäuser befinden sich innerhalb folgender Ausdehnung:

$Y_{\max} = 50,324547$	
$X_{\min} = 9,166023$	$X_{\max} = 13,546325$
$Y_{\min} = 47,483690$	

Beispieldatensatz:

ID_KH_INM	=	0916100000100100	
KH_NAME	=	KL Ingolstadt	
ANSCHRIFTNAME	=	Klinikum Ingolstadt	
V_STUFE	=	II	III; II; I; F; BG
VERTRAGSSTATUS	=	Plan-KH	Plan-KH; Vertrags-KH; HS-KL
TD_REA	=	1	1 = geeignet; 0 = ungeeignet
TD_POLYT	=	1	1 = geeignet; 0 = ungeeignet
TD_SHT	=	1	1 = geeignet; 0 = ungeeignet
TD_STROKE	=	1	1 = geeignet; 0 = ungeeignet
TD_STEMI	=	0	1 = geeignet; 0 = ungeeignet

TD_SEPSIS = 0 | 1 = geeignet; 0 = ungeeignet
KOO_LON = 11,377359
KOO_LAT = 48,770289

2.2 Einsatzdaten Integrierter Leitstellen in Bayern

2.2.1 Datenerfassung der Integrierten Leitstellen Bayern

Der Freistaat Bayern ist in insgesamt 26 Integrierte Leitstellen eingeteilt, die über die gemeinsame Notrufnummer 112 erreicht werden. Seit 2002 kann unter der Notfallnummer nicht mehr nur die Feuerwehr, sondern auch der Rettungsdienst gerufen werden, daher die Bezeichnung der Integrierten Leitstelle. Diese können schnelle Hilfe durch die gezielte Alarmierung im Brandfall oder bei einem medizinischen Notfall leisten. Der Leitstellendisponent nimmt den Anruf entgegen, alarmiert die entsprechenden Einsatzkräfte und begleitet den Einsatz. Träger der integrierten Leitstellen sind die jeweiligen Zweckverbände für Rettungsdienst und Feuerwehralarmierung (ZRF), die die Aufgabe haben den öffentlichen Rettungsdienst sicherzustellen.

Das Bayerische Staatsministerium des Innern für Bau und Verkehr (BayStMI) und die Sozialversicherungsträger in Bayern haben das Institut für Notfallmedizin und Medizinmanagement (INM) am LMU Klinikum München mit der Pflege einer Strukturdatenbank, die Informationen zu Rettungsdienststandorten und Krankenhausstrukturdaten enthält, und einer Einsatzdatenbank beauftragt. Die Einsatzdaten der Integrierten Leitstellen Bayerns werden regelhaft an das INM übermittelt. Einmal pro Quartal erhält das INM die Einsatzdaten aus dem Einsatzleitsystem ELDIS III by Eurofunk Kappacher GmbH im XML-Format. Die Einsatzdaten der Leitstellen werden durch ausgewählte Daten der Zentralen Abrechnungsstelle für den Rettungsdienst Bayern (ZAST GmbH) sowie durch Strukturdaten der räumlichen Verteilung und zeitlichen Besetzung der Rettungsdienststandorte ergänzt.

Neben dem Datum und der Uhrzeit des Einsatzes und den Fahrzeugen, die am Einsatz beteiligt waren, werden auch einzelne Zeitstempel dokumentiert. Aus den Daten kann unter anderem herausgelesen werden, wann das jeweilige Fahrzeug alarmiert wurde, wann es ausgerückt ist, den Zeitpunkt der Ankunft beim Patienten, wann mit dem Patienten wieder losgefahren wurde und den Zeitpunkt der Einlieferung in der Klinik.

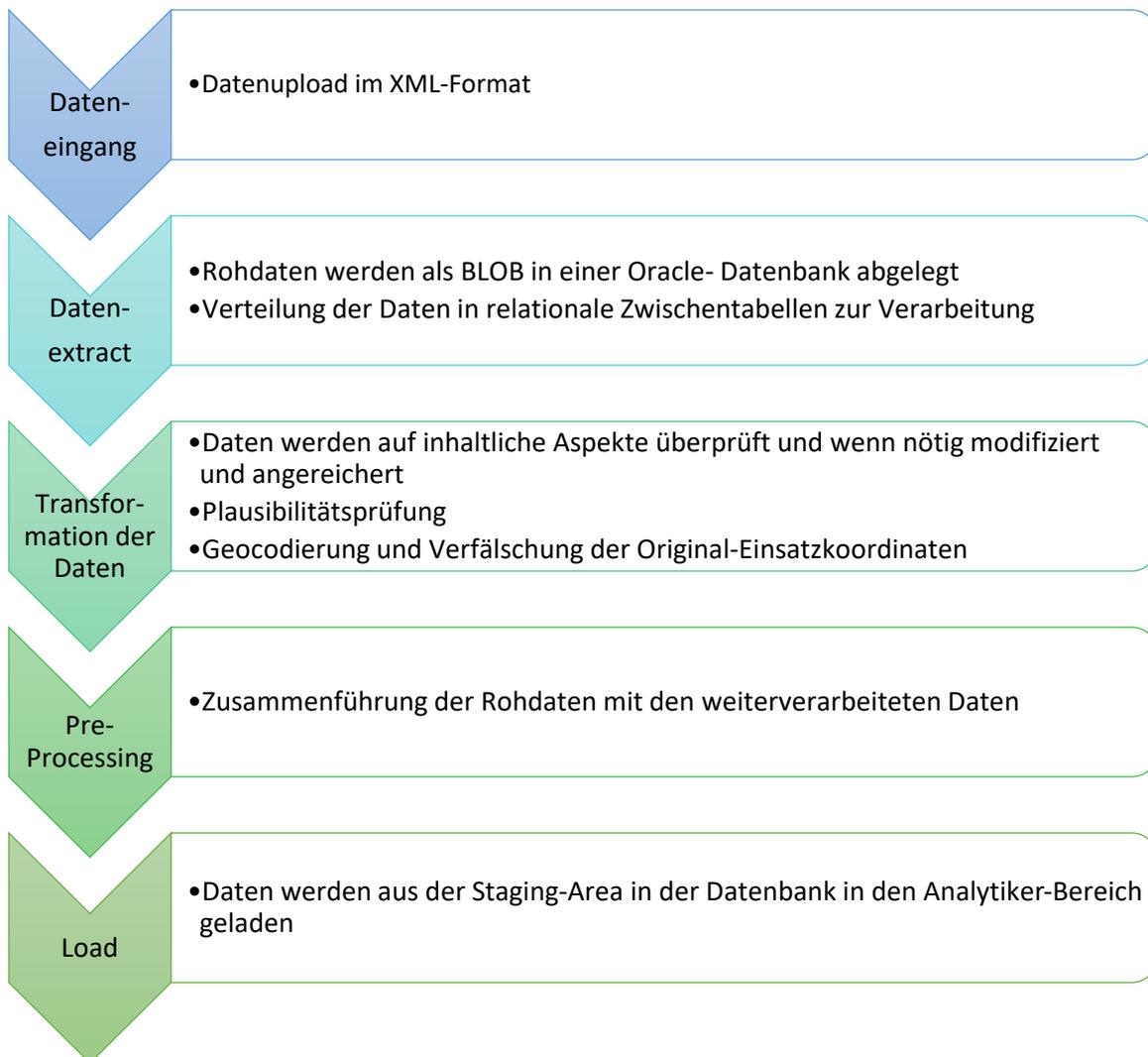


Abb. 5: Prozess der Datenverarbeitung im INM

Aus einsatztaktischen Gründen erlaubt das Einsatzleitsystem ELDIS IIIby dem Disponenten, bestimmte Eingaben zu überspringen. Zu diesen Eingabefeldern gehören auch Zeitangaben, die für die Berechnung von Zeitintervallen von Bedeutung sind (vgl. Abb. 2). Daher werden Zeit- und Ortsangaben des Einsatzablaufs hinsichtlich Vollständigkeit und Plausibilität überprüft und Auffälligkeiten korrigiert, bzw. ergänzt. Falls einzelne Zeitintervalle eines Einsatzes beispielsweise negative Werte enthalten, dies ist dann der Fall, wenn die Ankunft am Einsatzort vor der Abfahrt zum Einsatzort dokumentiert wurde, werden alle Zeitangaben dieses Fahrzeugs gelöscht, um fehlerhafte Interpretationen dieses Einsatzes zu vermeiden.

Die Geokoordinaten der Einsatz- und Zielorte durchlaufen ein Verfahren der Anonymisierung, gemäß der geltenden Datenschutzbestimmung, darüber hinaus werden die Daten aus ELDIS IIIby einer Überprüfung der Plausibilität der Ortsangaben unterzogen. Es wird u. a. geprüft, ob die Angaben zum Einsatzort (Postleitzahl und Ort) mit den Geokoordinaten übereinstimmen. Hierzu wurde seitens des INM ein aufwändiges, softwaregestütztes Verfahren entwickelt, welches die dokumentierten Ortsangaben in einem mehrstufigen Prozess auf Ebene der Gemeindeteile verortet. Sollten sich die Ortsangaben der Geokoordinaten von den Angaben zum Einsatzort (Postleitzahl und Ort) unterscheiden, wird eine individuelle Prüfung der Einsatzdokumentation vorgenommen.

2.2.2 Beschreibung der Einsatzdaten

Die vom INM bereitgestellten Einsatzdaten des bayrischen Rettungsdienstes liegen als Tabelle im TXT-Format vor. Zur weiteren Verarbeitung wird diese in eine SQLite-Datenbank (Zeichenkodierung ISO-8859-2 Latin-2 Central European aufgrund der Umlaute und „ß“) importiert.

Insgesamt finden sich 716.332 Datensätze aus einem Zeitraum von 01.01.2021 bis 31.12.2021 in der Tabelle. In ELDIS Illby werden neben den Angaben zu Postleitzahl, Ort und Objekt (z. B. Arztpraxis oder Altenheim) auch die Geokoordinaten der Einsatz- und Zielorte durch die Leitstelle erfasst. Die Basistabelle beinhaltet somit die X- und Y-Koordinaten für den Standort, von welchem der Rettungswagen ausgerückt ist. Dies bedingt nicht die Wache des entsprechenden Fahrzeugs. Es ist möglich, dass ein Rettungswagen einen Einsatz zum Beispiel mit der Übergabe eines Patienten im Krankenhaus der Leitstelle als beendet meldet, und dieser auf dem Weg zurück zur Wache für einen Folgeinsatz alarmiert wird. Somit liegt in diesem Fall der Standort des Einsatzbeginns auf dem Weg zur Wache. Zudem finden sich die Geokoordinaten der Einsatzstelle in der bereitgestellten Tabelle. Diese Geokoordinaten werden, unter Berücksichtigung der Vorgaben des Datenschutzes, für die Verortung des Einsatzaufkommens herangezogen. Die Geokoordinaten selbst werden aus Datenschutzgründen mit reduzierter Auflösung herausgegeben, so dass keine Rückschlüsse auf einzelne Adressen möglich sind. Eine Ausnahme stellen bekannte Objekte dar, die Koordinaten für zum Beispiel Altenheime oder Arztpraxen dürfen mit voller Stellenzahl gespeichert und für die weitere Verarbeitung genutzt werden. Im weiteren Verlauf der Arbeit wird die Strecke vom Einsatzort zu Kliniken geroutet. Da das Fahrverhalten eines Rettungswagens mit Patient im Fahrzeug wahrscheinlich anders ausfällt, als auf der Anfahrt zu einem Einsatzort, so wäre es angebracht entsprechende Fahrten aus den Einsatzdaten für die Fahrprofilerstellung zu verarbeiten. Die Anzahl der dafür in Frage kommenden Datensätze wäre aber zu gering, bzw. sind nicht alle Attribute zuverlässig dokumentiert. Aufgrund dessen stellt die Referenz für das Fahrprofil der Weg von Wache zum Einsatzort dar. Die räumlichen Informationen der Start- und Zielorte werden in die SQLite-Datenbank über die Funktion *AddGeometryColumn* mit dem Geometriotyp *POINT* und dem SRID 4326 (geodätisches Referenzsystem WGS84) importiert.

Aufgrund der großen Anzahl der Datensätze bietet es sich an einen Spatial Index auf die Geometry des Startpunkts (*GEOMETRY_VON*) und die Geometry des Zielortes (*GEOMETRY_NACH*) zu erstellen.

	f_table_name	f_geometry_column	geometry_type	coord_dimension	srid	spatial_index_enabled
1	rtw_einsatz	geometry_von		1	2 4326	1
2	rtw_einsatz	geometry_nach		1	2 4326	1

Abb. 6: Registrierung der Geometrie in der Systemtabelle *geometry_columns*

Da in SQLite kein Datenformat für Datums- und Zeitangaben vorgesehen ist, werden entsprechende Inhalte bei Import der Basisdaten als TEXT gespeichert. Um die Möglichkeit zu bewahren mit dem Zeitwert zu arbeiten wird dieses zunächst über die *julianday()*-Funktion in eine Gleitkommazahl umgewandelt. Die verwendete Funktion berechnet die Anzahl der Tage seit dem 24. November 4714 v. Chr. 12:00 Uhr Greenwich-Zeit im gregorianischen Kalender, bis zum eingegebenen Zeitpunkt. Wird das Ergebnis wiederum in eine *time()*-Funktion

eingefügt, so ergibt sich die bekannte Zeitangabe aber nicht mehr als TEXT, sondern im Datentyp REAL.

Tab. 5: Datentypen der Einsatzdatentabelle (Beobachtungszeitraum: 01.01.2021 - 31.12.2021)

Spaltenname	Datentyp	PrimaryKey	Beschreibung
ID_EINSATZ	integer	X	Fortlaufende eindeutige Kennzeichnung der einzelnen Datensätze
DATUM	text		Datum des Einsatzbeginns
TIME	real		Uhrzeit des Einsatzbeginns
PK_WACHE	integer		numerische Kennzeichnung der Wache des Rettungswagens, der als erster am Einsatzort eingetroffen ist
WACHE_NAME	text		Name der Wache des Rettungswagens, der als erster am Einsatzort eingetroffen ist
KFZART_FUNKRUF	text		Funkrufname des ersteintreffenden Rettungswagens
EINSATZORT	text		Name der administrativen Einheit des Einsatzortes
VON_X	real		X-Koordinate des Standorts des ersteintreffenden Rettungswagens zu Beginn des Einsatzes
VON_Y	real		Y-Koordinate des Standorts des ersteintreffenden Rettungswagens zu Beginn des Einsatzes
NACH_X	real		X-Koordinate des Einsatzortes
NACH_Y	real		Y-Koordinate des Einsatzortes
FAHRZEIT	real		die Zeit, die sich aus den dokumentierten Zeitstempeln des Ausrückzeitpunkts bis zum Ankunftszeitpunkt am Einsatzort ergibt
GEOMETRY_VON	POINT		Geometrische Entität des Standorts des Rettungswagens
GEOMETRY_NACH	POINT		Geometrische Entität des Einsatzortes

Beispieldatensatz:

ID_EINSATZ = 54889533
DATUM = 14.01.21
TIME = 14:38 | Format: hh:mm
PK_WACHE = 322
WACHE_NAME = RW Aschaffenburg MHD

KFZART_FUNKRUF	=	RTW Johannes Aschaffenburg 71/1	
EINSATZORT	=	Obernau - Aschaffenburg	
VON_X	=	9,162200	
VON_Y	=	49,987390	
NACH_X	=	9,135041	
NACH_Y	=	49,933206	
FAHRZEIT	=	10,12	in Minuten

Im Datensatz enthalten ist auch der Funkrufname *KFZART_FUNKRUF* des ersteintreffenden Rettungswagens. Dieser setzt sich aus der Art des Rettungsmittels, der Hilfsorganisation, die die Heimatwache des Rettungsmittels betreibt, die örtliche Zuordnung gefolgt von der Teilkennzahl für Landrettungsdienst und der laufenden Nummer für Fahrzeuge gleicher Art zusammen (Bayerisches Staatsministerium des Innern, für Sport und Integration 06.11.2014). RTW steht im Beispieldatensatz für Rettungswagen, Johannes für den Malteser Hilfsdienst mit der Heimatwache in Aschaffenburg. 71/1 bezeichnet den ersten Rettungswagen der entsprechenden Wache.

2.3 Digitales Straßennetz

2.3.1 Digitales Straßennetzes aus OpenStreetMap-Daten

Das OpenStreetMap-Projekt ist eine Community-gesteuerte editierbare Weltkarte. Es setzt auf lokales Wissen und individuelle Beiträge gegenüber Massenimporten. Die Menge der von der Community bereitgestellten Geodaten des OSM-Projekts steigt so immer weiter an. Zum Stand 03.02.2023 waren 10.006.651 User registriert, die insgesamt 8.195.276.218 Knoten, 917.975.331 Wege und 10.638.455 Relationen erfasst haben (OpenStreetMap Wiki 2023). In vielen Bereichen entspricht die Datenqualität bereits der kommerzieller Kartendaten, und, besonders in abgelegenen Gebieten, übertrifft diese sogar. Somit werden auch abgelegene Gebiete abgedeckt, in welchen keine Kartierungsversuche unternommen wurden. Die Daten sind kostenlos und frei verfügbar, also ohne urheberrechtliche Beschränkungen unter der Open Data Commons Open Database License (ODbL³). Dies bedeutet allerdings auch, dass Produkte, die OpenStreetMap-Daten nutzen, ebenfalls unter die Creative Commons-Lizenz fallen und somit auch ohne Beschränkungen weitergegeben werden dürfen.

³ Der genaue Inhalt der Lizenz kann hier gefunden werden: <https://opendatacommons.org/licenses/odbl/1-0/>

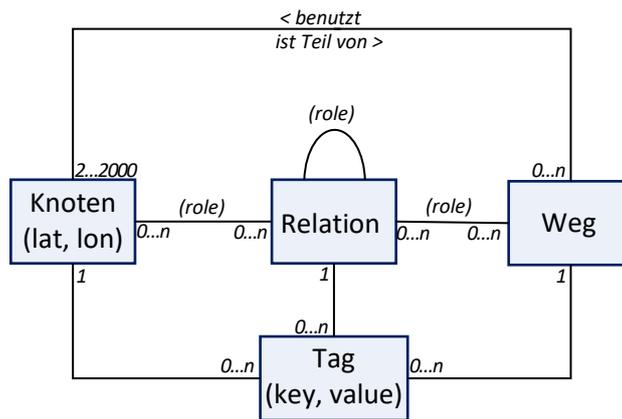


Abb. 7: Vereinfachtes konzeptionelles OSM-Datenmodell (Minghini und Frassinelli 2019; Ramm und Topf 2010)

Die drei Basiskomponenten im OSM-Datenmodell sind Knoten, Wege und Relationen (Eindhoven University of Technology 2023, 55–57; Ramm und Topf 2010, 55–57). Jedes Element ist mit Attributen versehen, die die spezifischen Features näher beschreiben. Um die einzelnen Objekte zu identifizieren, erhalten diese eine eindeutige numerische Kennung. Dabei wird ein unterschiedlicher Nummernraum angewendet, sodass die drei Objekttypen dieselben IDs aufweisen können. Daher ist es möglich, dass zum Beispiel jeweils ein Knoten und ein Weg die ID 43 tragen. Knoten repräsentieren einzelne Punkte auf der Karte mit Längen- und Breitenangabe im WGS84 Referenzsystem. Zum einen können Knoten punktartige Elemente darstellen, wie Points of Interest (Parkbank, Bushaltestelle, Denkmal, Hydrant) mit zusätzlichen Merkmalen als Tag gespeichert, zum anderen können Knoten den Verlauf von Wegen bzw. Formen codieren, sie dienen hierbei als Stützpunkte.

Beispiel XML für einen Knoten:

```
<node id="2821328966" lat="48.541985" lon="12.1788287" version="1">
  <tag k="height" v="2.5"/>
  <tag k="historic" v="boundary_stone"/>
  <tag k="name" v="Burgfriedensäule"/>
</node>
```



Wege werden somit durch eine geordnete Liste von mindestens zwei bis maximal 2000 Knoten definiert, die einer Straße (Polyline) oder dem Umriss eines Hauses (geschlossenes Polygon) entsprechen können (Ramm und Topf 2010, 57; Eindhoven University of Technology 2023, 57; OpenStreetMap Wiki contributors 2023). Ist die maximale Anzahl an Punkten überschritten, so werden Multipolygone modelliert, die zusammengenommen die Gesamtfläche ergeben. Die Datenstruktur unterscheidet sich nicht grundsätzlich zwischen einer Polylinie und einem Polygon. Polygone weisen zusätzlich denselben Knoten am Anfang und am Ende des Weges auf, zudem muss ein Attribut angegeben sein, das eine Fläche impliziert, wie zum Beispiel *area=yes* oder *building=yes*. Linien können auch ein geschlossenes Element darstellen, ohne eine Fläche zu bilden. Dies ist zum Beispiel bei Kreisverkehren der Fall, hier finden sich zusätzliche Tags, die entsprechende Merkmale wie *junction=roundabout* beschreiben. Der Standort der Wege wird nicht gespeichert, sondern dieser setzt sich aus der Lage der Knotenkennungen zusammen.

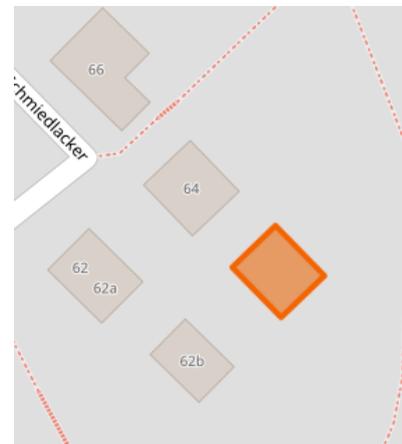
Beispiel XML für ein linienförmiges Objekt:

```
<way id="13859782" version="1">
  <nd ref="463831429"/>
  <tag k="highway" v="residential"/>
  <tag k="maxspeed" v="30"/>
  <tag k="name" v="Grünlandstraße"/>
</way>
```



Beispiel XML für ein flächenförmiges Objekt, Anfangs- und Endknoten sind gleich:

```
<way id="946149974" version="1">
  <nd ref="8758811191"/>
  <nd ref="8758811192"/>
  <nd ref="8758811193"/>
  <nd ref="8758811194"/>
  <nd ref="8758811191"/>
  <tag k="building" v="yes"/>
</way>
```



In der OpenStreetMap-Datenbank werden neben den Knoten und Wegen auch Relationen erfasst. Relationen sind ebenfalls eine geordnete Liste, die entweder Knoten, Wege oder sogar andere Relationen enthält. Diese Liste besteht aus Teilnehmern, die eine bestimmte Rolle erfüllen. Es wird verwendet, um logische oder geographische Beziehungen zwischen Objekten zu modellieren (Ramm und Topf 2010, 57–58; Eindhoven University of Technology 2023, 57–58). So ist es möglich komplizierte Flächen darzustellen, oder öffentliche Verkehrsrouten (z.B. die Fahrstrecke einer Buslinie) als Liste von Wegen zu kennzeichnen. Im nachfolgenden Beispiel wird die Relation zwischen Wegen und einem Knoten beschrieben. Von Weg_(from) 45325342 kommend kann am Knoten_(via) 575917034 nur nach rechts auf den Weg_(to) 100292937 abgebogen werden.

Beispiel XML für eine Relation:

```
<relation id="1433953" version="1">
  <member type="way" ref="45325342" role="from"/>
  <member type="way" ref="100292937" role="to"/>
  <member type="node" ref="575917034"
role="via"/>
  <tag k="restriction" v="only_right_turn"/>
  <tag k="type" v="restriction"/>
</relation>
```



Knoten und Wege weisen Attribute auf, um deren Merkmalstyp zu beschreiben. Die Anzahl der Attribute ist nicht beschränkt. Dieser sogenannte Tag ist die Kombination aus einem Schlüssel und einem Wert (*key k=" value v="*) und ist in UTF-8 kodiert, um internationalen Schriftkonventionen zu entsprechen. Zum Beispiel wird der Name eines Objekts mit dem Tag *name* definiert, somit erhält ein Denkmal (*historic = memorial*) in München den Tag *name = Maxmonument*. Für eine einheitliche Tag-Nutzung müssen bei der Digitalisierung von Objekten vorgegebene Konventionen eingehalten werden. Ist ein Polygon-Feature mit dem Tag *highway* deklariert, so stellen diese Plätze oder die Fläche von Fußgängerzonen (*highway = pedestrian area = yes*) dar.

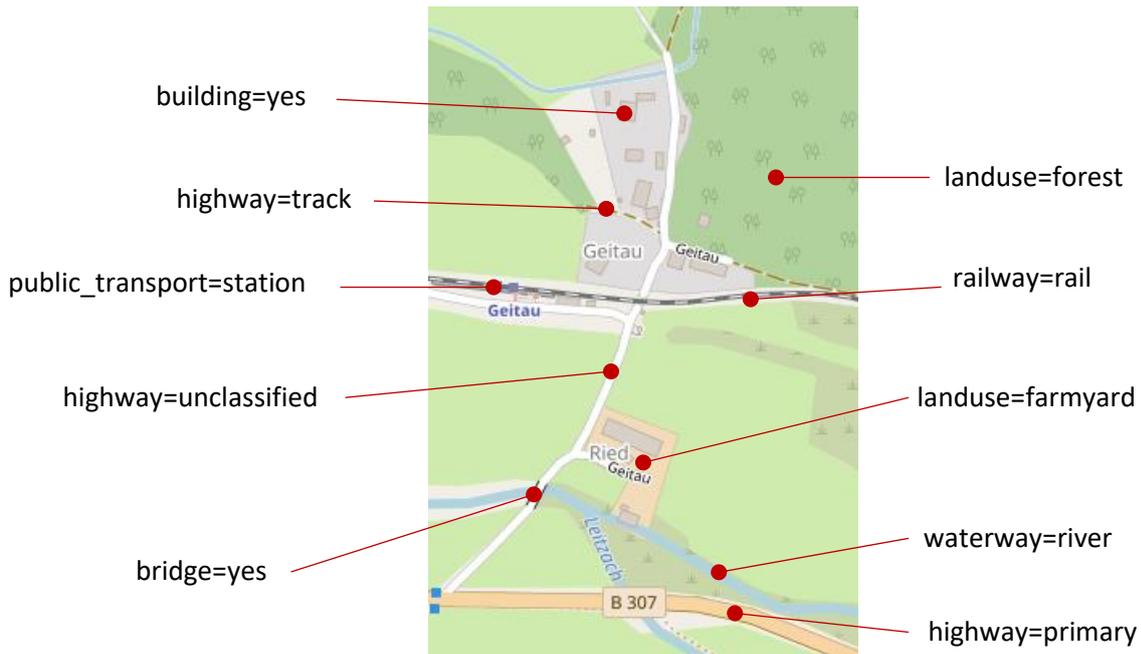


Abb. 8: OSM-Kartenausschnitt mit Beispielen an Key-Value-Paaren auf Map-Features

Straßen können mit dem Schlüssel *highway* identifiziert werden. Dieser gibt an, welche Art von Straße modelliert werden soll. Weitere Tags charakterisieren die Straße zusätzlich, wie zum Beispiel der Name der Straße (Tag *name*) und Anzahl der Fahrspuren (Tag *lanes*).

Die OpenStreetMap-Daten können auch für Routingzwecke verwendet werden. Daher muss das Straßennetz eine geeignete Diagrammstruktur aufweisen. Die Konnektivität zeigt sich besonders in Kreuzungsbereichen. Treffen zwei Straßen aufeinander, so müssen sich die entsprechenden Kanten an dieser Stelle einen Knoten teilen, nur dann sind die beiden Straßen als verbunden zu betrachten. Befindet sich an dieser Stelle zum Beispiel eine Brücke oder ein Tunnel, so kreuzen sich zwar die überlagernden Straßen, durch die unterschiedliche Straßenebene erhalten sie aber keinen gemeinsamen Knoten. Somit besteht nicht die Möglichkeit von einer Straße auf die andere zu gelangen. Straßen werden generell als eine Kante erfasst mit zwei Fahrrichtungen, sofern keine Einbahnstraße vorliegt. Eine Ausnahme bilden zum Beispiel Autobahnen. Mehrspurige Straßen, deren

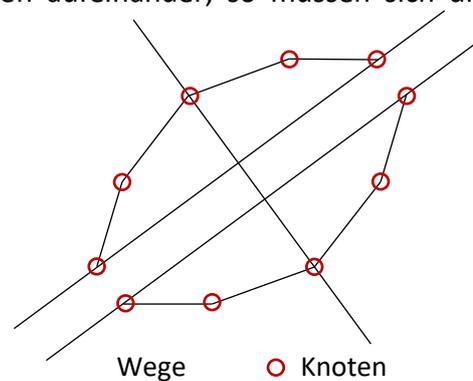


Abb. 9: Konnektivität am Beispiel vereinfachter Autobahnausfahrten

Fahrs Spuren durch eine physikalische Barriere voneinander getrennt sind, erhalten eine Kante je Fahr richtung mit entsprechender Einbahnregelung. Eine Autobahn mit vier Fahrs Spuren, je zwei für jede Richtung, würde als zwei Kanten dargestellt werden, die jeweils nur in eine Richtung befahrbar sind. Wege werden als geordnete Liste von Knoten gespeichert, sodass die Datenbasis in der Graphterminologie aus gerichteten Kanten besteht. Ein Straßensegment zwischen zwei Knoten weist grundsätzlich zwei Richtungen von Knoten A nach Knoten B und umgekehrt von Knoten B zu Knoten A auf. Eine Ausnahme bilden zum Beispiel, wie bereits beschrieben Autobahnen und Einbahnstraßen. Letztere erhalten zusätzlich den Tag *oneway* = *yes*. Die Straßenkategorie *highway* = *motorway* impliziert *oneway* = *yes*, erhält diesen Tag aber nicht zusätzlich.

2.3.2 Beschreibung des Straßendatensatzes

Das OpenStreetMap-Datenmodell wird in einem spezifischen XML-Datenformat mit der Datei-Endung *.osm* bereitgestellt. OSM-Daten sind so konzipiert, dass sie einfach über das Internet gesendet und empfangen werden können, daher sind *.osm*-Dateien in XML codiert. Die enthaltenen geografischen Daten liegen somit in einem strukturierten, geordneten Format vor. Die OSM-XML-Datei beginnt mit einem `<osm>`-Element, gefolgt von einem `<bounds>`-Element, das die maximale Ausdehnung der Daten angibt. Das Hauptelement ist eine sortierte Liste mit OSM-Objekten, die in Blöcken angeordnet ist. Es beginnt mit den Nodes, danach die Ways und anschließend die Relationen. Das Attribut *version* wird an unterschiedlichen Stellen angegeben. Im OSM-Tag weist dieses Attribut auf die Version der Datenbank-API, im Objekt selbst gibt es die laufende Versionsnummer an. Unter *generator* findet sich der Name des Programms, das die Datei generiert hat.

Beispiel XML für eine *.osm*-Datei:

```
<?xml version='1.0' encoding='UTF-8'?>
<osm version="0.6" generator="osmconvert 0.8.11" timestamp="2023-01-16T01:00:02Z">
  <bounds minlat="48.539" minlon="12.178" maxlat="48.542" maxlon="12.186"/>
  <node id="8109257501" lat="48.5396125" lon="12.1706098" version="1">
    <tag k="name" v="Carossahöhe"/>
    <tag k="natural" v="peak"/>
    <tag k="ele" v="474"/>
  </node>
  ...
  <way id="38985134" version="1">
    <nd ref="463831491"/>
    <tag k="highway" v="residential"/>
    <tag k="maxspeed" v="30"/>
    <tag k="name" v="Am Steinberg"/>
  </way>
  ...
  <relation id="5996400" version="1">
    <member type="node" ref="4000222714" role="platform"/>
    <tag k="name" v="Am Lehel"/>
    <tag k="public_transport" v="platform"/>
    <tag k="type" v="public_transport"/>
  </relation>
  ...
</osm>
```

Durch die fortlaufende Digitalisierung durch die OSM-Community wächst der Geodatenbestand des OpenStreetMap-Projekts kontinuierlich an. Die Datenbank mit allen Rohdaten ist in einer Datei namens Planet.osm frei verfügbar und modelliert somit die gesamte Erde. Je nach Region, können die OSM-Rohdaten viel Speicherplatz benötigen, daher sind sie in komprimierter Form mit den Endungen .bz2 oder .pbf (Protocol Buffer Binary Format) erhältlich. Dateien mit .pbf-Komprimierung können schneller eingelesen werden, zur Verarbeitung werden aber spezifische Werkzeuge benötigt, wie OSM-Editoren. Die .pbf-Komprimierung erlaubt Dateien blockweise zu decodieren. In der Standardkonfiguration enthält jeder Block etwa achttausend Einträge (OpenStreetMap Wiki contributors 2023). Die in dieser Arbeit verarbeiteten Daten wurden auf der Webseite der Firma Geofabrik⁴ kostenfrei heruntergeladen, wobei die Auswahl auf das Bundesland Bayern eingeschränkt wurde. Es besteht auch die Möglichkeit die Daten im Shapefile-Format zu erhalten, allerdings wird nur ein Standardsatz an Attributen mitgeliefert. So fehlt zum Beispiel das Attribut *construction*, das für die weitere Bearbeitung der Daten relevant ist.

Tab. 6: Anzahl der OSM-Elemente in Bayern nach Regierungsbezirk zum Stand: 29.01.2023

Regierungs- bezirk	Knoten		Wege		Relationen
	Anzahl	davon key:highway	Anzahl	davon key:highway	Anzahl
Oberbayern	19.072.706	77.992	2.875.683	817.968	39.657
Niederbayern	8.610.377	19.025	1.216.514	294.210	13.642
Oberpfalz	7.540.805	18.024	1.110.203	301.720	12.566
Oberfranken	8.801.281	34.560	1.049.576	275.634	16.026
Mittelfranken	7.496.202	25.452	1.284.274	306.839	16.276
Unterfranken	9.481.332	23.258	1.265.720	319.282	16.776
Schwaben	10.455.321	26.422	1.672.597	513.614	17.741
Bayern	71.458.024	224.733	10.474.567	2.829.267	132.684

Für die Entwicklung eines Geschwindigkeitsprofils für Rettungswägen sind jene Elemente relevant, die den Schlüssel *highway* aufweisen. Dieser findet sich sowohl bei Wegen als auch bei Knoten. Für die Beurteilung der Straßensegmente hinsichtlich ihrer Eignung für die Befahrung durch Rettungswägen, sind lediglich Wege mit dem Schlüssel *highway* als relevant zu betrachten. Nachfolgende Tabelle listet die möglichen Ausprägungen des Tags, das den Schlüssel *highway* enthält, aus dem OSM-Datensatz für Bayern auf und gibt die Anzahl der entsprechenden Attribute an.

⁴ <https://download.geofabrik.de/europe/germany.html> (Stand: 29.01.2023)

Tab. 7: Ausprägungen des Wege-Elements im OSM-Datensatz für Bayern zum Stand 29.01.2023 (OpenStreetMap Wiki contributors 2023)

Element	Wert	Anzahl	OSM-Bedeutung
way	abandoned	5	stillgelegtes Straßenobjekt, das stark verfallen ist
way	bridleway	477	Reitweg
way	bus_guideway	7	Strecke für Spurbus
way	bus_stop	457	Bushaltestelle
way	busway	1	nur für die Personenbeförderung bestimmte Straße
way	chain_up_area	1	Bereich zum Anlegen von Schneeketten
way	construction	1.436	Straße befindet sich im Bau; Typ der Straße findet sich im Tag construction=*
way	corridor	1.510	Gang/Flur/Korridor innerhalb eines Gebäudes (Indoor Mapping)
way	crossing	19	Fußgängerüberweg
way	cycleway	18.819	Fahrradweg
way	demolished	10	vollständig zurückgebaute Straße
way	disused	37	unbenutztes Straßenobjekt, das sich aber noch in angemessenem Zustand befindet
way	elevator	189	öffentlicher Aufzug
way	emergency_bay	11	Nothaltebucht neben einer Straße
way	escape	3	Notfallspur
way	footway	257.780	ausgewiesener Gehweg
way	living_street	18.013	Spielstraße, verkehrsberuhigter Bereich
way	motorway	13.075	Autobahn; mindestens 2 Fahrspuren, die baulich voneinander getrennt sind; nur in eine Richtung befahrbar
way	motorway_link	7.422	Autobahnauffahrt; nur in eine Richtung befahrbar
way	no	27	kein (physischer) Weg, für Objekte, bei welchen das nicht offensichtlich ist
way	passing_place	5	Ausweichstelle auf einer einspurigen Straße
way	path	281.700	allgemeiner Weg für den nicht motorisierten Verkehr
way	pedestrian	5.702	Fußgängerzone
way	platform	1.830	Bahnsteig mit erhöhter Konstruktion am Straßenrand
way	primary	30.286	Bundesstraße
way	primary_link	3.700	Verbindungsstraße zwischen Bundesstraßen oder zwischen Bundesstraße und untergeordneter Straße
way	proposed	1.766	Straße die sich in Planung befindet
way	raceway	279	Rennstrecke
way	razed	6	vollständig entfernte Straße

Element	Wert	Anzahl	OSM-Bedeutung
way	residential	356.803	Gemeindestraße innerhalb von Ortschaften als Zugangswege zu Gebäude
way	rest_area	401	Autobahnparkplatz oder Rastplatz
way	road	299	unbekannte Klassifikation als vorübergehende Kennzeichnung als Straße
way	secondary	71.489	Landstraße
way	secondary_link	2.277	Verbindungsstraße zwischen Landstraßen oder zwischen Landstraße und untergeordneter Straße
way	service	704.961	Verbindungsstraße zu oder in Industriegebieten, Parkplätzen, Passagen, etc.
way	services	122	Raststätte an einer Autobahn
way	steps	39.095	Treppe
way	tertiary	79.200	Kreisstraße
way	tertiary_link	1.041	Verbindungsstraße zwischen Kreisstraßen oder zwischen Kreisstraße und untergeordneter Straße
way	track	813.017	Wirtschafts-, Feld- oder Waldweg
way	trail	1	allgemeiner Weg für den nicht motorisierten Verkehr
way	trunk	5.799	Schnellstraße
way	trunk_link	3.810	Auffahrt auf eine Schnellstraße; nur in eine Richtung befahrbar
way	turning_circle	2	Wendestelle
way	unclassified	106.310	Gemeindestraße
way	via_ferrata	65	Klettersteig
way	yes	2	unbekannte Klassifikation als vorübergehende Kennzeichnung als Straße

Aus der Vielzahl an Attributen, die unter dem Schlüssel *highway* auftreten, müssen die Wege identifiziert werden, die für das Routing mit Rettungswagen als geeignet zu betrachten sind:

Tab. 8: Identifikation der befahrbaren Straßenkategorien (OpenStreetMap Wiki contributors 2023)

Straßen-Hierarchie	Wert - Beschreibung der Straßenkategorie
Hauptstraßen	motorway - Autobahn; mindestens 2 Fahrspuren, die baulich voneinander getrennt sind; nur in eine Richtung befahrbar
	motorway_link - Autobahnauffahrt; nur in eine Richtung befahrbar
	trunk - Schnellstraße
	trunk_link - Auffahrt auf eine Schnellstraße; nur in eine Richtung befahrbar
	primary - Bundesstraße
	primary_link - Verbindungsstraße zwischen Bundesstraßen oder zwischen Bundesstraße und untergeordneter Straße
	secondary - Landstraße
	secondary_link - Verbindungsstraße zwischen Landstraßen oder zwischen Landstraße und untergeordneter Straße
	tertiary - Kreisstraße
	tertiary_link - Verbindungsstraße zwischen Kreisstraßen oder zwischen Kreisstraße und untergeordneter Straße
Nebenstraßen	unclassified - Gemeindestraße
	residential - Gemeindestraße innerhalb von Ortschaften als Zugangswege zu Gebäude
	living_street - Spielstraße
	pedestrian - Fußgängerzone
Untergeordnete Straßen	service - Verbindungsstraße zu oder in Industriegebieten, Parkplätzen, Passagen, etc.
	track - Wirtschafts-, Feld- oder Waldweg

Neben dem Schlüssel *highway* zur Identifikation der Straßenkategorien sind weitere Attribute relevant für die Routingfähigkeit des Straßennetzes:

Bürcken: *bridge* = *yes*

Tunnel: *tunnel* = *yes*

Höchstgeschwindigkeit: *maxspeed* = <Nummer>

Einbahnstraße: *oneway* = *yes*

Kreisverkehr: *junction* = *roundabout* oder *highway* = *mini_roundabout*

Baustelle: *highway* = *construction* und *construction* = <Straßenkategorie>

Da nicht sichergestellt sein kann, dass für jedes Weg-Element die Information zur Höchstgeschwindigkeit im Attribut *maxspeed* eingetragen ist, werden weitere Tags herangezogen. Die Angabe der Höchstgeschwindigkeit kann in den Schlüsseln *zone:traffic*, *zone:maxspeed* und *source:maxspeed* vermerkt sein, ohne dass *maxspeed* vorhanden ist. Da es von Land zu Land unterschiedliche Standardregelungen für den Verkehr gibt, sind die angegebenen Werte länderspezifisch. In Deutschland wird hier zwischen Innerorts (*:urban*)

und außerhalb geschlossener Ortschaften (:*rural*) sowie Richtgeschwindigkeit auf der Autobahn (:*motorway*) differenziert. Die nachfolgenden Tabellen mit der Übersicht der jeweiligen Ausprägungen wurden aus dem bayrischen OSM-Datensatz zum Stand 29.01.2023 extrahiert.

zone:traffic:

Gibt die impliziten Standardverkehrsregeln für eine Straße an, die nicht explizit durch Schilder festgelegt sind.

Tab. 9: Höchstgeschwindigkeiten aus dem Tag zone:traffic

Tag	Höchstgeschwindigkeit	Ausprägung
zone:traffic	-	DE:motorway
zone:traffic	100km/h	DE:rural
zone:traffic	50km/h	DE:urban
zone:traffic		urban
zone:traffic		AT:urban
zone:traffic	30km/h	DE:speed:30
zone:traffic		DE:30
zone:traffic		30

zone:maxspeed:

Ist ein Tempolimit nicht an eine Straße, sondern an einen Bereich gebunden, so erhält der Weg diesen Tag.

Tab. 10: Höchstgeschwindigkeiten aus dem Tag zone:maxspeed

Tag	Höchstgeschwindigkeit	Ausprägung
zone:maxspeed	-	DE:motorway
zone:maxspeed	100km/h	DE:rural
zone:maxspeed	50km/h	DE:urban
zone:maxspeed		DE:50
zone:maxspeed		AT:50
zone:maxspeed	40km/h	DE:40
zone:maxspeed	30km/h	DE:30
zone:maxspeed		de:30
zone:maxspeed		30
zone:maxspeed		AT:30
zone:maxspeed	20km/h	DE:20
zone:maxspeed	10km/h	DE:10

source:maxspeed:

Dieses Attribut gibt die Quelle der Information zur Höchstgeschwindigkeit an, wie zum Beispiel 'sign DE:274-30' für ein Verkehrszeichen „30 km/h“, daher sollte auch das Attribut *maxspeed* vorhanden sein. Für den Fall, dass lediglich *source:maxspeed* einen Wert aufweist, *maxspeed* hingegen nicht vorkommt, kann es als Angabe von zonalen Geschwindigkeitsbegrenzungen herangezogen werden.

Tab. 11: Höchstgeschwindigkeiten aus dem Tag *source:maxspeed*

Tag	Höchstgeschwindigkeit	Ausprägung
source:maxspeed	-	DE:motorway
source:maxspeed		sign;DE:motorway
source:maxspeed	130km/h	130
source:maxspeed		AT:motorway
source:maxspeed	100km/h	DE:rural
source:maxspeed		de:rural
source:maxspeed		DE:rural;sign
source:maxspeed		sign;DE:rural
source:maxspeed		rural
source:maxspeed		AT:rural
source:maxspeed	90km/h	CZ:rural
source:maxspeed	80km/h	80
source:maxspeed	50km/h	50
source:maxspeed		DE:urban
source:maxspeed		de:urban
source:maxspeed		DE:urban;sign
source:maxspeed		DE.urban
source:maxspeed		de_urban
source:maxspeed		sign;DE:urban
source:maxspeed		source:maxspeed=DE:urban
source:maxspeed		urban
source:maxspeed		AT:urban
source:maxspeed		CZ:urban
source:maxspeed	40km/h	AT:zone40
source:maxspeed	30km/h	DE:zone:30
source:maxspeed		DE:zone:30;DE:bicycle_road

Tag	Höchstgeschwindigkeit	Ausprägung
source:maxspeed		DE:zone30
source:maxspeed		de:zone30
source:maxspeed		sign:zone30
source:maxspeed		zone30
source:maxspeed		DE:30zone
source:maxspeed		DE:30
source:maxspeed		DE:zone
source:maxspeed		zone
source:maxspeed		DE:274[30]
source:maxspeed		DE:274-30
source:maxspeed		sign DE:274-30
source:maxspeed		DE:bicycle_road
source:maxspeed		DE:bicycle_zone
source:maxspeed		DE:Fahrradstraße
source:maxspeed		AT:zone:30
source:maxspeed		AT:zone30
source:maxspeed		DE:zone:20
source:maxspeed	20km/h	DE:zone20
source:maxspeed		CZ:living_street
source:maxspeed		DE:zone:10
source:maxspeed	10km/h	DE:zone10

2.4 Weitere Daten

2.4.1 Administrative Grenzen

Für die Verarbeitung räumlicher Daten werden die administrativen Grenzen in Bayern herangezogen. Diese werden kostenfrei vom Landesamt für Digitalisierung, Breitband und Vermessung zur Verfügung gestellt. Die Grenzverläufe aller bayrischen Verwaltungsgebiete werden aus den Daten des Amtlichen Liegenschaftskatasterinformationssystem (ALKIS®), unter monatlicher Aktualisierung, abgeleitet und unter der Lizenz CC BY 4.0 zur freien Nutzung bereitgestellt⁵. Die Ausgabedatei ist das Shapefile *VerwaltungsEinheit.shp* im projizierten Koordinatensystem ETRS 1989 UTM Zone 32N (EPSG:25832) verortet. Da die OpenStreetMap-Daten und die Standortdaten der Krankenhäuser im geodätisches Referenzsystem WGS84

⁵ <https://geodaten.bayern.de/opengeodata/OpenDataDetail.html?pn=verwaltung> (abgerufen am 15.02.2023)

(EPSG: 4326) vorliegen müssen die Datensätze umprojiziert werden. Dies erfolgt in ArcGIS Pro über die Datumstransformationsmethode ETRS_1989_To_WGS_1984.

Das Shapefile enthält flächenhafte Geometrien, die die Grenzverläufe der Gemeinde-, Landkreis-, Regierungsbezirksgrenzen und die bayrische Landesgrenze repräsentieren. Um die einzelnen Verwaltungseinheiten zu unterscheiden, wird ein amtlicher Gemeindegeschlüssel AGS mit folgender Systematik am Beispiel des AGS 09171137038 angegeben:

- 09** 09 = Bundesland Bayern
- 1** 1-7 = Regierungsbezirk
- 7** 6 = kreisfreie Stadt; 7 - 9 = Landkreis
- 1** 1,2 = fortlaufende Nummer
- 1** 1,2 = Gemeinde; 4 = gemeindefreies Gebiet
- 37** fortlaufende Nummer
- 038** fortlaufende Nummer: Gemeindeteil mit Hauptgemeindeteil = 001

Tab. 12: Struktur und Dateninhalte der ALKIS® Verwaltungsgebiete

Spaltenname	Datentyp	Beschreibung	Beispiel
oid	Text	Objektidentifikator	DE091710112112
aktualit	Text	Beginn Lebenszeitintervall in JJJJ-MM-TT	2023-01-26Z
art	Text	Art der Verwaltungseinheit	Bundesland; Regierungsbezirk; Kreis / kreisfreie Stadt; Gemeinde
name	Text	Name der Verwaltungseinheit	Burghausen
rs	Text	Regionalschlüssel: {LL}{R}{KK}{VVVV}{GGG} L = Land (09 für Bayern) R = Regierungsbezirk (1 - 7) K = Landkreis / kreisfreie Stadt V = Verwaltungsgemeinschaft G = Gemeinde	Kreis / kreisfreie Stadt: 09171 Verwaltungsgemeinschaft: 091710112 Gemeinde: 091710112112
ags	Text	Amtlicher Gemeindegeschlüssel: {LL}{R}{KK}{GGG}{TTT} L = Land (09 für Bayern) R = Regierungsbezirk (1 - 7) K = Landkreis / kreisfreie Stadt G = Gemeinde T = Gemeindeteil	Kreis / kreisfreie Stadt: 09171 Gemeinde 09171137 Gemeindeteil: 09171137038
uebojekt	Text	OID des nächsten übergeordneten Objekts	DE09171
uebname	Text	Name des nächsten übergeordneten Objekts	Landkreis Altötting

Gemeindeteile sind nicht als Flächen erhältlich, jedoch die Koordinaten des Mittelpunktes des Hauptortes eines Gemeindeteils. Diese werden dem INM vom Bayerischen Landesamt für Statistik jährlich zur Verfügung gestellt. Als weitere Betrachtungsebene werden die Gemeinden herangezogen, jedoch werden für diese nicht die Schwerpunkte der Geometrien als Referenzpunkt ausgewählt, sondern die Koordinaten des Hauptortes der Gemeinde ({TTT} = ,001').

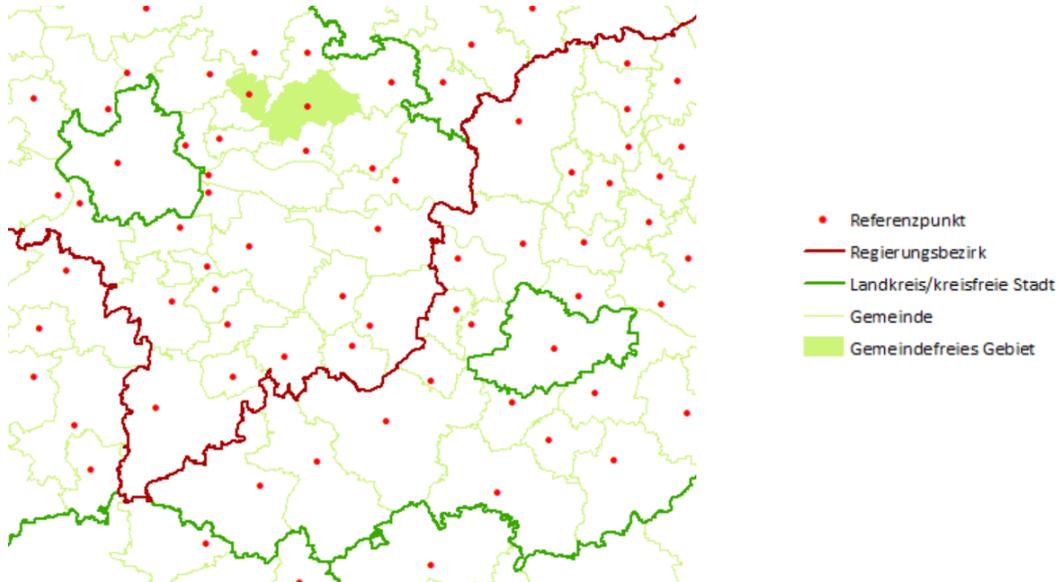


Abb. 10: Ausschnitt der administrativen Grenzen und Referenzpunkte

2.4.2 Bevölkerungsdaten

Um Aussagen über die Versorgung der Bevölkerung treffen zu können, werden die Einwohnerzahlen der Gemeinden, Kreise und Regierungsbezirke aus dem statistischen Bericht⁶ (A1200C 202243) herangezogen. Diese stellen eine Fortschreibung der auf dem Zensus vom 09. Mai 2011 beruhenden Bevölkerungszahlen dar. Die Auswertungen beschränken sich auf die Ebene der Gemeinden, einschließlich kreisfreier Städte, da gemeindefreie Gebiete keine Personen mit Hauptwohnsitz aufweisen. Aus dem Shapefile der Gemeinden kann die Fläche der Polygone, umgerechnet von m² in km², erhalten werden, und daraus die Bevölkerungsdichte je Gemeinde/kreisfreie Stadt berechnet werden.

Tab. 13: Datentypen der Bevölkerungstabelle BASIS_BEV

Spaltenname	Datentyp	PrimaryKey	Beschreibung
ID_GEM	text	X	Amtlicher Gemeindeschlüssel; 8-stellig ohne fortlaufende Nummer der Gemeindeteile
NAME	text		Name der Gemeinde, bzw. kreisfreien Stadt
EW	integer		Einwohnerzahl
GEM_AREA_QKM	float		Gemeindefläche in km ²
BEV_P_AREA	float		Einwohner je Quadratkilometer der Gemeinde/kreisfreie Stadt

⁶ https://www.statistik.bayern.de/statistik/gebiet_bevoelkerung/bevoelkerungsstand (Stand: 3. Vierteljahr 2022)

Beispieldatensatz:

ID_GEM = 09171114
NAME = Emmerting
EW = 4.164
GEM_AREA_QKM = 31,658019
BEV_P_AREA = 131,530655

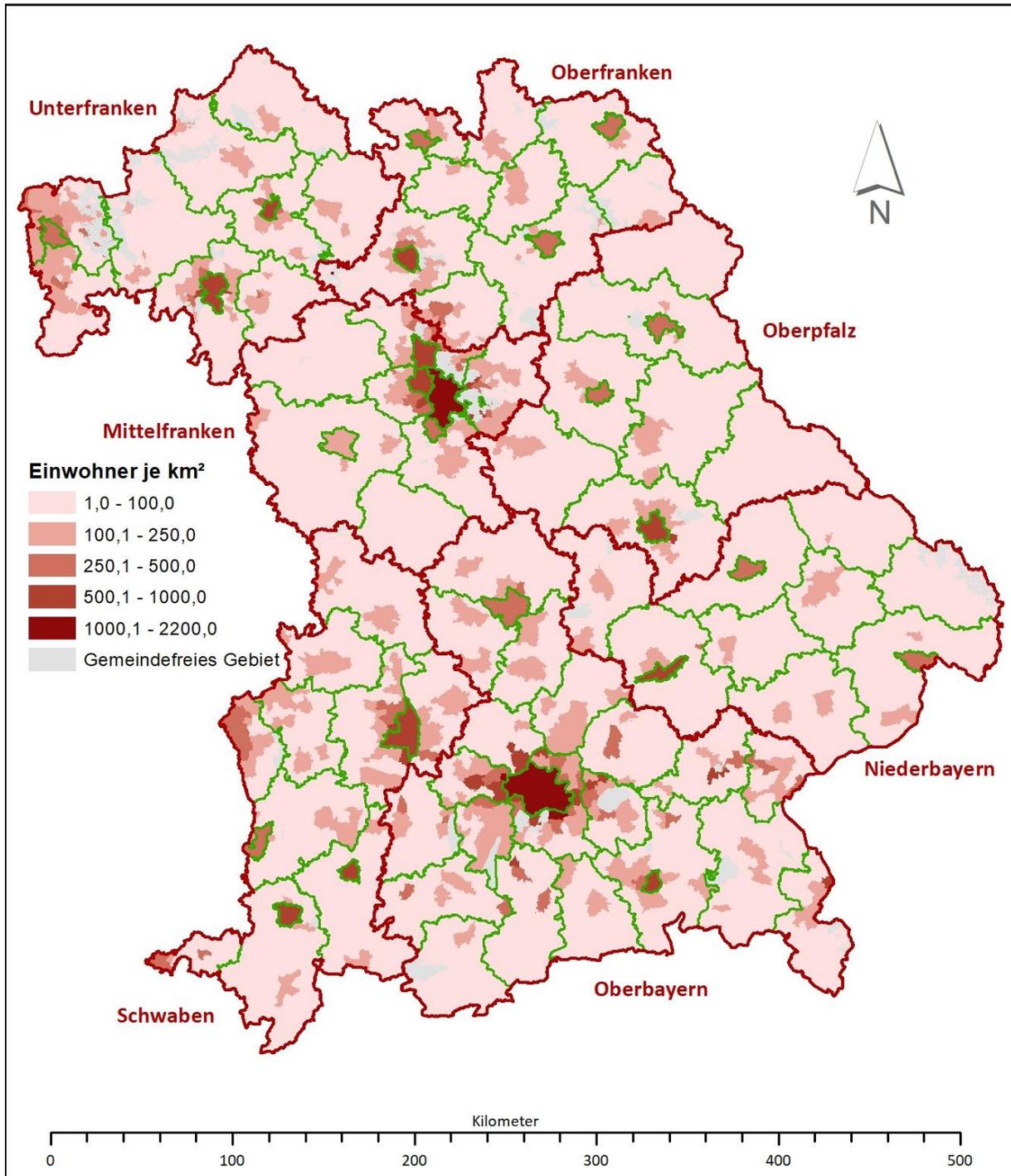


Abb. 11: Verteilung der Bevölkerung auf Gemeindeebene

3 Entwicklung eines Geschwindigkeitsprofils für Rettungswägen

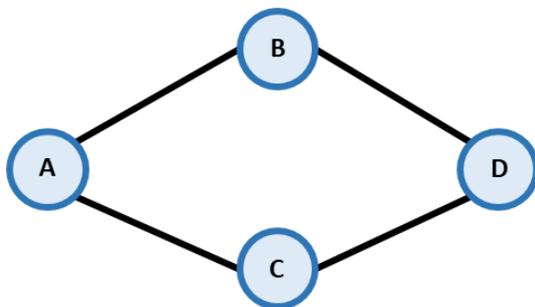
3.1 Routing-Basis

3.1.1 Routing-Theorie

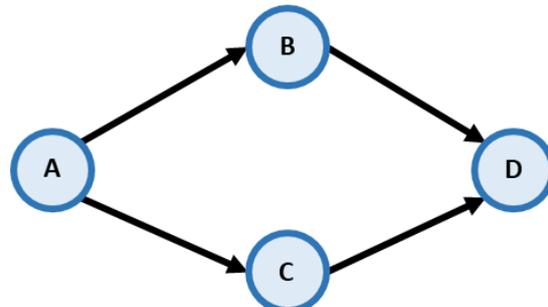
A) Straßennetzwerk im Graphenmodell

Der gerichtete Graph $G = (V, E, w)$ besteht aus einer endlichen Menge an Knoten V (vertices) die durch Kanten E (edges) miteinander verbunden sind. Dabei gilt, dass ein Knoten mit mindestens einer Kante verknüpft sein muss, eine Kante aber mindestens zwei Knoten aufweisen muss. Mit den Kanten sind nicht-negative Kosten, bzw. Gewichtungen w verbunden. Transponiert auf ein routingfähiges Straßennetzwerk werden die Knoten durch Straßenabschnitte, was den Kanten entspricht, miteinander verbunden. Ein Knoten, an dem eine verbundene Kante in eine andere verbundene Kante übergeht, stellt in einem Straßendiagramm eine Kreuzung dar, an der sich zwei Straßensegmente schneiden. Die Kosten können die Weglänge sein, oder die benötigte Fahrzeit entlang des Straßenabschnitts. Dabei ist die topologische Konsistenz obligatorisch, sodass Knoten, die an Kanten anschließen, die genau gleichen Koordinaten aufweisen, wie das Ende der relevanten Kante. Zudem müssen Knoten und Kanten eindeutig identifizierbar sein. Eine

Bei dem Shortest-Path-Problem ist ein Graph gegeben, ein Startpunkt $s \in V$ und ein Zielpunkt $t \in V$, wobei die Länge des kürzesten Weges zwischen s und t im Graphen berechnet werden muss, also die Distanz $dist(s,t)$. Der Pfad $p_{s,t}$ des kürzesten Weges mit der Länge k ist eine Folge $p_{s,t} = v_0, v_1, v_2, \dots, v_k$ von Knoten. Über die geringste Summe der Gewichte aller Kanten es Pfades $p_{s,t}$ ergibt sich der kürzeste Pfad von s nach t .



Ungerichteter Graph



Gerichteter Graph

Dabei finden sich verschiedene Variationen des Problems. Bei dem Single-Source-Shortest-Path-Problem (SSSP) muss die Distanz von einem gegebenen Startpunkt ausgehend zu allen anderen Knoten des Graphen bestimmt werden, wie es bei der Wegesuche in der Navigation zu finden ist. Dem entgegengesetzt ist das Single-Destination-Shortest-Path-Problem (SDSP), also die Berechnung der Distanz von allen Knoten ausgehend zu einem Zielknoten. Ist eine Menge von Startpunkten S und eine Menge von Zielpunkten T gegeben, wobei gilt $s \in S$ und $t \in T$ sowie $S = T = V$, wird die kürzeste Distanz $dist_{min}(s,t)$ zwischen allen Quell- und Zielknoten berechnet (Bast et al. 2016). Dies wird im All-Pairs-Shortest-Path-Problem (APSP) beschrieben.

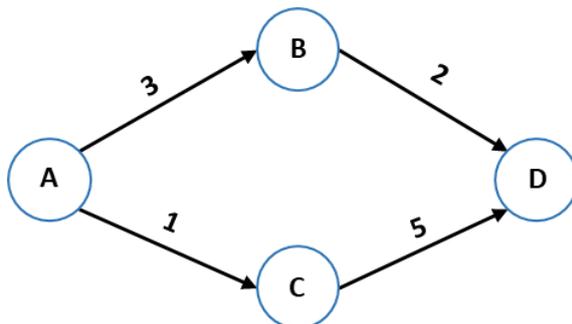
B) Routing-Algorithmen

Routenanfragen, die sich auf Straßennetzwerke beziehen, die einen größeren Maßstab wie einen ganzen Kontinent umfassen, sind sehr umfangreich und rechenintensiv. Um die Rechenzeit effektiver zu nutzen, wurden Algorithmen entwickelt, die optimale Routen in kürzester Zeit berechnen. Obwohl sich die Shortest-Path-Algorithmen auf Straßennetze konzentrieren, sind sie auf beliebige Graphen anwendbar. Die unterschiedlichen Routing-Algorithmen bieten unter anderem unterschiedliche Kompromisse in Bezug auf Abfragezeiten, Vorverarbeitungsaufwand, Speicherplatznutzung und Robustheit gegenüber Eingabeänderungen. Hingegen ist die Qualität der Routingergebnisse kein entscheidender Faktor, da für die gängigsten Algorithmen beweisbare exakte Lösungen gefunden werden.

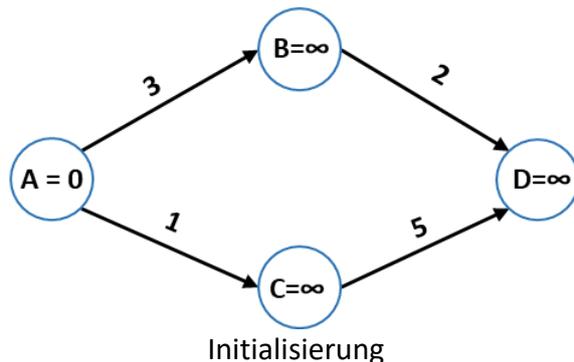
Dijkstra-Algorithmus

Die Standardlösung für das Single-Source-Shortest-Path Problem, mit ausschließlich positiven Kantenkosten, ist der Dijkstra Algorithmus. Dieser hat seinen Anfang in einem Startknoten, von welchem ausgehend schrittweise der jeweils nächste erreichbare Knoten mit den geringsten Kosten einen günstigen Weg bildet (Velden et al. 2014b). Dabei werden alle Knoten des Graphen iterativ durchlaufen, wodurch an der bisher kürzesten Wegstrecke Aktualisierungen vorgenommen werden. Die Basis besteht hier aus einer Prioritätenschlange von Knoten, die nach den Distanzen zum Startknoten geordnet sind. Bei der Initialisierung des Algorithmus weist der Weg zum Startknoten die Kosten 0 auf, da keine Entfernung zu sich selbst besteht. Da der Weg zu alle anderen Knoten noch unbekannt ist werden die Distanzen auf unendlich gesetzt. Beim Durchlaufen des Algorithmus erfahren die Kosten für die Knoten eine Verbesserung, indem vom vordersten Knoten der Warteschlange ausgehend die jeweiligen Nachbarknoten auf deren Kosten geprüft werden. Ist ein Nachbarknoten in der Warteschlange enthalten und weist der Pfad über die entsprechende Kante geringere Kosten auf als die bisherigen gefundenen Kosten, so werden die Kosten des entsprechenden Knotens auf den neuen Wert verringert, was als Relaxation bezeichnet wird. Die Kosten für einen Knoten setzen sich aus der Summe der Distanzen zu den Vorgängerknoten zusammen.

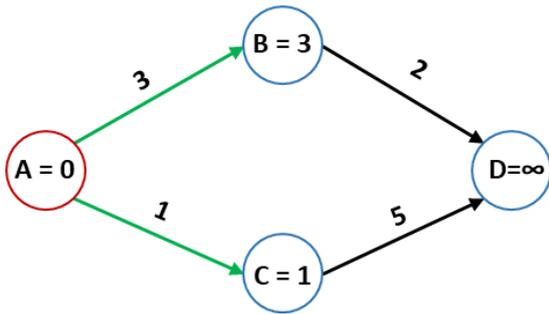
1.



2.

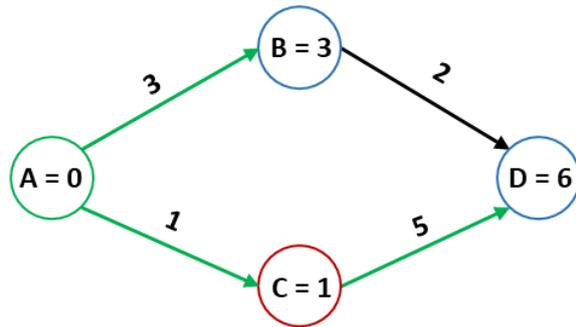


3.



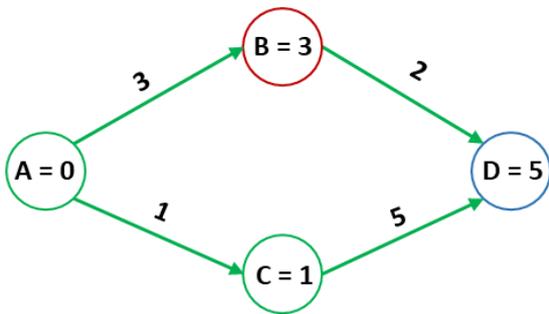
Warteschlange: C, B
In Bearbeitung: A

4.



Warteschlange: B, D
In Bearbeitung: C

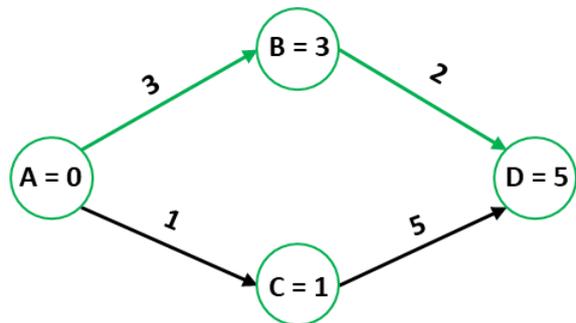
5.



Relaxation von Knoten D

Warteschlange: D
In Bearbeitung: B

6.



Kürzester Pfad von A nach D

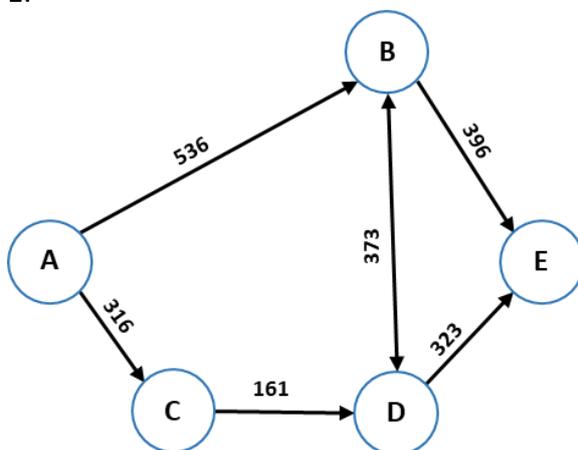
Abb. 12: Schematischer Ablauf des Dijkstra-Algorithmus

A*-Algorithmus

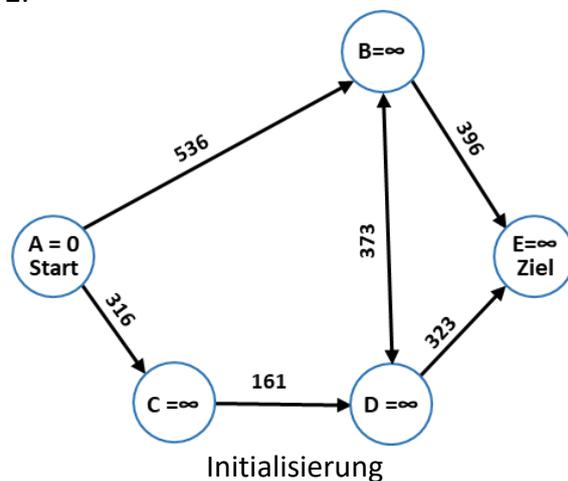
Der A*-Algorithmus ist eine Erweiterung des Dijkstra-Algorithmus, der zunächst nur den kürzesten Pfad zwischen zwei Knoten identifizieren kann. Um das Single-Source-Shortest-Path-Problem zu lösen, muss der Algorithmus entsprechend oft ausgeführt werden. Dabei erfolgt eine zielgerichtete Suche nach den kostengünstigsten Pfad und wird somit der Klasse der informierten Suchverfahren zugeordnet. Hierbei werden die Knoten nicht willkürlich ausgewählt, sondern diesen über eine Schätzfunktion eine ungefähre Entfernung zum Zielknoten zugewiesen (Velden et al. 2014a). Das bringt den Vorteil, dass die Knoten mit der jeweils höchsten Wahrscheinlichkeit ausgewählt werden und somit nicht alle Knoten eines Graphen betrachtet werden müssen. Dadurch läuft der Algorithmus schneller und effizienter durch als der Dijkstra-Algorithmus. Jeder Knoten wird einer Klasse zugeordnet. Wurde ein Knoten noch nicht bearbeitet, ist auch der Weg zu ihm unbekannt, daher spricht man hier von der Klasse der unbekanntenen Knoten. Wie auch im Dijkstra-Algorithmus ist eine Prioritätswarteschlange implementiert mit den Knoten, zu welchen ein Weg bereits ermittelt wurde, von diesem aber nicht ausgegangen werden kann, dass er der kürzeste ist. Die Knoten werden mit einem f-Wert in die Warteschlange aufgenommen. Der f-Wert eines Knotens ist die Summe aus den bisherigen Gesamtkosten vom Startknoten ausgehend und einem Schätzwert, der die geschätzten Kosten vom Knoten zum Zielknoten angibt. Dies stellt die Klasse der bekannten Knoten dar. Sind Knoten fertig bearbeitet, so bilden diese eine eigene Klasse. Ein Knoten ist dann als abgeschlossen zu betrachten, wenn für diesen der kürzeste Weg

vom Startknoten berechnet wurde. Um den f-Wert zu erhalten, wird neben den Kosten vom Startknoten zum aktuellen Knoten eine geschätzte Distanz zum Zielknoten benötigt. Dies erfolgt über eine Schätzfunktion, die nicht vom Algorithmus festgelegt wird, daher ist der Nutzer frei in der Wahl. Dabei ist darauf zu achten, dass die Kosten nicht überschätzt werden dürfen. Wird ein Pfad von Punkt A zu Punkt B gesucht, so bietet sich zum Beispiel die Luftlinie als geeignete Schätzung an, da der Streckenverlauf nicht länger als die euklidische Distanz ausfallen kann und somit eine Überschätzung ausgeschlossen werden kann. Auch bei dem A*-Algorithmus wird der Startknoten während der Initialisierung der Warteschlange mit einem Wert von 0 hinzugefügt, da zu Beginn lediglich ein Abstand zu sich selbst bekannt ist. Alle anderen Knoten werden auf *null* gesetzt und erhalten somit einen unbekanntes Wert. In der Warteschlange findet sich der Knoten mit dem niedrigsten f-Wert immer an erster Stelle, sodass dieser für die weitere Bearbeitung priorisiert werden kann. Ist ein Knoten fertig bearbeitet werden anschließend seine Nachbarknoten betrachtet. Dabei werden drei Fälle unterschieden. Der Nachbarknoten kann bereits abgearbeitet sein, ist er dies nicht, so findet sich entweder ein Eintrag in der Warteliste oder er wurde bisher nicht bewertet. Der f-Wert kann für einen Knoten in der Warteliste aktualisiert werden, sollte dieser über eine neue Kante kürzer berechnet werden. Ein Knoten wird neu in die Warteschlange aufgenommen, indem der entsprechende f-Wert berechnet und in die Liste eingereiht wird. Der Algorithmus wird so lange durchläuft, bis der Zielknoten aus der Warteschlange ausgewählt wird. Wird dies nicht erreicht, obwohl sich kein weiterer Knoten mehr in der Warteschlange befindet, so kann der Zielknoten vom Startknoten ausgehend nicht erreicht werden, also kein möglicher Weg vorhanden ist.

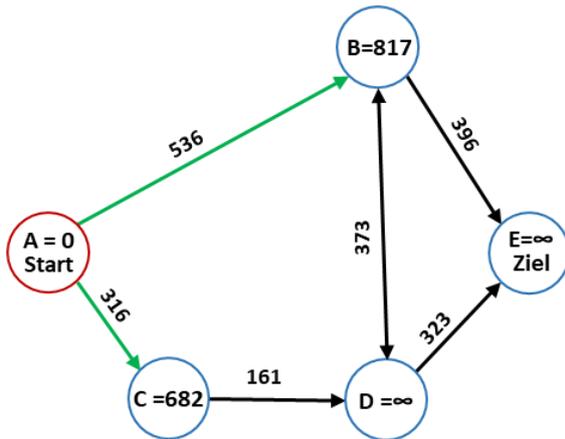
1.



2.

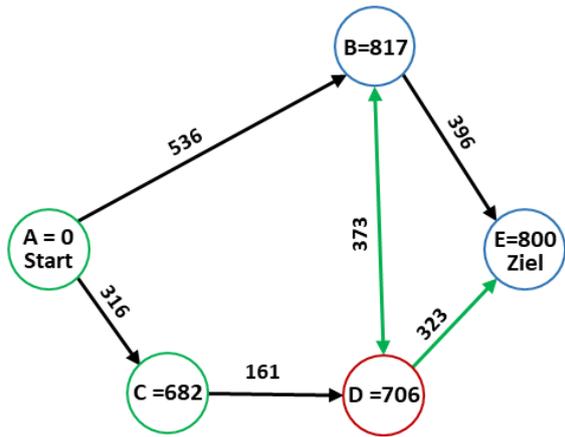


3.



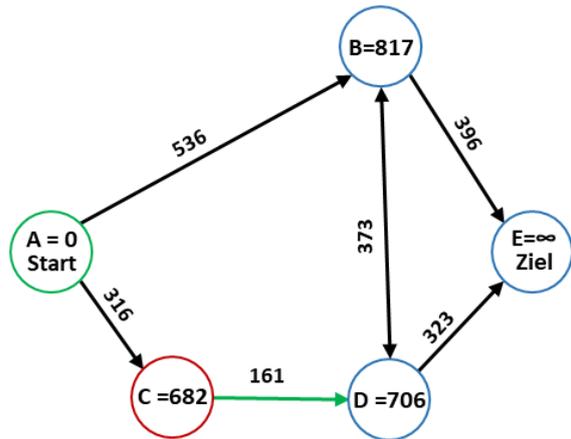
Warteschlange: C, B
 In Bearbeitung: A
 f-Wert B = 536 + Schätzwert 281
 f-Wert C = 316 + Schätzwert 366

5.



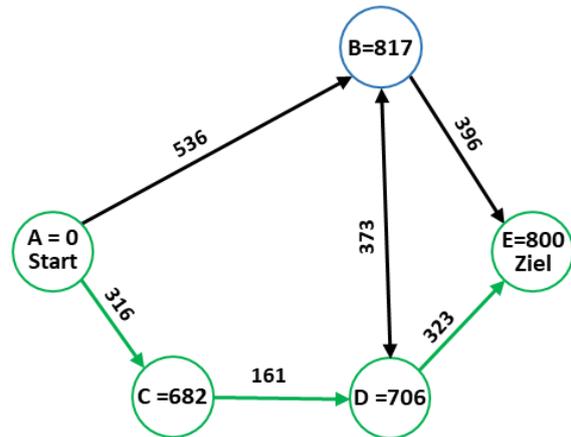
Warteschlange: E, B
 In Bearbeitung: D
 f-Wert E = 316 + 161 + 323 + Schätzwert 0

4.



Warteschlange: D, B
 In Bearbeitung: C
 f-Wert D = 316 + 161 + Schätzwert 229

6.



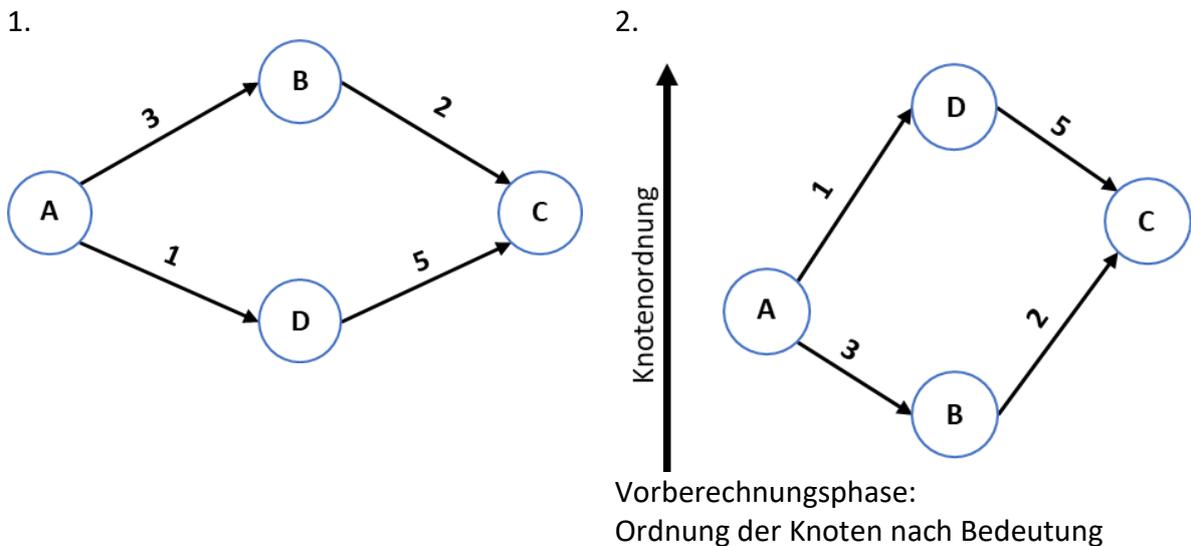
Kürzester Pfad von A nach E

Abb. 13: Schematischer Ablauf des A*-Algorithmus

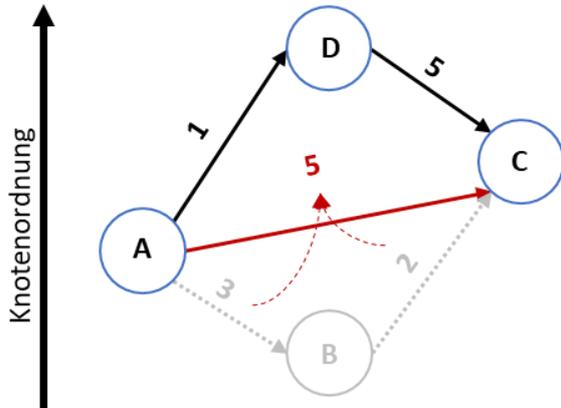
Contraction Hierarchies

Es handelt sich hierbei um ein Konzept der Kantenkontraktion, auf Basis einer hierarchischen Ordnung der Kanten nach ihrer Bedeutung. Die Rangordnung entsteht durch die iterative Zusammenfassung der unbedeutendsten Knoten, wodurch der kürzeste Pfad durch Abkürzungen ersetzt wird (Geisberger et al. 2008). Der Suchalgorithmus der Contraction Hierarchies ist bidirektional, wobei die Vorwärtssuche die Kanten berücksichtigt, die zu höherwertigeren Knoten führen, Kanten in Richtung weniger bedeutender Knoten finden in der Rückwärtssuche Beachtung. Die großen Vorteile diesen Verfahrens sind die fünf Mal schnellere Abfragezeiten, im Vergleich zu Dijkstra-basierten Algorithmen, sowie der niedrigere Speicherplatzverbrauch. Voraussetzung ist ein gewichteter und gerichteter Graph, dessen Knoten unterschiedliche Wichtigkeiten aufweisen. Zudem wird ein augmentierter Graph $G' = (V', E')$ virtuell und den Graphen G überlagernd angelegt, mit einer Menge an Knoten V' und Kanten E' , in welchem die Kantenkontraktion vorgenommen wird und somit

die bewahrten kürzesten Pfaddistanzen des Inputgraphen enthält. Knoten finden sich in einer Prioritätenwarteschlange, sortiert nach der Knotenordnung, also der Attraktivität den Knoten mit den entsprechenden Kanten zu kontrahieren. Welcher Knoten kontrahiert wird ist von verschiedenen Faktoren abhängig, zum einen von der Anzahl der bisher hinzugefügten Abkürzungen und zum anderen der Anzahl der in nächster Umgebung bereits kontrahierten Knoten (Bast et al. 2016). Führt der kürzeste Weg über die Kanten (a, b) und (b, c), wobei gilt, dass a und c Nachbarn von b sind, so ist es ausreichend lediglich den Knoten b, einschließlich der ein- und ausgehenden Kanten zu betrachten. Mit Hilfe des Dijkstra-Algorithmus wird geprüft, ob die Wege (a, b) und (b, c) Teilpfade eines kürzesten Weges sind. Bei der Ausführung des Algorithmus muss der Knoten b ignoriert werden, um herauszufinden, ob der Knoten c von a ausgehend mit den Kosten $cost(a, b) + cost(b, c)$ erreicht werden könnte. Finden sich nur Distanzen mit höheren Kosten, so wird eine Abkürzungskante (a, c) mit den Kosten $cost(a, c) = cost(a, b) + cost(b, c)$ der Menge der Kanten E' hinzugefügt. Dieser Kontraktionsprozess ist so zu betrachten, dass alle möglichen Abkürzungen in der Menge an Kanten E' integriert werden. Ziel ist, dass der verkürzte Graph G' so wenig Kanten wie möglich aufweist. Um schließlich den kürzesten Pfad im Originalgraphen zu rekonstruieren, werden diejenigen Abkürzungskanten rekursiv entpackt, die im augmentierten Graphen G' den kürzesten Weg anzeigen. Dies wird ermöglicht durch die Speicherung des kontrahierten Knotens an der Abkürzungskante.

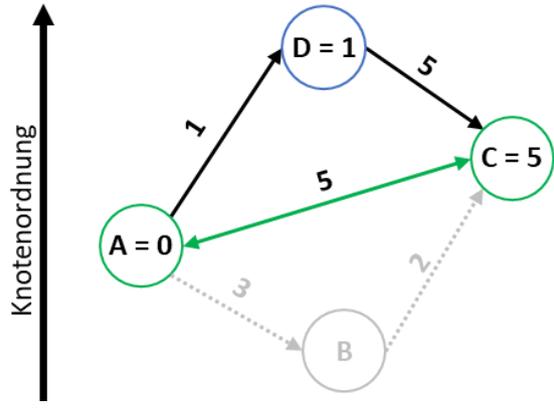


3.



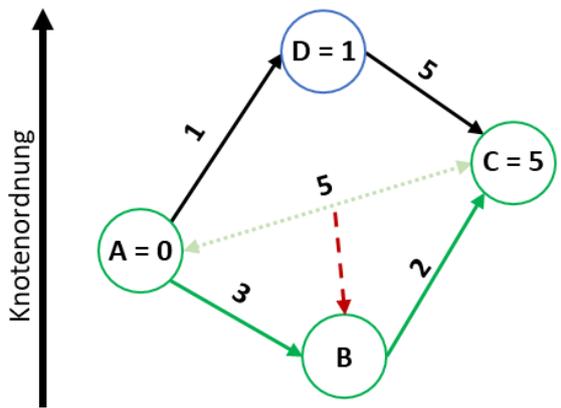
Vorberechnungsphase:
Knotenkontraktion von B
Teilgraph mit Knoten höherer Ordnung als A

4.



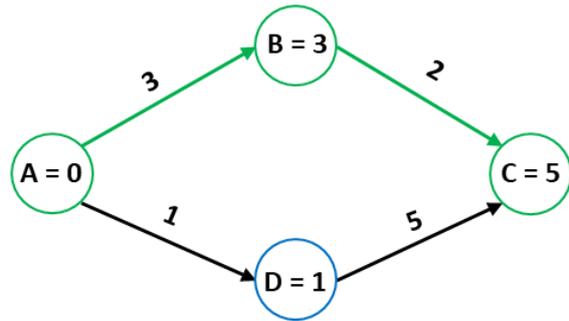
Anfragephase:
Bidirektionale Variante Dijkstra-Algorithmus
Kürzester Pfad = Schnittmenge der Vorwärts- und Rückwärtssuche

5.



Anfragephase:
Rekursives Entpacken der Abkürzungskante

6.



Kürzester Pfad von A nach C über B

Abb. 14: Schematisches Konzept der Contraction Hierarchies

3.1.2 Auswahl der Routing-Engine

Es gibt eine Vielzahl an Routing Services, die auf Basis von OpenStreetMap Daten oder selbst erstellten Daten genutzt werden können. Dabei weisen Open-Source Routing-Engines erhebliche Vorteile gegenüber proprietäre Systeme auf. Man kann flexibler handeln bei der Verwendung gehosteter Dienste, man ist an keinen Hersteller gebunden, sodass eine Installation zu eigenen Bedingungen erfolgen kann, durch einen offen abrufbaren Quellcode können Algorithmen untersucht werden und man unterliegt keinem zu bezahlenden Geschäftsmodell, wodurch die Nutzung kostenfrei ist. Nachfolgend werden drei ausgewählte Routing-Engines beschrieben, die mit zu den bekanntesten gehören.

A) Valhalla

Valhalla ist eine open-source Routing-Engine mit Libraries, um OpenStreetMap-Daten nutzen zu können. Diese weist eine gekachelte hierarchische Datenstruktur auf, wodurch geringere Speicherkapazitäten notwendig sind. Es werden Abbiegerestriktionen, Informationen zu Autobahnausfahrten, administrative Informationen und Zeitzonen zur Verfügung angeboten, sowie unterschiedliche Routingprofile, wie Auto, Fahrrad und Fußgänger. Zudem finden sich Tools zur Bearbeitung von komplexen Routing-Abfragen, wie Zeit- und Distanzmatrizen, Isochronen und Routenoptimierungen, wie bei der Problemstellung des Travelling Salesman, mit der zusätzlichen Möglichkeit zum Offline-Routing. Durch die Plug-In Architektur können die Laufzeitkosten für Kanten und Knoten innerhalb des Graphen dynamisch erfasst werden und so das Anpassen und die Generierung von alternativen Routen ermöglicht werden. Regionale Besonderheiten können eingegeben werden, wie zum Beispiel administrative Informationen zu Grenzübergängen. Über die Erhöhung der Kosten, also ein Zeitaufschlag bei der Bildung des Routenpfades, oder Implementierung einer Strafe, die ebenfalls auf die Kosten angewendet wird, kann ein Länderübergang bei der Pfaderstellung vermieden werden (Valhalla Organization 2023a). Mit der Bereitstellung einer Api in C++ wird Routing auch auf mobilen Offline-Geräten ermöglicht, zudem können alle Angebote, wie Route, Isochronen, Matrix, usw. über die Valhalla-Python-Bibliothek verwendet werden. Valhalla ist in verschiedene Library-Module organisiert, die für unterschiedliche Funktionen verantwortlich sind. Zum Beispiel das Modul *MJOLNIR*, das Open Data in gekachelte Valhalla Graphen umwandelt, das Modul *THOR* (Tiled Hierarchical Open Routing), das einen Pfad mit entsprechender Attribution entlang der Graphen-Hierarchie erzeugt, oder das Modul *ODIN* (Open Directions and Improved Narrative), in welchem Wegbeschreibungen und Richtungsinformationen entlang des Pfades ausgegeben werden (Valhalla Organization 2023c).

Um den günstigsten Pfad zu berechnen, wendet Valhalla unterschiedliche Algorithmen an, die dynamische Laufzeitkosten verwenden (Valhalla Organization 2023b). Damit werden Kosten auf Basis einer Vielzahl von Attributen, die in den Kacheln des Routinggraphen gespeichert sind, generiert. Durch unterschiedliche Kostenberechnungsmethoden und -optionen können verschiedene Arten von Routen oder Routen mit unterschiedlichen Eigenschaften generiert werden. Die verwendeten Algorithmen sind Varianten des Dijkstra Algorithmus. Dabei werden unterschiedliche Klassen aus dem *PathAlgorithmus* abgeleitet. Die Klasse *Unidirectional A** wird für zeitabhängiges Routing verwendet und ist ein Vorwärts- oder Rückwärtsrichtungs-A*-Algorithmus. Nicht zeitabhängige Routen werden mit dem *Bidirectional-A**-Algorithmus berechnet. Bei der Klasse *MultiModal* handelt es sich um ein Vorwärtsrichtungs-A*-Algorithmus mit einer Logik zum Wechsel zwischen Fußgänger- und Transitmodus. Dieser Algorithmus ist aufgrund der Beschaffenheit von Transitfahrplänen zeitabhängig.

Der Straßengraph ist in Kacheln, unterschiedlicher Größe, strukturiert, die sich in drei Hierarchieebenen gliedern (Valhalla Organization 2023d). Ebene 0 enthält die größeren Hauptstraßen, wie Autobahn, Schnellstraßen und Bundesstraßen, und werden in 4-Grad-Kacheln gespeichert, Land- und Kreisstraßen finden sich auf der Hierarchieebene 1 in 1-Grad-Kacheln. In 0,25-Grad-Kacheln werden Nebenstraßen, wie Gemeindestraßen, auf Ebene 2 zusammengefasst.

Die Routingbasis wird aus verschiedenen open source Datensätzen erstellt. Vorrangig werden die Straßeninformationen aus OpenStreetMap genutzt, um ein Routing-Netzwerk zu erstellen.

Länder- und Bundeslandinformationen für Straßen werden aus den Verwaltungspolygonen identifiziert. Durch die Verwendung der Relationen-Informationen können zusätzliche Attribute für die Straßen abgeleitet werden und ermöglicht das Erzeugen von Routing-Einschränkungen. Transitland stellt Valhalla Informationen zu öffentlichen Verkehrsmitteln bereit. Durch eine Verbindung zum Straßen- und Wegenetz von OpenStreetMap wird multimodales Routing, wie eine Wegstrecken-Kombination aus Fußgänger und öffentliche Verkehrsmittel, ermöglicht (Valhalla Organization 2023b). Geländedaten werden genutzt, um höhenbeeinflusste Routen zu erstellen, z.B. für Fahrradrouten. Die verwendeten europäischen Geländedaten (EU-DEM layers) basieren auf Copernicus-Daten und -Informationen. Jeder Knoten, bzw. jede Kreuzung im Straßennetz enthält Zeitoneninformationen. Die hierfür verwendeten Daten enthalten Polygonddefinitionen der verschiedenen Zeitonen der Welt.

B) GraphHopper

GraphHopper ist eine schnelle, speichereffiziente open-source Routing-Engine. Sie ist nutzbar für Server, Desktop und Mobile Geräte, sowie offline (GraphHopper GmbH 2023b). Durch die Apache Lizenz ist es einfach GraphHopper in andere Produkte zu integrieren und ist so eine Business Alternative zu anderen Routing Engines. Dabei kann sie als Java Bibliothek genutzt werden oder als eigenständiger Webserver, um Routingfunktionen durchzuführen. Diese umfassen unter anderem Distanz- und Zeitberechnungen, Isochronenberechnung. Neben GTFS-Daten (General Transit Feed Specification), die Verkehrsdaten von öffentlichen Verkehrsbetrieben zugänglich macht, sind OpenStreetMap-Daten standardmäßig implementiert. Straßentypen, Geschwindigkeitsbegrenzungen, Oberflächen, Barrieren und Zugangsbeschränkungen werden neben weiteren Attributen aus den OpenStreetMap-Daten verarbeitet. Für jedes Straßensegment werden die Geschwindigkeiten verschiedener Fahrzeugprofile gespeichert, wobei für einzelne Straßenabschnitte Prioritäten festgelegt werden können. GraphHopper bietet vordefinierte Routing-Profilen, wie Auto, Fahrrad, Rennrad, Mountainbike, Motorrad, und verschiedene Gewichtungsmöglichkeiten, wie schnellste oder kürzeste Route, wobei individuelle Profil- und Gewichtungsanpassungen vorgenommen werden können. Um das Routingergebnis zu verfeinern, kann auf Höhendaten zugegriffen werden, was Routing basierend auf Höhenunterschieden ermöglicht wird.

Auch können GPS-Tracks verarbeitet werden, indem diese auf Straßensegmente mittels Map Matching übertragen werden (GraphHopper GmbH 2023). Dafür wird der Viterbi-Algorithmus angewendet. Hierbei werden für jede GPS-Position eine Menge an möglichen Kandidaten im Straßennetz innerhalb eines bestimmten Radius ausgewählt. Der Algorithmus berücksichtigt bei der Kandidatenauswahl sowohl die Distanzen zwischen GPS-Positionen und Map-Matching-Kandidaten als auch die Routing-Distanzen zwischen aufeinanderfolgenden Kandidaten.

GraphHopper unterstützt verschiedene kürzester-Pfad Routing-Algorithmen, wie Dijkstra und A* (unidirektional und bidirektional) sowie die Beschleunigungstechnik der Contraction Hierarchies. Um die Abkürzungskanten der Contraction Hierarchies zu sichern, ist eine spezielle Speicherung als *CHGraph* implementiert. Insgesamt werden drei unterschiedliche Algorithmen für die Durchführung des Routings angeboten (Karich 2017). Der Speed Modus nutzt Contraction Hierarchies, wodurch weniger RAM genutzt wird und daher sehr schnellen und leichtgewichtigen Response sendet. Als Einschränkung sind hier lediglich vordefinierte

Fahrzeugprofile möglich. Im Flexible Modus werden der Dijkstra- und A*-Algorithmus ohne Vorverarbeitung der Contraction Hierarchies angewendet. Für jeden Startknoten kann eine gewünschte Kursrichtung angegeben werden, sodass es wahrscheinlicher wird, dass eine Route in die ausgewiesene Richtung beginnt. Hierbei erhalten Straßen, die zu Beginn in eine andere Richtung führen, eine Strafe. Es ist zudem möglich alternative Routen zu berechnen, was allerdings Einfluss auf die Geschwindigkeit der Routing-Anfrage hat. Echtzeitänderung der Kantengewichte bietet volle Flexibilität, was aber mit einer eher langsamen Antwortzeit einhergeht. Der Hybrid Modus vereint die Vorteile des Flexible und des Speed Modus. Er ist sehr flexibel bei Attributänderungen und der Integration von Verkehrsdaten sowie bei Echtzeitänderungen auf die Kantengewichte. Insgesamt ist die Performance langsamer als beim Speed Modus, aber schneller als im Flexible Modus, zudem wird mehr Zeit und Speicherplatz für die Vorverarbeitung der Contraction Hierarchies erforderlich. Auf der anderen Seite wird für eine Anfrage weniger RAM verwendet. Im Hybrid Modus wird A* verwendet, mit einer Heuristik, die auf Orientierungspunkten basiert, und einer Dreiecksgleichung für die Zielrichtung. Ein Orientierungspunkt kann jeder Knoten auf dem Graphen sein. Auch bei diesem Modus kann eine Routing-Richtung angegeben werden, mit einer implementierten Strafe für die initiale Nutzung von Straßensegmenten, die in eine andere Richtung führen. Alternative Routenberechnung sind hier auch möglich, mit entsprechenden Auswirkungen auf die Geschwindigkeit der Routing-Anfrage.

C) Open Source Routing Machine (OSRM)

Die Open Source Routing Machine ist eine in C++ geschriebene Routing Engine, entwickelt, um OpenStreetMap-Daten zu verarbeiten. Während des Preprocessing, also der Phase der Vorverarbeitung, werden externe Memory-Algorithmen genutzt, um einen möglichst geringen Memory zu verbrauchen. Ein großer Teil der Datenvorverarbeitung läuft parallel, sodass Multi-Core-Prozessoren voll genutzt werden können. Informationen zur Route, Geometrie und Wegbeschreibungen werden für wichtige Abkürzungskanten vorberechnet, dadurch liefern Abfragen von längeren Distanzen schnellere Ergebnisse und somit wird die Routing-Prozedur erheblich effektiver, mit gleichzeitig nur einer beschränkten Nutzung des RAM (Luxen und Vetter 2011). Eine räumliche Baumstruktur wird genutzt, um umgekehrte Geocodierung zu erleichtern.

OSRM bietet verschiedene Services an, wie *Nearest*, bei dem Koordinaten auf das Straßennetz übertragen werden und die nächstgelegene Übereinstimmung zurückgibt. Weitere Services sind *Route*, um die schnellste Route zwischen zwei Koordinaten zu finden, *Table*, um die Dauer oder die Entfernung der schnellsten Route zwischen allen gegebenen Koordinatenpaaren zu berechnen, und *Match*, um aufgezeichnete Koordinaten wie GPS-Tracks auf plausible Weise auf ein logisches Modell der realen Welt zu übertragen (Project OSRM 2023a). Neben dem klassischen Routing, bietet OSRM auch die weitere Funktion *Trip*, um das Problems des Handelsreisenden (traveling salesman problem) zu lösen. Zudem werden in *Tile* Vektorkacheln angelegt, die die Metadaten des Routings beinhalten.

In OSRM stehen zwei Routing Algorithmen zur Verfügung, Contraction Hierarchies und ein mehrstufiger Dijkstra-Algorithmus, wodurch sehr schnelle Abfragezeiten entstehen und daher sehr geeignet für webbasierte Routing-Anwendungen ist (Project OSRM 2023b). Die Anwendung von Contraction Hierarchies wird für die Fälle empfohlen, in welchen die Abfrageleistung entscheidend ist, insbesondere für Relationen mit großer Entfernung. Bei

Abfragen, die schnell durchlaufen müssen, ist auch der mehrstufige Dijkstra-Algorithmus geeignet, wie zum Beispiel bei Live-Updates von Verkehrs-Daten. Von OSRM wird der Einsatz des Dijkstra-Algorithmus empfohlen, mit Ausnahme von speziellen Anwendungsfällen wie die Berechnung von sehr großen Distanzmatrizen, bei welchen die Verwendung der Contraction Hierarchies schneller ist. Dadurch, dass OpenStreetMap-Daten keinem festen Standard entsprechen, können Informationen auf unterschiedliche Weise beschrieben werden. Es ist somit notwendig die Routing-Daten in ein normalisiertes Format zu extrahieren. Von OSRM wird ein entsprechendes Tool zur Verfügung gestellt, das den Inhalt der exportierten OSM-Datei parst und Routing-Metadaten ausschreibt. Dabei wird anhand von unterschiedlichen Routingprofilen festgelegt, welche Straßenabschnitte befahrbar sind und ob Beschränkungen, wie Schranken oder die Deklaration als Privatstraße, die Zufahrt zu Straßen behindern. Somit beeinflussen Abbiegevorschriften und z.B. zeitliche Restriktionen die Berechnung des kürzesten Pfades. OSRM erleichtert den Import benutzerdefinierter Datensätze anstelle von OpenStreetMap-Daten durch die Nutzung eines einfachen Datenformats.

D) Gegenüberstellung der Routing-Engines

Zusammenfassend werden hier alle drei vorgestellten Routing-Engines gegenübergestellt.

Tab. 14: Eigenschaften der Routing-Engines nach (GIS-OPS UG 2020)

Eigenschaft		Valhalla	Graphhopper	OSRM
Fahrprofile	Auto	x	x	x
	Fußgänger	x	x	x
	Fahrrad	x	x	x
	Transit	x	x	-
API	Routing	x	x	x
	Isochronen	x	x	-
	Matrix	x	-	x
	Map Matching	x	x	x
Features	Abbiegevorschriften	x	x	x
	Vermeidungen von Polygonen	x	x	-
	dynamische Fahrzeugattribute	x	x	-
	alternative Routen	x	x	x
	Rundfahrten	-	x	-
	Höhenangaben	x	x	-

Routing-Engine: Valhalla

Vorteile:

Flexibilität

Das Besondere ist die dynamische Parametrisierung bei Abfragen, während gleichzeitig eine gute Leistung mit geringem Speicherbedarf aufrechterhalten wird. Jede Routen-Abfrage kann zusätzlich zu den vordefinierten Profilen flexibel verändert werden, durch Modifikation der Kosten, Strafen und anderer Faktoren.

RAM-Anforderungen

Der OSM-Planet, also alle Daten die durch die OpenStreetMap-Gemeinschaft gesammelt und in Planet.osm gespeichert werden, lässt sich mit 16 GB RAM für alle Profile aufbauen.

Besondere Merkmale

Es ist möglich Abfahrts- und Ankunftszeiten anzugeben. Zudem werden bei der Routenberechnung zeitliche Beschränkungen und Live- oder historische Verkehrsdaten berücksichtigt.

Nachteile:

Leistung

Statt eine Vorverarbeitung werden Beschleunigungstechniken verwendet, die in die Graphikstruktur durch Hierarchien und Verknüpfungen implementiert ist. Standard-Routing ist so zwar recht performant, Matrix-Anfragen dauern aber vergleichsweise lange.

Setup

Die Dokumentation zur Einrichtung von Valhalla ist recht gering.

Routing-Engine: GraphHopper

Vorteile:

Leistung

Die Engine gehört mit zu den schnellsten auf dem Markt. Selbst Routen über ganze Kontinente dauern selten länger als einige hundert Millisekunden.

Flexibilität

Die Laufzeitflexibilität mit benutzerdefinierten Profilen hat einen großen Einfluss auf die Performance auf längeren Strecken.

Transit

Das Routing ermöglicht die Berücksichtigung von öffentlichen Verkehrsmitteln. Dabei werden auch Echtzeitinformationen für den öffentlichen Nahverkehr verarbeitet.

Nachteile:

RAM-Anforderungen

Um die beste Leistung zu erzielen, wird in der Vorverarbeitung viel RAM benötigt. Für die Vorverarbeitung des OSM-Planeten werden 64 GB bis 128 GB RAM pro Profil verbraucht.

Anpassbarkeit

Zusätzliche Profile zu erstellen ist anspruchsvoll, da Java-Quellcode mit benutzerdefinierten Regeln zur Verwendung von OSM-Attributen geschrieben werden muss.

Gesperrte Features

Einige Features wurden für die kostenlose Nutzung gesperrt, wie die Matrix API und das LKW-Fahrprofil. Diese sind nur in der kommerziellen Directions API enthalten.

Routing-Engine: OSRM

Vorteile:

Leistung

Routingberechnungen dauern nur wenige Millisekunden, selbst bei sehr langen Strecken von über 1.000 Kilometern. Auch die Berechnung von Distanzmatrizen liefert sekundenschnell Millionen von Ergebnissen.

Infrastruktur

Es gibt viele Open-Source-Projekte, die mit OSRM verbunden sind. Viele Community-Tutorials und Add-Ons erleichtern den Einstieg in die Nutzung von OSRM.

Anpassbarkeit

Profile sind in der einfachen Skriptsprache Lua definiert. Es bietet viele Beispiele und fertige Funktionen, die man nutzen kann, um neue Profile zu definieren oder bestehende anzupassen.

Verkehr

Geschwindigkeitsaufzeichnungen können über einfache CSV-Dateien aktualisiert werden, damit diese auf regelmäßige Verkehrsaktualisierungen Anwendung finden.

Nachteile:

RAM-Anforderungen

Wird der vollständige OSM-Planet herangezogen, so benötigt die Vorverarbeitung mehr als 250 GB RAM pro Profil. Die Laufzeit verkürzt sich aber, zum Beispiel beim Absetzen von Anfragen, auf 35-40 GB.

Flexibilität

Gewichtungsinformationen sind fest im Graphen gespeichert und können nicht dynamisch zugewiesen werden. So ist es nicht möglich in einer Anfrage für Straßensegmente unterschiedliche Gewichtungen festzulegen.

Da im Weiteren ein Geschwindigkeitsprofil für Rettungswägen auf Basis der OpenStreetMap-Straßendaten entwickelt und implementiert wird, ist es notwendig, dass die Routing-Engine selbst definierte Profile verarbeiten kann. Um die Einzugsbereiche der Krankenhäuser zu erhalten, werden Matrixberechnungen unvermeidlich sein. Daher erscheint OSRM als geeignetste Engine, um das Routing mit den modifizierten Straßensegmenten durchzuführen. Auch die besonders schnelle Leistungsfähigkeit beim Berechnen von vielen Relationen bekräftigt die Entscheidung für OSRM.

3.2 Operationalisierung des Straßengraphen

Die Straßendaten aus OpenStreetMap beinhalten Informationen zum Routing, unter anderem auch den Routing-Modus ‚Auto‘.

Beim Routing wird die optimierte Reisezeit, bzw. Reiseweg ermittelt, also der Weg mit den geringsten Kosten. Straßenkreuzungen im Straßennetz werden durch Knoten repräsentiert, Straßenabschnitte, die diese verbinden, durch Kanten. Die Gewichtung der Kanten stellt die Zeit dar, die es braucht entlang des Straßensegments zu fahren. Da die Geschwindigkeiten auf verschiedenen Straßentypen variiert, sind die kürzeste und die schnellste Route typischerweise unterschiedlich. Der Weg von A nach B ist eine Sequenz von Kanten, wobei der kürzeste Weg die niedrigste Summe der Kantengewichtungen unter allen möglichen Pfaden aufweist. Dabei gilt, je länger das Straßensegment, umso stärker ist die Gewichtung, bzw. je schneller die Reisezeit umso geringer die Gewichtung. Die Geschwindigkeit ist dabei abhängig von der Straßenkategorie und der Art des Fahrzeugs, bzw. Fortbewegungsmittels.

Geschwindigkeitsprofile stellen das Routingverhalten für verschiedene Transportmittel wie Auto, Fahrrad und Fuß dar. Sie beschreiben, ob es möglich ist, entlang einer bestimmten Art von Weg zu routen, ob ein bestimmter Knoten passiert werden kann und die Maximalgeschwindigkeit bei Durchfahren eines Wegabschnitts. Dies fließt in die Art und Weise ein, wie der Routing-Graph erstellt wird und beeinflusst somit die Ausgaberouten. Diese beinhalten Abbiegebeschränkungen, die entscheidend sind für die korrekte Fahrzeugnavigation und helfen verbotene Abbiegungen zu vermeiden.

3.2.1 Implementierung des OSRM-Backends

Um die Routingfunktion der OSRM nutzen zu können muss das Backend implementiert werden. Da der OSRM Routing-Service für Windows nicht länger verfügbar ist, wird dieser in einem Docker-Container unter Linux ausgeführt (Docker, Inc. 2023). Ein Container ist ein Sandbox-Prozess, der von allen anderen Prozessen isoliert ist. Er läuft in einer unabhängigen Umgebung, die losgelöst von der umgebenden Infrastruktur und dem Betriebssystem arbeitet. Der Vorteil von Containern ist, dass sie auf lokalen Maschinen oder virtuellen Rechnern ausgeführt oder in der Cloud bereitgestellt werden können und kompatibel mit jedem Betriebssystem sind. Auch untereinander laufen Container isoliert voneinander und führen ihre eigene Software, Binärdaten und Konfigurationen aus. Ein Container ist die ausführbare Instanz eines isolierten Dateisystems. Dieses benutzerdefinierte Dateisystem wird von einem Container-Image bereitgestellt. Da das Image das Dateisystem des Containers enthält, muss es alles enthalten, was zu Ausführen einer Anwendung erforderlich ist – alle Abhängigkeiten, Konfigurationen, Skripte, Binärdaten, usw. Das Image enthält auch andere Konfigurationen für den Container, zum Beispiel Umgebungsvariablen, einen auszuführenden Standardbefehl und andere Metadaten.

Nach der Installation von Docker Desktop for Windows muss das OSRM-Docker-Image des OSRM-Backends geholt werden (*pull osrm/osrm-backend:latest*). Die Version v5.27.1, veröffentlicht am 14. Oktober 2022, wird als Backend genutzt⁷. Mit dem Pre-Processing wird der OSRM-Backend-Dienst initialisiert. Dazu wird mit dem OSRM-Tool Extractor (*osrm-extract*) zunächst das Kartenmaterial der heruntergeladenen .osm.pbf-Datei entpackt und aus der OSM-Basiskarte ein Graph extrahiert, der für das Routing eingesetzt werden kann. Die

⁷ <https://hub.docker.com/r/osrm/osrm-backend/> (Stand: 29.12.2022)

Basisdaten enthalten Informationen, die für das Routing irrelevant sind, wie zum Beispiel die Position von Bushaltestellen, Wegkreuze oder Parkbänke, diese werden in diesem Schritt ausgeschlossen. Zudem entsprechen die Daten keinem harten Standard, sodass wichtige Informationen auf verschiedene Weise beschrieben werden können. Daher ist es notwendig, die Routing-Daten in ein normalisiertes Format zu extrahieren.

Mit der Angabe des zu verwendenden Fahrprofils werden die Geschwindigkeiten der einzelnen Straßenkategorien festgelegt und definiert, was befahrbar ist unter Berücksichtigung von Privatstraßen, Schranken, usw. In OSRM sind Fahrprofile nicht reine Konfigurationsdateien, sie sind Skripte, geschrieben in der Skriptsprache Lua (Project OSRM 2023). Der Grund dafür ist, dass OSM-Daten komplex sind und es nicht möglich ist einfache Tag-Mappings zu definieren. Die Skriptsprache Lua bietet die Möglichkeit alle möglichen Tag-Kombinationen zu verarbeiten, die in den Knoten und Wegen zu finden sind. Über das Profil ist es möglich festzustellen, welche Wege in welche Richtung mit welcher Geschwindigkeit geroutet werden können. Die Basisstruktur eines .lua-Profiles gibt die API-Version an, ruft erforderliche Library-Dateien auf, definiert eine Setup-Funktion und verschiedene Prozessfunktionen sowie eine Rückgabe-Funktionstabelle. Beim Laden des Profils wird einmalig die Setup-Funktion aufgerufen. Sie gibt eine Hashtabelle an Konfigurationen zurück, in welcher die Geschwindigkeiten für verschiedene Straßenkategorien angegeben werden können. Viele Anpassungen können durch Modifizieren dieser Konfigurationstabelle vorgenommen werden. An dieser Stelle können auch zum Beispiel Höhendaten geladen werden, wenn diese für die Verarbeitung von Wegen berücksichtigt werden sollen. Daran anschließen die Funktionen *process_node* und *process_way*, die beim Extrahieren der OSM-Daten (*osrm-extract*) für jeden Knoten und Weg aufgerufen werden. Die Funktion *process_turn* verarbeitet jede Abbiegemöglichkeit im Netzwerk und setzt Strafen in Abhängigkeit von Abbiegewinkel und Ampeln. Die Standardeinstellung für den Aufschlag an Ampeln wird belassen, da Rettungswägen lediglich in Schrittgeschwindigkeit eine rote Ampel überfahren dürfen und es durch den Rückstau von wartenden Verkehrsteilnehmern eine zusätzliche Verzögerung gibt, sodass der Standardwert weiterhin angenommen wird.

Basisstruktur des car.lua-Profiles:

```
-- Car profile

api_version = 4

Set = require('lib/set')
Sequence = require('lib/sequence')
Handlers = require("lib/way_handlers")
Relations = require("lib/relations")
find_access_tag = require("lib/access").find_access_tag
limit = require("lib/maxspeed").limit
Utils = require("lib/utils")
Measure = require("lib/measure")

function setup()
    return {
        ...
    }
end
```

```

function process_node(profile, node, result, relations)
  ...
end

function process_way(profile, way, result, relations)
  ...
end

function process_turn(profile, turn)
  ...
end

return {
  setup = setup,
  process_way = process_way,
  process_node = process_node,
  process_turn = process_turn
}

```

Um das klassische Routing nach kürzester Zeit oder kürzester Weg zu verfeinern, bietet OSRM das Konzept der Gewichte und Raten an. Dabei stellt die Rate ein abstraktes Maß dar, das einzelnen Wegen zugewiesen werden kann, um bestimmte Wege anderen vorzuziehen. Der Routingalgorithmus bevorzugt entsprechende Wege mit hoher Rate. Wenn bestimmte Straßenkategorien priorisiert werden sollen, muss die Rate entsprechend erhöht werden. Ein weiterer Faktor ist die Gewichtung. Sie kann als Widerstand oder Kosten beim Durchfahren des Weges betrachtet werden, daher werden Wege mit niedrigen Kosten/geringer Gewichtung bei der Berechnung der Route bevorzugt. Wird die Gewichtung eines Weges auf einen festen Wert gesetzt, so wird die Länge und die Rate des Weges ignoriert.

Für die Fahrstreckenberechnung der Einsatzfahrten wird das Standardprofil für Autos *car.lua* dahingehend angepasst, dass Baustellen vom Routing nicht ausgeschlossen werden (siehe Anhang I: *car_con.lua*):

```

function process_way(profile, way, result, relations)
  local data = {
    highway = way:get_value_by_key('highway'),
    bridge = way:get_value_by_key('bridge'),
    route = way:get_value_by_key('route'),
    -- Anpassung Baustellen:
    construction = way:get_value_by_key('construction'),
  }
  if data.construction ~= nil and data.construction ~= 'minor' then
    data.highway = data.construction
    data.construction = nil
  end
end
...

```

Die Umfahrung von temporären Baustellen verfälscht das Routing-Ergebnis der Einsatzfahrten, da nicht auszuschließen ist, dass zum Zeitpunkt der Einsatzfahrt entsprechende Baustellen in OSM bereits erfasst waren. Da im Endergebnis ein allgemeingültiges Geschwindigkeitsprofil für Rettungswägen entstehen soll, dürfen Baustellen nicht in die Strecken-Berechnung einbezogen werden. Als Gewichtung wird *routability* belassen, das Routen mit der kürzesten Zeit berechnet und die Priorisierung von Wegen über

Raten und Gewichtungen zulässt. In einem Notfall muss ein Einsatzfahrzeug so schnell wie möglich am Einsatzort eintreffen, um dem Verunglückten eine erste qualifizierte Hilfe zu leisten. Daher kann von der Routenpräferenz der zeitlich schnellsten Fahrstrecke ausgegangen werden.

Da für das Routing der Algorithmus Multi-Level-Dijkstra (*--algorithm mld*) ausgewählt wird, muss eine rekursive Partitionierung des OSM-Graphen in Zellen (*osrm-partition*) vorgenommen werden, um im Anschluss Anpassungen zu erfahren (*osrm-customize*), indem Routing-Gewichtungen für alle Zellen berechnet werden. Die mögliche Alternative der Contraction Hierarchies wird eher für die Berechnung von größeren Distanzen empfohlen, da aber lediglich innerhalb von Bayern in einem relativ geringen Radius Routen gefunden werden müssen, bietet sich der MLD-Algorithmus an. Sobald der Graph angepasst ist, wird der http-Server mit Port 5000 (*osrm-routed*) gestartet, anschließend kann der OSRM-Backend-Dienst als Docker-Image eigenständig gestartet und ausgeführt werden.

URL um die Routing-Engine anzusprechen und eine Route von A nach B zu berechnen:

```
http://127.0.0.1:5000/route/v1/driving/<X_Startkoordinate>,<Y_Startkoordinate>;<X_Zielkoordinate>,<Y_Zielkoordinate>
```

3.2.2 Zuweisung der relevanten Straßenkategorien

Die aus OpenStreetMap verfügbaren Straßenkategorien sind für die Berechnung von Geschwindigkeitsprofilen zu allgemein. Rettungsfahrzeuge im Notfalleinsatz mit Blaulicht und Martinshorn sind zwar von der Straßenverkehrsordnung ausgenommen und genießen Sonderrechte (§35 StVO), allerdings bedingt die kastenförmige Bauart Einbußen bei höheren Geschwindigkeiten. Ein Rettungswagen kann auf Autobahnen in Bereichen mit Richtgeschwindigkeit 130 km/h, aufgrund des schweren Aufbaus, nicht die volle Geschwindigkeit ausfahren und viele Fahrzeuge werden bei höheren Geschwindigkeiten sogar abgeregelt. Hingegen bei Geschwindigkeitsbegrenzungen von 80 km/h auf Autobahnen ist es ihm erlaubt schneller zu fahren. Daher werden die einzelnen für die Befahrung gültigen Straßenkategorien in spezifischere Wegeklassen aufgeteilt, die die Höchstgeschwindigkeit miteinbeziehen. Die Höchstgeschwindigkeit wird aus dem Attribut *maxspeed* gelesen, bzw. aus *zone:traffic*, *zone:maxspeed* und *source:maxspeed*.

Um Baustellen als regelhafte Straße zu deklarieren, muss auf die eigentliche Straßenkategorie, die bei *key:highway=construction* im Schlüssel *construction* zu finden ist, verwiesen werden. Baustellen mit dem Wert *minor* werden zwar auch als Baustellen klassifiziert, dadurch, dass diese aber als befahrbar deklariert sind, muss auf diese nicht weiter eingegangen werden.

Tab. 15: Kategorienbildung aus den OSM-Wege-Elementen

Kategorie OSM	Kategorie neu	Anzahl Kanten	Geschwindigkeitsklasse
motorway	autobahn_x	47.292	$v > 120$ km/h oder v unbekannt
	autobahn_120	6.162	$100 < v \leq 120$ km/h
	autobahn_100	2.730	$80 < v \leq 100$ km/h
	autobahn_80	2.521	$0 < v \leq 80$ km/h
motorway_link	autobahn_link	43.846	

trunk	schnell_x	5.673	$v > 100$ km/h oder v unbekannt
	schnell_100	11.715	$60 < v \leq 100$ km/h
	schnell_60	2.839	$0 < v \leq 60$ km/h
primary	bundes_x	69.756	$v > 80$ km/h oder v unbekannt
	bundes_80	29.678	$50 < v \leq 80$ km/h
	bundes_50	37.142	$0 < v \leq 50$ km/h
pedestrian	fussgaenger	48.381	
mini_roundabout	kreisverkehr	57.395	
secondary tertiary	land_x	796.351	$v > 50$ km/h oder v unbekannt
	land_50	332.138	$30 < v \leq 50$ km/h
	land_30	35.585	$0 < v \leq 30$ km/h
trunk_link primary_link secondary_link tertiary_link	link	52.130	
unclassified	neben_x	998.002	$v > 50$ km/h oder v unbekannt
	neben_50	160.321	$0 < v \leq 50$ km/h
residential	anlieger_50	2.258.394	
living_street	spiel	89.283	
service	zufahrt	2.299.360	
track	feldweg	7.136.399	
yes turning_loop turning_circle road	sonstige	1.285	

Über ein Python-Skript (siehe Anhang II: `osm2sqlite.py`) wird in einer SQLite-Datenbank eine Tabelle `EDGES` angelegt in welche die einzelnen Kanten des `bayern-latest.osm`-Files eingefügt werden. Die Kanten eines Weges basieren auf der Liste der IDs der am Weg beteiligten Knoten (`<nd ref="ID_KNOTEN"/>`). Jede Kante ist über zwei Knoten (`NODE_VON` und `NODE_NACH`) definiert, diese werden der Tabelle `EDGES` hinzugefügt. Bei der Geschwindigkeitsprofilerstellung wird für beide Fahrtrichtungen das gleiche Geschwindigkeitslimit angenommen, daher wird der Knoten mit der niedrigeren ID als `NODE_VON` und derjenige mit der höheren ID als `NODE_NACH` definiert. Jeder Kante, also `NODE_VON-NODE_NACH`-Kombination, wird anschließend die neue Straßenkategorie zugewiesen. Das Feld `ISAREA` gibt an, ob der Straßenabschnitt über ein Liniensegment oder

über ein Polygon definiert ist. Bei einer Fläche, wie zum Beispiel bei Fußgängerzonen, wird entlang der Außenkanten geroutet. Die Tabelle *EDGES* wird später genutzt, um die Anteile der verschiedenen Straßenkategorien zu ermitteln.

Tab. 16: Datentypen der Tabelle *EDGES*

Spaltenname	Datentyp	PrimaryKey	Beschreibung
NODE_VON	integer	X	ID des Ausgangsknoten
NODE_NACH	integer	X	ID des Zielknoten
ID	integer	X	ID des Wege-Elements
ISAREA	integer		gibt an, ob die Basis für den Straßenabschnitt als Polygon oder als Linie definiert ist
OSM_HIGHWAY	text		Straßenkategorie aus OSM
OSM_MAXSPEED	text		Angabe der Höchstgeschwindigkeit
KATEGORIE	text		selbst definierte Straßenkategorie

Beispieldatensatz:

NODE_VON = 36464808

NODE_NACH = 4470972803

ID = 5210121

ISAREA = 0

| 1 = Basis Fläche; 0 = Basis Linie

OSM_HIGHWAY = residential

OSM_MAXSPEED = 50

| aus maxspeed, zone:traffic,
zone:maxspeed, source:maxspeed

KATEGORIE = anlieger_50

3.2.3 Ermittlung der Einsatz-Fahrstrecken

Im nächsten Schritt werden die Anteile der verschiedenen Straßentypen an den Einsatzfahrten bei Routing mit dem angepassten Standard-Profil, mit befahrbaren Baustellen, ermittelt (vgl. Anhang III: *calc_routes.py*). Die Anteile werden benötigt, um später die Fahrzeiten für verschiedene Straßenkategorien zu berechnen. Dadurch, dass in den Einsatzdaten lediglich die Start- und die Zielkoordinaten dokumentiert sind, ist die gefahrene Strecke unbekannt. Einzig die Fahrdauer (*fz_einsatz*), aus den dokumentierten Zeitstempeln, kann als Referenz herangezogen werden.

Aus dem Routing-Ergebnis werden die Knoten der Fahrstrecke mit den Knoten-Kombinationen *NODE_VON* und *NODE_NACH* aus der Tabelle *EDGES* zusammengeführt. Es kann für eine *NODE_VON-NODE_NACH*-Kombination mehrere Einträge in der Tabelle *EDGES* geben, da die Kante, die durch die beiden Knoten definiert wird, in unterschiedlichen Wege-Elementen enthalten sein kann.

- Kante in unterschiedlichen *id*: dieser Fall tritt auf, wenn Kanten bei sich überschneidenden Wege-Elementen doppelt erfasst wurden, oder es handelt sich um Kanten von zwei benachbarten Flächen, die jeweils das Attribut *highway* aufweisen:

	node_von	node_nach	id	isarea	osm_highway	osm_maxspeed	kategorie
1	25033439	298501531	1127345551	0	tertiary	-1	land_x
2	25033439	298501531	1127462478	0	tertiary	-1	land_x

	node_von	node_nach	id	isarea	osm_highway	osm_maxspeed	kategorie
1	9906272123	9906272129	198930053	1	service	-1	zufahrt
2	9906272123	9906272129	1080796265	1	service	-1	zufahrt

- Kante ist Teil einer Straße (*isarea* = 0) und gleichzeitig ein Teil eines flächigen Weges (*isarea* = 1). Dies tritt vornehmlich bei Fußgängerzonen auf, die als Fläche erfasst sind und zusätzlich ein Linienweg vorhanden ist, bzw. die Kante der Fußgängerzone mit einem Straßensegment überlagert ist.

	node_von	node_nach	id	isarea	osm_highway	osm_maxspeed	kategorie
1	11494110	4568646157	2589743	0	residential	30	anlieger_50
2	11494110	4568646157	106272210	1	pedestrian	-1	fussgaenger

Bei der finalen Auswahl muss eine eindeutige Zuordnung der Kante zur Kategorie erfolgen. Daher werden Liniensegmente den Kanten eines Polygons vorgezogen. Sollte *ISAREA* bei den Auswahlmöglichkeiten identisch sein, so wird nach der Angabe der Maximalgeschwindigkeit entschieden. Der Datensatz mit der höchsten *OSM_MAXSPEED* wird für die Zuordnung selektiert. Damit sind nun die entsprechenden Kategorien der einzelnen befahrenen Straßensegmente bekannt sowie die jeweilige potenziell gefahrene Distanz pro Straßenkategorie.

Konnte für eine *NODE_VON-NODE_NACH*-Kombination keine Kante gefunden werden, so ist auch keine Zuordnung zu einer Kategorie möglich. Daher werden entsprechende Straßensegmente der Hilfskategorie *XXX* zugewiesen.

Die benötigte Fahrzeit (*FZ_OSM*), die aus dem Routing hervorgegangen ist, wird als Referenz zur gemessenen Fahrzeit der Einsatz-Zeitstempel (*FZ_EINSATZ*) extrahiert. Für die gesamte geroutete Strecke wird die Durchschnittsgeschwindigkeit in km/h (*V_EINSATZ*) berechnet. Diese ergibt sich aus der gerouteten Gesamtstrecke dividiert durch die Fahrzeit aus den Zeitstempeln.

Als Ergebnis wird eine Tabelle ausgegeben, die die gefahrene Distanz je Einsatzfahrt differenziert nach Straßenkategorie angibt. Zusätzlich finden sich je Fahrstrecke die Durchschnittsgeschwindigkeit, die dokumentierte und geroutete Fahrzeit. Nachfolgende Tabelle gibt Lage- und Streuparameter der einzelnen Straßenkategorien an. So führt zum Beispiel bei insgesamt 15.897 Einsatzfahrten die Fahrstrecke über Straßensegmente der Kategorie *AUTOBAHN_120*. Die kürzeste Strecke wurde mit 0,59 Meter bei einer Einsatzfahrt zurückgelegt, die längste mit 27,32 km.

Tab. 17: statistische Kennwerte zur Beschreibung der gerouteten Einsatzfahrten nach Straßenkategorie

	AUTOBAHN_X	AUTOBAHN_120	AUTOBAHN_100	AUTOBAHN_80	AUTOBAHN_LINK
Anzahl Fahrten	39.672	15.897	10.478	11.293	51.543
Mittelwert [m]	6.241,79	2.550,67	1.499,60	1.400,80	860,90
Std.-Abw. [m]	4.768,19	2.489,50	1.456,60	1.498,87	538,78
Minimum [m]	6,49	0,59	24,34	9,08	2,35
Maximum [m]	76.673,01	27.322,06	17.381,92	11.326,44	8.058,48
	SCHNELL_X	SCHNELL_100	SCHNELL_60		
Anzahl Fahrten	20.755	51.212	34.784		
Mittelwert [m]	2.529,57	2.413,37	953,78		
Std.-Abw. [m]	3.002,50	2.751,36	1.005,82		
Minimum [m]	7,02	6,28	6,19		
Maximum [m]	35.961,59	49.969,18	10.418,37		
	BUNDES_X	BUNDES_80	BUNDES_50		
Anzahl Fahrten	154.991	162.841	215.556		
Mittelwert [m]	3.290,93	1.260,95	1.076,73		
Std.-Abw. [m]	3.309,09	1.313,51	1.088,27		
Minimum [m]	0,94	0,45	0,36		
Maximum [m]	46.232,99	15.711,04	13.184,57		
	LAND_X	LAND_50	LAND_30		
Anzahl Fahrten	451.154	609.502	145.411		
Mittelwert [m]	3.679,80	1.504,26	357,10		
Std.-Abw. [m]	3.821,69	1.218,99	326,64		
Minimum [m]	0,07	0,07	0,27		
Maximum [m]	46.481,29	14.549,05	3.402,71		
	NEBEN_X	NEBEN_50	ANLIEGER_50	SPIEL	
Anzahl Fahrten	148.633	124.503	655.843	33.627	
Mittelwert [m]	774,91	447,33	596,51	116,88	
Std.-Abw. [m]	1.062,53	474,15	502,85	105,80	
Minimum [m]	0,07	0,15	0,11	0,13	
Maximum [m]	11.368,01	11.850,27	7.243,75	1.296,91	

	FUSSGAENGER	KREISVERKEHR	LINK	
Anzahl Fahrten	1.304	294.840	199.015	
Mittelwert [m]	90,55	72,50	283,40	
Std.-Abw. [m]	83,10	56,68	270,55	
Minimum [m]	2,43	0,76	3,14	
Maximum [m]	532,82	951,96	2.176,11	

	ZUFAHRT	FELDWEG	SONSTIGE	XXX
Anzahl Fahrten	2	1	1	3.349
Mittelwert [m]	52,14	34,27	57,35	99,11
Std.-Abw. [m]	11,24	0,00	0,00	147,25
Minimum [m]	40,91	34,27	57,35	2,23
Maximum [m]	63,38	34,27	57,35	1.887,63

Zufahrtsstraßen (*SERVICE*), Feldwege (*TRACK*) und sonstige Straßenkategorien treten im Routing nur bei wenigen Einsatzfahrten auf. Dies könnte darauf zurückzuführen sein, dass zum einen die Start- und Zielkoordinaten eine entsprechende Wegführung außerhalb dieser Wege am schnellsten ist. Zum anderen sind diese Straßenkategorien nicht als öffentliche Straße zu klassifizieren, daher ist ein Abbiegen in ebendiese Straßen durch entsprechende Reglementierung nicht möglich.

Tab. 18: statistische Kennwerte der Fahrzeit-Variablen und der Geschwindigkeit der Einsatzfahrten vor der Datenbereinigung

	V_EINSATZ [km/h]	FZ_EINSATZ [min]	FZ_OSM [min]
Anzahl Fahrten	716.180	716.321	716.227
Mittelwert	54,92	7,91	8,53
Std.-Abw.	456,15	4,33	5,48
Minimum	0,00	0,02	0,00
25%-Perzentil	28,40	4,73	4,24
50%-Perzentil	43,90	7,18	7,58
75%-Perzentil	60,40	10,27	11,78
Maximum	195.742,80	96,72	92,10

Lediglich für elf Einsatzfahrten konnte keine Route gefunden werden. Neun Zielorte dieser nicht gerouteten Fahrten befinden sich in der Fußgängerzone in Erlangen, am Ende der Universitätsstraße. Die Abbiegevorschriften zeigen ein Links-Abbiege-Verbot (*restriction = only_right_turn*), sodass ein Erreichen der Einsatzorte nicht möglich ist (vgl. Abb. 15).

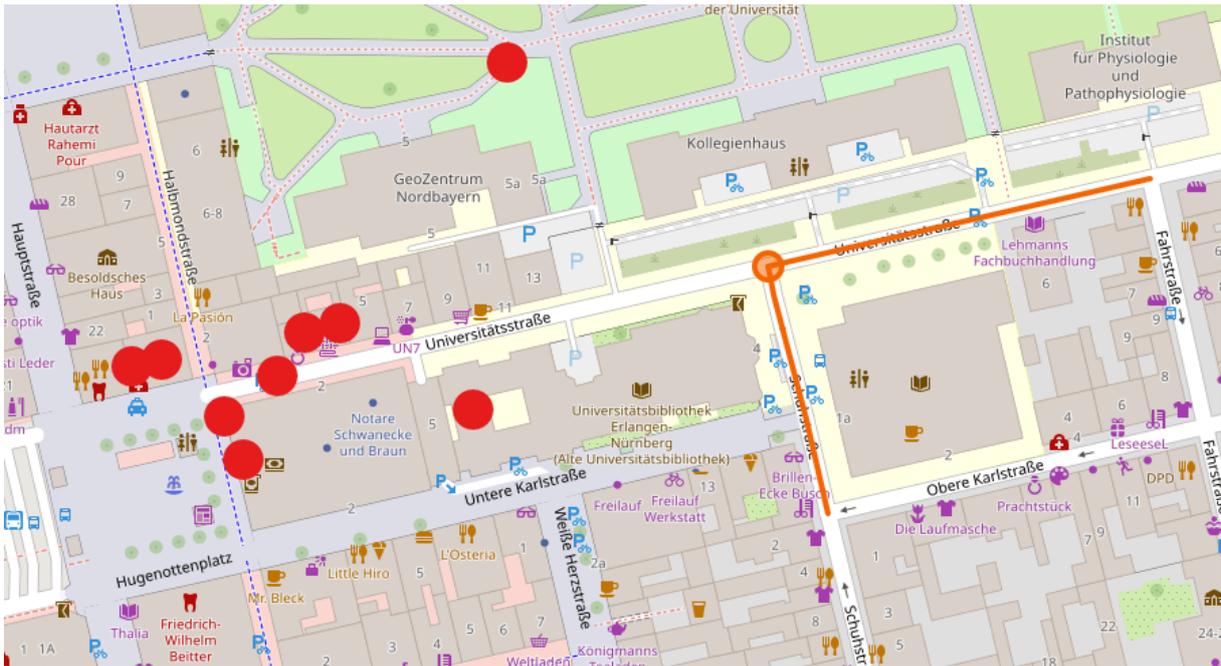


Abb. 15: nicht erreichbare Einsatzorte in Fußgängerzone

Für zwei weitere Einsatzfahrten wurde keine Route gefunden. Dies könnte daran liegen, dass die Zielkoordinaten auf Punkte innerhalb einer Obstplantage verweisen und der nächstgelegene Weg als ein nicht befahrbarer Feldweg ausgewiesen ist.



Abb. 16: nicht erreichbare Einsatzorte in Obstplantage

3.2.4 Berechnung eines Geschwindigkeitsprofils für RTW

Um eine saubere Datenbasis zur weiteren Geschwindigkeitsmodellierung zu erhalten, muss der geroutete Output bereinigt, also Ausreißer und unplausible Datensätze identifiziert und entfernt werden. Da von 716.332 Einsätzen lediglich elf nicht geroutet werden konnten, umfasst die Datenbasis für die Weiterverarbeitung 716.321 Datensätze.

FZ_EINSATZ mindestens 2/3 der FZ_OSM bzw. FZ_OSM mind. 2/3 FZ_EINSATZ (n = 547.599):

Um Ausreißer und Fehler in der Dokumentation der Zeitstempel auszuschließen werden Einsatzfahrten ausgewählt, bei welchen die dokumentierte Fahrzeit und die geroutete Fahrzeit nicht zu weit voneinander abweichen.

FZ_EINSATZ maximal 15 Minuten (n = 672.686):

Laut der Verordnung zur Ausführung des Bayerischen Rettungsdienstgesetzes (AVBayRDG Abschnitt 1 §2) müssen „Notfälle [...] in der Regel spätestens 12 Minuten nach dem Ausrücken eines [...] Rettungsmittel erreicht werden“⁸. Dies bezieht sich auf das ersteintreffende qualifizierte Rettungsmittel, das bei einem Notfall mit Verdacht auf Tracer-Diagnose auch ein Notarzteeinsatzfahrzeug sein kann, das zusätzlich zum Rettungswagen alarmiert wird. So kann es sein, dass der Notarzt binnen maximal 12 Minuten vor Ort ist und qualifizierte erste Hilfe leisten kann, der Rettungswagen aber einige Minuten später erst am Einsatzort ankommt. Da es sich um eine zeitkritische Diagnose handelt, werden lediglich Einsätze mit einer dokumentierten Fahrzeit von maximal 15 Minuten für die Geschwindigkeitsmodellierung herangezogen.

V_EINSATZ maximal 130 km/h (n = 706.236):

Da ein Rettungswagen bei höheren Geschwindigkeiten abgeriegelt werden, ist es nicht wahrscheinlich, dass diese schneller als 130 km/h fahren. Das Feld V_EINSATZ wird über die Gesamtstrecke aus dem Routing dividiert durch die dokumentierte Fahrzeit berechnet. Geschwindigkeiten von über 130 km/h deuten auf eine fehlerhafte Dokumentation der Zeitstempel für Ausrücken und für Ankunft am Einsatzort hin.

Bei insgesamt 511.932 Datensätzen sind alle Bedingungen erfüllt.

Tab. 19: statistische Kennwerte der Fahrzeit-Variablen der Einsatzfahrten nach der Datenbereinigung

	V_EINSATZ [km/h]	FZ_EINSATZ [min]	FZ_OSM [min]
Anzahl	511.932	511.932	511.932
Mittelwert	46,52111	7,63116	8,33745
Standardabweichung	16,89179	3,21191	4,10304
Minimum	6,5	0,02	0,015
25%-Perzentil	32,9	5,05	5,00333
50%-Perzentil	44,9	7,35	7,845
75%-Perzentil	58,6	10,0	11,15833
Maximum	129,8	15,0	22,42167

⁸ Verordnung zur Ausführung des Bayerischen Rettungsdienstgesetzes (AVBayRDG): §2 Abs. 1 Satz 3 (Erster Teil Organisation und Durchführung – Abschnitt 1 Allgemeines)

Zunächst wird untersucht, ob ein linearer Zusammenhang zwischen der dokumentierten Fahrzeit ($FZ_EINSATZ$) und der über das OSM-Straßennetz gerouteten Fahrzeit (FZ_OSM) besteht. Dazu bietet sich eine lineare Regression an. Das Regressionsmodell untersucht einen angenommenen funktionalen Zusammenhang zwischen einer abhängigen Variable Y und einer unabhängigen Variable X. Die Regressionsgerade stellt die minimale Summe der quadrierten Abstände der einzelnen Datenpunkte zur Geraden dar, sie bildet den linearen Zusammenhang zwischen den Fahrzeiten ab. Dabei beschreibt das Residuum die Abstandsquadrate zur Regressionsgeraden. Davon lässt sich die Korrelation zwischen den dokumentierten und den gerouteten Fahrzeiten abschätzen.

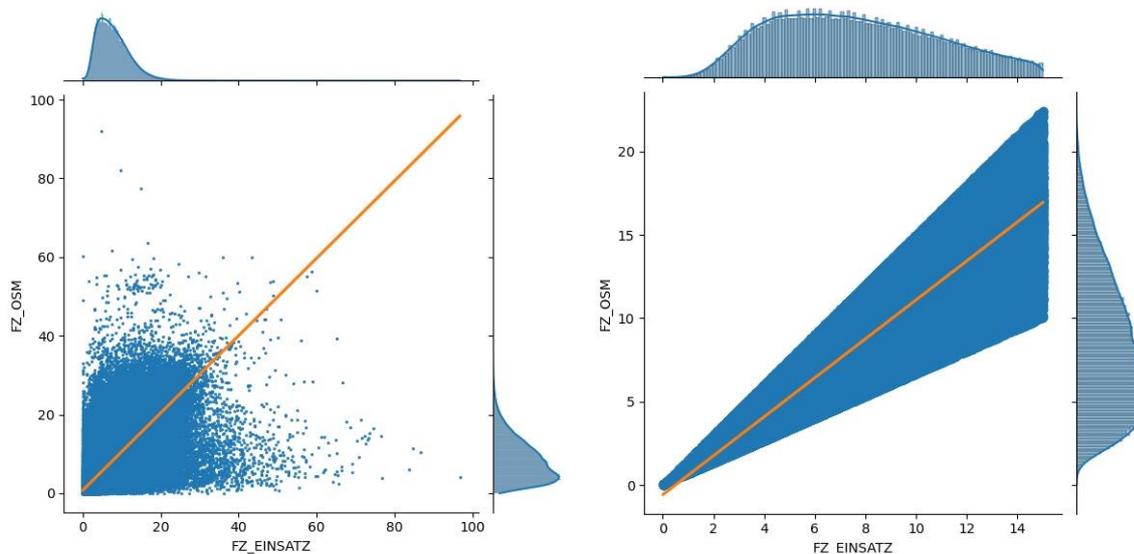


Abb. 17: linearer Zusammenhang der dokumentierten und der gerouteten Fahrzeiten vor (links) und nach (rechts) der Datenbereinigung

Mit den Datensätzen, wie sie aus dem Routing entstehen, lässt sich ein nur schwacher linearer Zusammenhang mit einem R^2 von 60,3% erkennen, mit einer stärkeren Streuung bei längeren Fahrzeiten. Nach der Datenbereinigung wird mit einem R^2 von 83,78% die Güte der Anpassung deutlicher, sodass das lineare Modell gut an die Daten angepasst ist und die Streuung weitgehend erklärt. Aber auch hier divergiert die Streuung mit zunehmender Länge der Fahrzeit. Dadurch, dass ein linearer Zusammenhang zwischen der dokumentierten und der gerouteten Fahrzeit feststellbar ist, eignen sich die Daten zur weiteren Schätzung der Geschwindigkeiten je Straßenkategorie.

Tab. 20: Gegenüberstellung der Ergebnisse der linearen Regression vor und nach der Datenbereinigung

	vor Datenbereinigung	nach Datenbereinigung
y-Achsen-Abschnitt	0,7526551	-0,5852097
Steigung	0,9825311	1,16924019
R^2	0,6030816	0,837767

Negative Werte in der Geschwindigkeitsmodellierung sind als ungültig oder unzulässig anzusehen. Die Methode der nicht negativen kleinsten Quadrate (NNLS) löst das Problem der negativen Werte der linearen Regression bei einer gegebenen m -mal- n -Modellmatrix A und

einem Zielvektor b mit m Elementen. Bei der klassischen linearen Regression können für die Schätzung der Regressionskoeffizienten negative Werte entstehen. Im Gegensatz dazu werden die Schätzungen bei NNLS auf Null oder positive Werte beschränkt. Das lineare Modell wird mit Koeffizienten dahingehend angepasst, sodass die Residualsumme der Quadrate zwischen den beobachteten Zielen im Datensatz und den durch die lineare Annäherung vorhergesagten Ziele minimiert werden. Somit ist unter der Annahme $x \geq 0$ das zu findende Ziel:

$$\arg \min_{x \geq 0} \| Ax - b \|_2^2$$

Modellmatrix A als Array (unabhängige Variablen):

m = Anzahl der gerouteten Einzeleinsatzfahrten

n = Straßenkategorien

Zielvektor b (abhängige Variable):

b = dokumentierte Fahrzeit der Einzeleinsätze

Um die kleinste Quadrate Schätzung für die Modellkoeffizienten zu erhalten, werden die dokumentierte Fahrzeiten auf die einzelnen Straßenkategorien regressiert (vgl. Anhang IV: `leastquares.py`). Dabei wird versucht die Summe der quadrierten Abweichungen zu minimieren. In Python bietet die Funktion `sklearn.linear_model.LinearRegression(*, positive = True)` die minimierte Lösung für das Gleichungssystem $Ax=b$ unter der Bedingung, dass alle Elemente von x nicht negativ sind. Aus dieser Funktion ergeben sich folgende geschätzte Werte für die Koeffizienten x und damit das geschätzte Fahrprofil je Straßenkategorie, umgerechnet in km/h:

Tab. 21: geschätzte Geschwindigkeiten je Straßenkategorie

Kategorie	Geschwindigkeit geschätzt [km/h]	Geschwindigkeitsklasse nach Höchstgeschwindigkeit	
		Minimum [km/h]	Maximum [km/h]
AUTOBAHN_X	133,81	> 120	∞
AUTOBAHN_120	129,16	> 100	≤ 120
AUTOBAHN_100	105,47	> 80	≤ 100
AUTOBAHN_80	88,66	> 0	≤ 80
AUTOBAHN_LINK	59,98	> 0	∞
SCHNELL_X	135,72	> 100	∞
SCHNELL_100	110,49	> 60	≤ 100
SCHNELL_60	59,56	> 0	≤ 60
BUNDES_X	106,91	> 80	≤ 100
BUNDES_80	79,76	> 50	≤ 80
BUNDES_50	55,96	> 0	≤ 50
LAND_X	87,72	> 50	≤ 100
LAND_50	56,56	> 30	≤ 50

Kategorie	Geschwindigkeit geschätzt [km/h]	Geschwindigkeitsklasse nach Höchstgeschwindigkeit	
		Minimum [km/h]	Maximum [km/h]
LAND_30	36,69	> 0	≤ 30
NEBEN_X	65,54	> 50	≤ 100
NEBEN_50	54,49	> 0	≤ 50
ANLIEGER_50	44,76	> 0	≤ 50
SPIEL	20,42	> 0	≤ 7
FUSSGAENGER	14,68	> 0	≤ 7
KREISVERKEHR	60,02	> 0	≤ 30
LINK	50,29	> 0	≤ 100
ZUFAHRT	0	-	-
FELDWEG	0	> 0	≤ 7
SONSTIGE	0	-	-
XXX	22,94	-	-

Die Straßenkategorien ZUFAHRT, FELDWEG und SONSTIGE weisen keine Geschwindigkeit auf, da aufgrund der geringen Anzahl der befahrenen Straßenabschnitte keine Schätzung möglich ist. Für diese Kategorien werden manuell die Geschwindigkeiten bestimmt. Feldwege werden aufgrund der zumeist schlechte Oberfläche auf Schrittgeschwindigkeit von 7 km/h gesetzt, Zufahrtsstraßen und sonstige Straßen erhalten die Standardgeschwindigkeit aus OpenStreetMap von 30 km/h.

3.2.5 Validierung des Geschwindigkeitsprofils für RTW

Um das Ergebnis zu validieren, werden unter Verwendung der `sklearn.linear_model.LinearRegression(*, positive = True)`-Funktion zehn Stichproben mit jeweils 200.000 Einsatzfahrten entnommen.

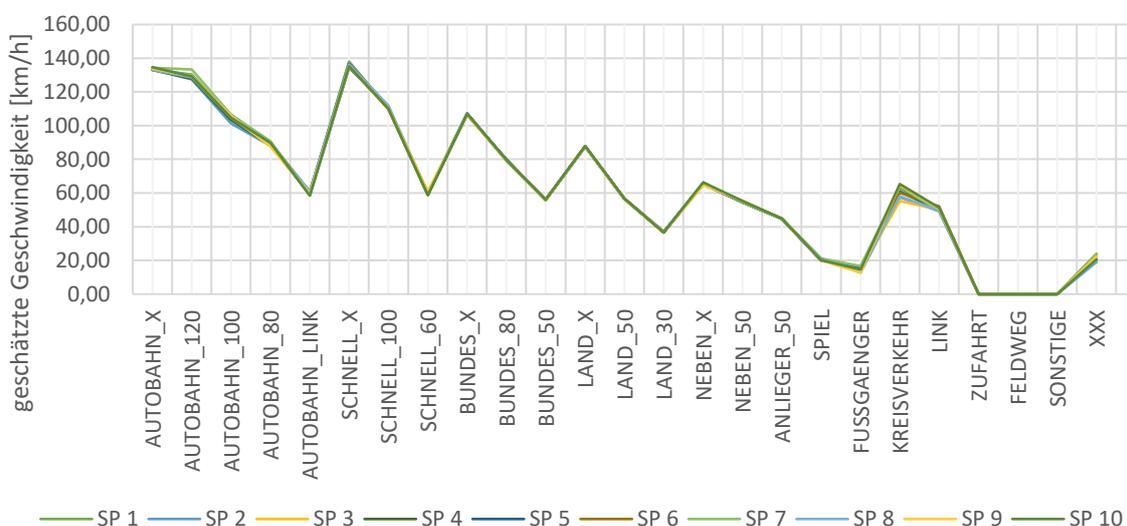


Abb. 18: NNLS-Geschwindigkeitsmodellierung von zehn Stichproben

Tab. 22: NNLS-Geschwindigkeitsmodellierung von zehn Stichproben - Einzelergebnisse in km/h

	AUTOBAHN_X	AUTOBAHN_120	AUTOBAHN_100	AUTOBAHN_80	AUTOBAHN_LINK
SP 1	133,28	130,24	105,17	89,55	60,69
SP 2	133,2	127,41	101,28	87,99	60,82
SP 3	134,37	128,76	105,32	87,89	59,96
SP 4	133,15	127,79	103,32	88,25	60,56
SP 5	134,23	128,58	105,55	88,86	58,96
SP 6	132,99	128,93	106,44	90,04	60,27
SP 7	134,31	133,24	106,46	90,69	58,66
SP 8	133,22	128,67	104,71	88,49	61,06
SP 9	133,92	129,4	105,86	87,43	59,61
SP 10	134,57	129,25	104,63	89,77	58,45
Std.-Abw.	0,580072409	1,535005212	1,486662033	1,01411242	0,894418247

	SCHNELL_X	SCHNELL_100	SCHNELL_60
SP 1	137,93	110,46	60,36
SP 2	137,53	110,52	59,59
SP 3	136,88	111,16	59,37
SP 4	136,76	111,14	58,79
SP 5	134,49	110,72	59,38
SP 6	135,27	109,85	58,84
SP 7	135,32	110,61	59,56
SP 8	136,24	111,98	60
SP 9	135,07	110,36	60,75
SP 10	135,19	110,13	58,78
Std.-Abw.	1,105692543	0,574857374	0,635260576

	BUNDES_X	BUNDES_80	BUNDES_50
SP 1	106,87	79,49	55,99
SP 2	107,38	79,6	55,99
SP 3	106,73	79,6	55,95
SP 4	107,12	79,53	55,98
SP 5	106,51	80	55,82
SP 6	106,75	79,63	56,17

SP 7	107,04	79,03	55,6
SP 8	106,6	79,47	56,25
SP 9	106,52	79,52	55,95
SP 10	106,98	79,76	55,94
Std.-Abw.	0,268067156	0,231950426	0,167343957

	LAND_X	LAND_50	LAND_30
SP 1	87,71	56,44	36,71
SP 2	87,8	56,58	36,66
SP 3	87,91	56,58	36,44
SP 4	87,51	56,65	36,54
SP 5	87,54	56,59	36,7
SP 6	87,67	56,7	37,03
SP 7	87,7	56,45	36,77
SP 8	87,66	56,42	37,02
SP 9	87,67	56,42	36,62
SP 10	87,81	56,57	36,7
Std.-Abw.	0,11461239	0,095498691	0,17716941

	NEBEN_X	NEBEN_50	ANLIEGER_50	SPIEL
SP 1	66,21	54,51	45,01	20,76
SP 2	64,89	54,46	44,79	21,22
SP 3	65,84	54,24	44,88	20,81
SP 4	65,5	54,51	44,71	20,16
SP 5	65,49	54,59	44,67	19,89
SP 6	65,4	54,81	44,94	20,65
SP 7	65,98	54,5	44,69	21,09
SP 8	65,47	55,23	44,65	20,63
SP 9	64,74	55,45	44,78	20
SP 10	66,17	55,19	44,78	20,23
Std.-Abw.	0,470328609	0,382764941	0,113841996	0,431374547

	FUSSGAENGER	KREISVERKEHR	LINK
SP 1	13,9	61,19	49,33
SP 2	12,88	57,59	49,28
SP 3	14,95	55,29	50,23
SP 4	14,8	63,2	50,42
SP 5	15,93	60,71	50,28
SP 6	13,99	60,13	51,98
SP 7	16,68	63,83	49,65
SP 8	13,96	58,07	49,43
SP 9	12,77	64,98	50,69
SP 10	14,62	65,29	51,21
Std.-Abw.	1,166454457	3,174582807	0,83414627

	ZUFAHRT	FELDWEG	SONSTIGE	XXX
SP 1	0	0	0	21,59
SP 2	0	0	0	19,16
SP 3	0	0	0	20,91
SP 4	0	0	0	20,7
SP 5	0	0	0	22,97
SP 6	0	0	0	23,76
SP 7	0	0	0	23,56
SP 8	0	0	0	19,9
SP 9	0	0	0	22,13
SP 10	0	0	0	20,52
Std.-Abw.	0	0	0	1,479770252

3.2.6 Implementierung des Geschwindigkeitsprofils für RTW

Um das Geschwindigkeitsprofil anwenden zu können muss zunächst eine OSRM-Profil-Datei im Format `.lua` erstellt werden. Als Basis wird `testbot.lua` verwendet, das sich bei bekannten Geschwindigkeiten anbietet. Hier müssen die OSM-Kategorien entsprechend der angegebenen Höchstgeschwindigkeit (*function getSpeed*) den neuen Straßenkategorien (*function getKategorie*) zugewiesen werden, wie in Tab. 15 dargestellt. Zudem erfolgt der Eintrag mit den RTW-Geschwindigkeiten je Straßenkategorie (vgl. Anhang V: `RTW_con.lua`).

Im Pre-Processing des OSRM-Docker-Image des Backend-Dienstes wird das Geschwindigkeitsprofil bei der Anwendung des OSRM-Tool Extractor (*osrm-extract*) eingesetzt, die rekursive Partitionierung des OSM-Graphen in Zellen (*osrm-partition*) und

Anpassungen (*osrm-customize*) vorgenommen werden, um den Multi-Level-Dijkstra-Routingalgorithmus nutzen zu können. Mit dem Tool *osrm-routed* wird der http-Server mit Port 5000 gestartet. Nun können die Fahrzeiten für die Einsatzfahrten mit dem Geschwindigkeitsprofil für Rettungswägen mit folgender URL berechnet werden:

<http://127.0.0.1:5000/route/v1/driving/>

Das Ziel ist es nun die Fahrzeit zwischen Punkt A und Punkt B abzuschätzen, auf Basis des geschätzten Geschwindigkeitsprofils (vgl. Anhang VI: *calc_fz.py*). Dazu werden die dokumentierten Fahrten mit dem neuen Geschwindigkeitsprofil geroutet. Das Ergebnis wird als .csv-Tabelle ausgegeben, die als *OUT_BY_FZ_V_RAW* in die SQLite-Datenbank geladen wird. Da alle Einsatzfahrten geroutet werden konnten, enthält die vollständige Tabelle 716.332 Datensätze.

Tab. 23: Datentypen der Tabelle *OUT_BY_FZ_V_RAW*

Spaltenname	Datentyp	PrimaryKey	Beschreibung
PK_UID	integer	X	Fortlaufende eindeutige Kennzeichnung der einzelnen Datensätze
FZ_EINSATZ	real		dokumentierte Fahrzeit aus Einsatzdaten
FZ_RTW	real		Fahrzeit [min] berechnet über das RTW-Geschwindigkeitsprofil
DIST_RTW	real		Fahrstrecke [km] aus dem Routingergebnis
V_RTW	real		Durchschnittsgeschwindigkeit [km/h], berechnet aus $(DIST_RTW/FZ_RTW)*60$
V_EINSATZ	real		Durchschnittsgeschwindigkeit [km/h], berechnet aus $(DIST_RTW/FZ_EINSATZ)*60$

Auch an diesen Daten wird eine Datenbereinigung durchgeführt, Ausreißer und unplausible Werte, die durch eine fehlerhafte Dokumentation der Zeitstempel und somit einer zweifelhaften Angabe der Einsatzfahrzeit entstehen, werden identifiziert und entfernt. Wie schon vorher, wird auch hier eingeschränkt auf Fahrten, deren Fahrzeit maximal 15 Minuten beträgt ($n = 672.697$), die Durchschnittsgeschwindigkeit bei maximal 130 km/h liegt ($n = 702.735$) und die Einsatzfahrzeit mindestens zwei Drittel der Routingfahrzeit, bzw. die Routingfahrzeit mindestens zwei Drittel der Einsatzfahrzeit beträgt ($n = 448.710$). Insgesamt finden sich 416.977 gültige Datensätze in der Tabelle.

Mit diesen Datensätzen wird abermals eine lineare Regression mit der bereits zuvor angewendeten Regressionsfunktion *sklearn.linear_model.LinearRegression(*, positive = True)* durchgeführt, um letztendlich die Gleichung zur Abschätzung der Fahrzeit zwischen zwei Punkten zu erhalten (vgl. Anhang VII: *reg_koeff.py*). Neben X und Y sind weitere Elemente der Regressionsgleichung die Steigung der Regressionsgeraden (Regressionskoeffizient b) und der Punkt, an welchem die Regressionsgerade die y-Achse schneidet (Regressionskonstante a).

$$y = bx + a$$

Um b und a zu erhalten, werden für X die dokumentierten Fahrzeiten eingesetzt und für Y die mit dem RTW-Profil gerouteten Fahrzeiten. Der Y-Achsenabschnitt an welchem x gleich Null ist, wird als Regressionskonstante, bzw. als Additionskonstante bezeichnet. Die geschätzte Fahrzeit nimmt diesen Wert an, wenn die Geschwindigkeit Null beträgt. Dieser Wert wird auf die Fahrzeit addiert, um eine Über- bzw. Unterschätzung dieser auszugleichen.

	Kennwerte	
y-Achsen-Abschnitt a	-0,643638	$\triangleq 38,61828$ Sekunden
Steigung b	1,114828	
R²	0,791495	

Mit den berechneten Kennwerten kann folgende Gleichung zur Berechnung der geschätzten Fahrzeit abgeleitet werden:

$$\hat{Y}_{\text{geschätzte Fahrzeit}} = 1,114828x_{\text{geroutete Fahrzeit}} - 0,643638 \text{ Minuten}$$

4 Erreichbarkeitsanalyse der Krankenhäuser

4.1 Erreichbarkeitsanalyse

4.1.1 Erreichbarkeitsmodell

Es gibt unterschiedliche Ansätze regionale Erreichbarkeiten zu berechnen. Ein Ansatz bezieht sich auf räumliche Interaktionsmodelle, wie das Gravitationsmodell, unter der Annahme, dass sich Versorgungsbedarf und Versorgungsmöglichkeit gegenseitig anziehen, oder die „Two-Step Floating Catchment Area“ Methode (2SFCA) von (Radke und Mu 2000)), bei welcher das Verhältnis von Angebot (z.B. die Anzahl der verfügbaren Betten oder Anzahl an Krankenhauspersonal) zu Nachfrage (Anzahl potentieller Patienten) an jedem Standort einer Gesundheitseinrichtung innerhalb einer kritischen Reisezeit als wichtiger Parameter in die Gewichtung eingeht (Kim et al. 2018). Weitere Ansätze berücksichtigen das prognostizierte Verkehrsgeschehen und die geographische Erreichbarkeit. In nachfolgender Tabelle werden verschiedene repräsentative Erreichbarkeitsmodelle mit unterschiedlichen Ansätzen gegenübergestellt.

Tab. 24: Gegenüberstellung unterschiedlicher Erreichbarkeitsmodelle (Li und Wang 2022)

Methode	Merkmale	Vorteile	Nachteile
Kürzeste Distanz	Entfernungen stellen hier den Indikator für die Erreichbarkeit dar.	Es ist ein einfacher und intuitiver Berechnungsprozess.	Das Angebots- und Nachfragepotenzials wird nicht berücksichtigt.
Gravitationsmodell	Die Erreichbarkeitsmodellierung basiert auf räumlicher Interaktion.	Interaktionsintensität erhöht sich mit zunehmender Nähe über die Einbeziehung einer kontinuierlichen Distanzabnahmefunktion mit Berücksichtigung von Angebots- und Nachfragepotenzial.	Es ist keine Einschränkung des Suchschwellenwertes von Einrichtungen vorgesehen.
Huff-Modell	Entfernungen und die Größenordnung der Einrichtung stellen hier den Indikator für die Erreichbarkeit dar.	Die Größenordnung der Einrichtungen und die Entfernung werden berücksichtigt.	Der Wettbewerb zwischen den Einrichtungen bleibt unbeachtet.
2SFCA	Versorgungs- und Nachfrageeinrichtungen werden jeweils als Basis für die Entfernungsberechnung eingesetzt. Innerhalb des Suchradius erfolgt eine Aufsummierung der	Es ist ein einfacher Berechnungsprozess, der die Anzahl und die Größenordnung des Angebots- und Nachfragepotenzials berücksichtigt, mit der Definition eines Suchschwellenwertes.	Der Entfernungseffekt wird nicht miteinbezogen.

Methoden	Merkmale	Vorteile	Nachteile
	Verhältnisse von Angebot zu Nachfrage.	Der errechnete Index ist einfach zu interpretieren.	
E2SFCA	Basierend auf der 2SFCA-Methode wird die Distanzabnahmefunktion einbezogen.	Für genauere Analyseergebnisse geht der Einfluss der Entfernung in die Berechnung mit ein.	Mögliche Probleme entstehen mit der Distanzabnahmefunktion und des Suchschwellenwerts.

Die Two-Step Floating Catchment Area (2SFCA) Methode, wie sie von Luo und Wang (2003) vorgestellt wurde, ist eine wichtige Methode in der Forschung zur räumlichen Erreichbarkeit öffentlicher Einrichtungen, insbesondere im Bereich des Gesundheitswesens, und stellt einen Sonderfall der Gravitationsmethode dar (Guagliardo 2004). Sie misst die räumliche Erreichbarkeit in zwei Schritten. Im ersten Schritt wird die Anzahl der Bevölkerung innerhalb eines definierten Suchradius (Fahrzeitschwellenwert) um die relevanten Krankenhäuser ermittelt, um so das Verhältnis von Angebot zu potenzieller Nachfrage an jedem Standort einer Gesundheitseinrichtung innerhalb der Grenze der kritischen Reisezeit zu berechnen. Dabei werden die verfügbaren Ressourcen der Krankenhäuser, z.B. Anzahl der verfügbaren Betten, der Anzahl der Bevölkerung in Beziehung gesetzt. Im anschließenden zweiten Schritt werden für jede regionale Einheit alle Angebot-zu-Nachfrage-Verhältnisse der relevanten Kliniken innerhalb des Fahrzeitschwellenwertes aufsummiert und so der Erreichbarkeitswert, der sogenannte Spatial Accessibility Index (SPAI), der regionalen Einheit abgeleitet (Vo et al. 2015).

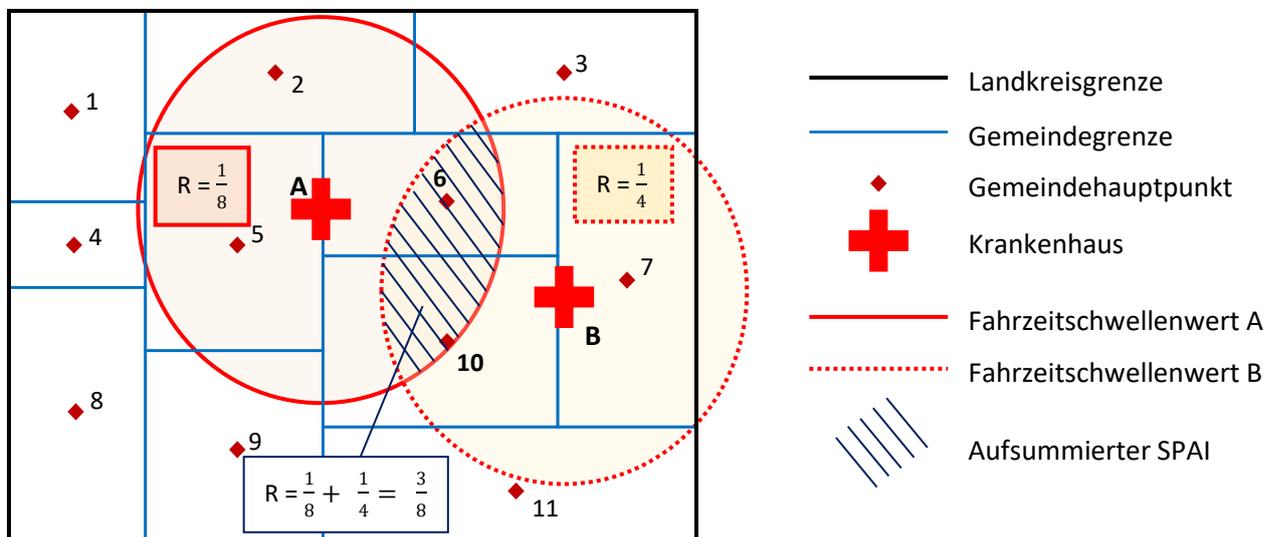


Abb. 19: Schematische Darstellung der 2-Step Floating Catchment Area (2SFCA) Methode

Es gibt verschiedene Erweiterungen der 2SFCA-Methode. Diese Erweiterungen umfassen kontinuierliche Funktionen, wie die Kerndichte (KD2SFCA) (Dai und Wang 2011) oder die Gaußsche Funktion, um Strafen bzw. Kosten zu berechnen, oder die Zuweisung von unterschiedlichen Gewichten zu diskreten Distanzabschnitten (E2SFCA), wobei weiter entfernte Abschnitte höhere Gewichte zugesprochen werden (Luo und Qi 2009). Neben dem Bezug auf die Gewichtungen, versuchen andere Erweiterungen den Suchradius auf bestimmte Erreichbarkeitsfaktoren, wie Bevölkerungsdichte und Zieleinrichtungshierarchie (Li und Wang

2022), anzupassen. Bei der 3SFCA-Methode wird die Konkurrenz unter Kliniken durch Verwendung eines Auswahlgewichts des Patienten berücksichtigt (Wan et al. 2012).

Um einige Einschränkungen der 2SFCA-Methode zu verbessern, wurde diese zur Enhanced 2-Step Floating Catchment Area Methode (E2SFCA) von Luo und Qi (2009) weiterentwickelt. Dabei liegt der Schwerpunkt in der Berücksichtigung des Entfernungseffekts, indem diskreten Erreichbarkeitszonen unterschiedliche Gewichtungen zugewiesen werden, in Abhängigkeit von der Entfernung zur Gesundheitseinrichtung. Es wird angenommen, dass die Präferenz für eine Einrichtung bei weiterer Entfernung abnimmt, entsprechend Tobler's erstem Gesetz der Geographie, bei welchem „everything is related to everything else, but near things are more related than distant things“ (Tobler 1970). Im ersten Schritt werden zunächst Fahrzeitzonen von einer Gesundheitseinrichtung ausgehend festgelegt und die Gesamtbevölkerung der einzelnen Zonen berechnet. So kann das Verhältnis von Anbieter zu Nachfragenden erhalten werden. Die definierten Erreichbarkeitszonen werden im zweiten Schritt auf die regionale Einheit angewendet und das Verhältnis von Anbieter zu Bevölkerung jeder Gesundheitseinrichtung, die in der entsprechenden Zone liegt, übertragen. Die nach Zone gewichtete Summe daraus ergibt den Spatial Accessibility Index.

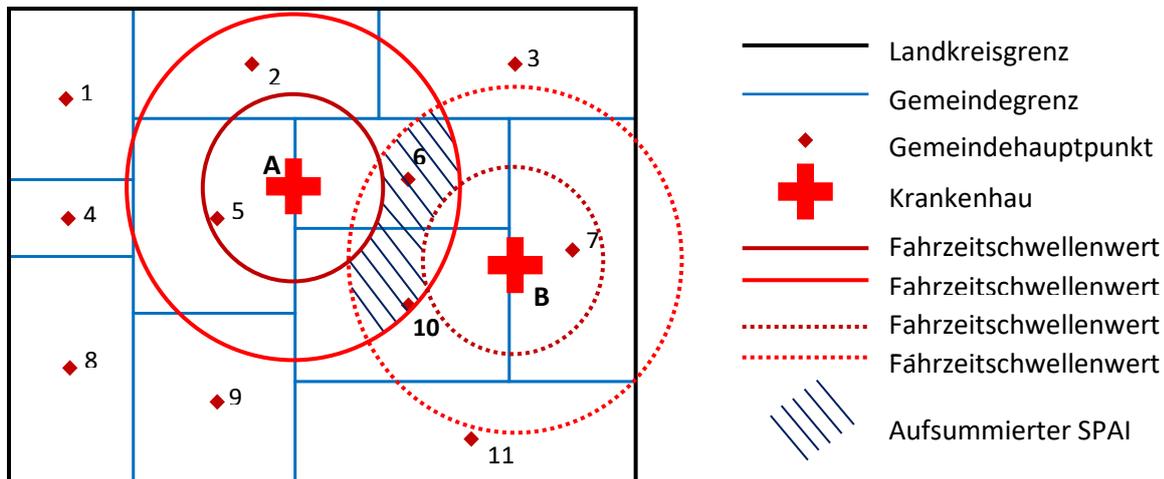


Abb. 20: Schematische Darstellung der Enhanced 2-Step Floating Catchment Area (E2SFCA) Methode

Für die Berechnung und Beurteilung der Erreichbarkeit von Krankenhäuser bei Verdacht auf eine Tracer-Diagnose, wird die um Erreichbarkeitszonen erweiterte Two-Step Floating Catchment Area Methode angewendet.

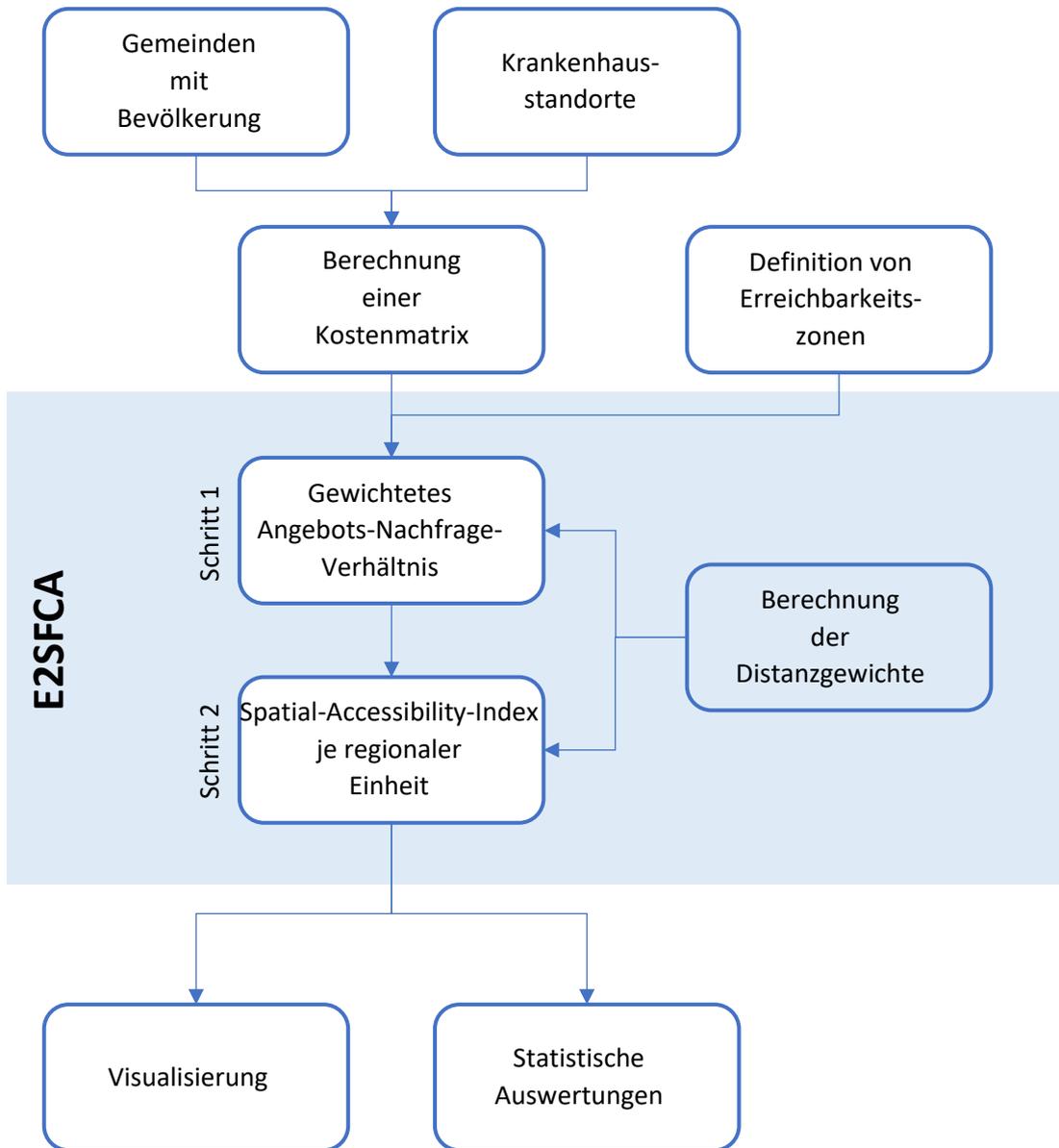


Abb. 21: Schematische Darstellung der Berechnung der Krankenhäuserreichbarkeiten

4.1.2 Eingangsparameter des Erreichbarkeitsmodells E2SFCA

A) Berechnung einer Kostenmatrix

Als Grundlage zur Berechnung einer Kostenmatrix bilden die Gemeindegrenzen und die Krankenhausstandorte. Da ein Routing von Polygon zu Koordinatenpunkt nicht möglich ist, müssen die Gemeindegrenzen auf einen Punkt abgeleitet werden. Wie in 2.4.1 beschrieben, wird für die Gemeinden der Punkt des Hauptgemeindeteils verwendet. Diese sind allerdings für kreisfreie Städte und gemeindefreie Gebiete nicht verfügbar. Bei gemeindefreien Gebiete wird der Polygonmittelpunkt, der sich innerhalb der Geometrie befinden muss, berechnet. Dieses Vorgehen ist bei kreisfreien Städten nicht anwendbar, da im Schwerpunkt des Polygons nicht grundsätzlich das Stadtzentrum zu finden ist. Bei lediglich 25 kreisfreien Städten wird der Referenzpunkt manuell digitalisiert und auf den Vorplatz des jeweiligen Hauptbahnhofs gesetzt.

Tab. 25: Referenzpunkte der regionalen Einheit

Regionale Einheit	Anzahl Datensätze	Referenzpunkt
Gemeinde	n = 2.031	Schwerpunkt des Hauptgemeindeteils
Gemeindefreies Gebiet	n = 173	Schwerpunkt
Kreisfreie Stadt	n = 25	Zentraler Punkt – Hauptbahnhof Vorplatz
Gemeindeebene gesamt	n = 2.229	

Zunächst wird das kartesische Produkt aus Gemeinden (n = 2.229) und Krankenhausstandorten (n = 148) gebildet, wobei noch nicht nach der Eignung der Kliniken zur Behandlung von Tracer-Diagnosen differenziert wird. Für jede Koordinatenkombination wird die Fahrzeit, entsprechend der aufgestellten Regressionsgleichung, abgeschätzt und die Entfernung berechnet (vgl. Anhang VIII: DM_GEM_KH.py). In der daraus entstandenen Distanzmatrix sind 329.892 Relationen enthalten.

Tab. 26: Anzahl der Relationen je Tracer-Diagnose bei 2.229 Gemeinden

Tracer-Diagnose	Krankenhäuser <i>Anzahl</i>	Relationen Distanzmatrix
Reanimation	88	196.152
Polytrauma	108	240.732
SHT	38	84.702
STROKE	90	200.610
STEMI	73	162.717
Sepsis	109	242.961

B) Definition von Erreichbarkeitszonen

Patienten mit zeitkritischer Tracer-Diagnosen sollten spätestens 60 Minuten nach dem Absetzen des Notrufs einer innerklinischen Diagnostik und Therapie zugeführt werden. Auswertungen der dokumentierten Einsatzdaten der Leitstellen durch das INM zeigen, dass ab Notrufeingang über die Alarmierung der Einsatzkräfte und Ankunft am Einsatzort, bis zur ersten Versorgung des Patienten am Einsatzort etwa eine halbe Stunde vergehen. Um die Prähospitalphase innerhalb der *golden hour* zu erfüllen, darf das Transportintervall, also das Zeitintervall von Abfahrt vom Einsatzort bis zur Ankunft in der nächsten geeigneten Klinik (vgl. Abb. 2), maximal 30 Minuten betragen (Bayeff-Filloff et al. 2014).

Signifikanten Einfluss auf die Überlebenswahrscheinlichkeit, bzw. auf das Ausmaß der körperlichen Schädigung bei Schwerverletzten, hat die Transportzeit. Daher werden Fahrzeitzone im Zehn-Minuten-Intervall, mit einem kritischen maximalen Fahrzeitschwellenwert von 30 Minuten festgelegt:

0 bis 10 Minuten ~ mehr als 10 bis 20 Minuten ~ mehr als 20 bis 30 Minuten

4.1.3 Berechnung der Krankenhäuserreichbarkeit nach E2SFCA

A) Ermittlung der Distanzgewichte

Zur Gewichtung der einzelnen Erreichbarkeitszonen (0-10, 10-20 und 20-30 Minuten) wird ein Distanzreibungskoeffizient auf Basis der Gauß'schen Gewichtungsfunktion unter der Bedingung $f(d_{\max}) \cong 0,01$ hergeleitet. Dies wird als kritischer Wert angesehen, wenn die Gauß'sche Funktion gegen Null läuft, also das Distanzgewicht bis zum kritischen Schwellenwert der maximalen Distanz zu Null tendiert. Somit erhalten größer werdende Distanzen kleiner werdende Distanzgewichte innerhalb der Maximaldistanz d_{\max} und sämtliche Standorte außerhalb des Fahrzeitschwellenwertes ein Distanzgewicht von Null.

$$f(d_{\max}) \cong 0,01 \rightarrow e^{-\frac{(d_{\max})^2}{\beta}} = 0,01$$

Bei einer maximalen Distanz d_{\max} von 30 Minuten beträgt der Distanzreibungskoeffizient $\beta = 195,43$. Anhand der mittleren Distanz der einzelnen Erreichbarkeitszonen können nun die Distanzgewichte W_r berechnet werden:

Tab. 27: Distanzgewichte der Erreichbarkeitszonen

0 – 10 Minuten	$W_{r=1} = f(5) = e^{-\frac{(5)^2}{195,43}}$	0,88
10 – 20 Minuten	$W_{r=2} = f(15) = e^{-\frac{(15)^2}{195,43}}$	0,316
20 – 30 Minuten	$W_{r=3} = f(25) = e^{-\frac{(25)^2}{195,43}}$	0,041

B) Schritt 1 – gewichtetes Angebots-Nachfrage-Verhältnis

In Schritt 1 der Enhanced 2-Step Floating Catchment Area Methode wird ein Angebot-Nachfrage-Verhältnis für jede Gesundheitseinrichtung ermittelt. Dabei erfolgt die Betrachtung der Krankenhäuser unabhängig voneinander, somit ist die Eignung für die Behandlung einer Tracer-Diagnose an dieser Stelle noch nicht relevant.

Formel zur Berechnung des Angebot-Nachfrage-Verhältnisses bei drei Fahrzeitzonen nach (Luo und Qi 2009):

$$R_j = \frac{S_j}{\sum_{k \in \{d_{kj} \leq d_0\}} P_k W_r} = \frac{S_j}{\sum_{k \in \{d_{kj} \in d_1\}} P_k W_1 + \sum_{k \in \{d_{kj} \in d_2\}} P_k W_2 + \sum_{k \in \{d_{kj} \in d_3\}} P_k W_3}$$

R_j = Angebot-Nachfrage-Verhältnis

S_j = Versorgungskapazität von Gesundheitseinrichtung j

P_k = Bevölkerung der regionalen Einheit k

d_0 = Maximaldistanz

d_{kj} = Distanz zwischen k und j

W_r = Distanzgewicht der Fahrzeitzone r (r = 1-3)

Für alle Krankenhäuser wird angenommen, dass die benötigten Ressourcen im Notfall jederzeit rund um die Uhr abrufbereit sind. Das Angebot je Gesundheitseinrichtung, also die verfügbare Versorgungskapazität S_j , wird auf 1 gesetzt, da die reale Anzahl an Notfallbetten unbekannt ist. Die Bevölkerungsinformationen P_k je Gemeinde, gemeindefreie Gebiete und kreisfreie Städte werden aus der amtlichen Statistik herangezogen. Aus der berechneten Distanzmatrix zwischen Gemeinden und Krankenhausstandorten werden die Werte für d_{kj} entnommen, mit einem Fahrzeitschwellenwert d_0 von 30 Minuten.

In nachfolgender Tab. 28 finden sich Beispieldatensätze, die das Angebot-Nachfrage-Verhältnis R für ausgewählte Kliniken zeigen. Zusätzlich ist die Gesamtzahl der Bevölkerung P_GES , die im 30-Minuten Einzugsbereich der jeweiligen Klinik liegt, sowie die Differenzierung der Gesamtbevölkerung nach definierten Erreichbarkeitszonen (P10, P20, P30) angegeben. Die gewichtete Einwohnerzahl nach Fahrzeitzone, aus welcher sich R berechnet, ist ebenso dargestellt.

Tab. 28: Beispieldatensätze für Krankenhäuser mit berechnetem R-Wert

ΣP_GES ($d_{kj} \leq 30$ Min.)	$\Sigma P10$ ($d_{kj} \leq 10$ Min.)	$\Sigma P20$ ($10 < d_{kj} \leq 20$)	$\Sigma P30$ ($20 < d_{kj} \leq 30$)	$\Sigma PW10$ ($W = 0,88$)	$\Sigma PW20$ ($W = 0,316$)	$\Sigma PW30$ ($W = 0,041$)	R
<i>roMed-Klinik Prien am Chiemsee:</i>							
291.040	22.386	44.445	224.209	19.699,68	14.044,62	9.192,569	0,000023
<i>Klinikum Passau:</i>							
249.212	53.498	38.451	157.263	47.078,24	12.150,516	6.447,783	0,000015
<i>LMU Klinikum München – Großhadern:</i>							
2.307.491	41.764	1.680.55	585.177	36.752,32	531.053,8	23.992,257	0,000002

Der R-Wert stellt das Verhältnis von einer Gesundheitseinrichtung zur Gesamtzahl der Bevölkerung, also den potenziellen Patienten innerhalb des kritischen Fahrzeitschwellenwertes von 30 Minuten dar und fällt somit sehr klein aus. Je höher dieser Wert ist, umso weniger potenzielle Patienten wohnen im Einzugsbereich der Klinik.

C) Schritt 2 – Spatial Accessibility Index

Im Schritt 2 der weiterentwickelten Two-Step Floating Catchment Area Methode wird der Erreichbarkeitsindex für jede regionale Einheit berechnet. Da das Angebot-Nachfrage-Verhältnis R aller Krankenhausstandorte, die innerhalb der kritischen Fahrzeit liegen, je Gemeinde aufsummiert und nach Erreichbarkeitszone gewichtet werden, muss die Krankenhausgrundgesamtheit auf die Relevanz nach Tracer-Diagnose eingeschränkt werden. Insgesamt wird so der Spatial Accessibility Index sechsmal, je Tracer-Diagnose, für alle Gemeinden berechnet.

Formel zur Berechnung des Spatial Accessibility Index bei drei Fahrzeitzone nach (Luo und Qi 2009):

$$A_i(T) = \sum_{j \in \{d_{ij} \in d_r\}} R_j W_r = \sum_{j \in \{d_{ij} \in d_1\}} R_j W_1 + \sum_{j \in \{d_{ij} \in d_2\}} R_j W_2 + \sum_{j \in \{d_{ij} \in d_3\}} R_j W_3$$

$A_i(T)$ = Erreichbarkeitsindex an regionaler Einheit i nach Tracer-Diagnose

R_j = Angebot-Nachfrage-Verhältnis an Gesundheitseinrichtung j

d_{ij} = Distanz zwischen i und j

W_r = Distanzgewicht der Fahrzeitzone r ($r = 1-3$)

Beispielhaft für ausgewählte Gemeinden wird in Tab. 29 der R-Wert für alle Krankenhäuser, die in Fahrzeitzone 0-10 Minuten je Gemeinde liegen, aufsummiert dargestellt. Entsprechend finden sich die summierten R-Werte für Kliniken in den Fahrzeitzonen 10-20 Minuten und 20-30 Minuten. RW gibt die entsprechend der Erreichbarkeitszone gewichtete R-Summe an. Der Spatial Accessibility Index für beispielhaft ST-Hebungsinfarkt, der sich aus der Summe der gewichteten R-Werte je Fahrzeitzone errechnet, ist für die entsprechende Gemeinde in der letzten Spalte angegeben.

Tab. 29: Beispieldatensätze mit berechnetem Spatial Accessibility Index nach Tracer-Diagnose

ΣR_{10} ($d_{ij} \leq 10$ Min.)	ΣR_{20} ($10 < d_{ij} \leq 20$)	ΣR_{30} ($20 < d_{ij} \leq 30$)	ΣRW_{10} ($W = 0,88$)	ΣRW_{20} ($W = 0,316$)	ΣRW_{30} ($W = 0,041$)	A_STEMI
<i>Naiba</i>						
0,000003	0,000027	0,000019	0,000026	0,000009	0,000001	0,000036
<i>Erding</i>						
0,000011	0	0,000029	0,00001	0	0,000001	0,000011
<i>Prien am Chiemsee</i>						
0	0	0,000026	0	0	0,000001	0,000001
<i>Cham</i>						
0	0	0	0	0	0	0

Ist in einer Spalte eine 0 zu finden, so konnte innerhalb der angegebenen Erreichbarkeitszone kein Krankenhaus gefunden werden. Am Beispiel Erding liegt somit kein Krankenhaus zwischen 10 und 20 Minuten Entfernung. Von der Gemeinde Cham ausgehend kann kein Krankenhaus innerhalb von 30 Minuten Fahrzeit erreicht werden. Dabei gilt, je größer der Spatial Accessibility Index ausfällt, umso besser ist die Erreichbarkeit von einer oder mehr Kliniken innerhalb einer Fahrzeit von 30 Minuten, bei entsprechender Tracer-Diagnose. Der Erreichbarkeitsindex erhöht sich mit zunehmender Anzahl an erreichbaren Krankenhäusern, einer geringeren Nachfrage, also bei niedrigen Einwohnerzahlen in den Gemeinden der Erreichbarkeitszonen, und mit zunehmender Nähe von Krankenhausstandorten (Angebotsstandort) zu Gemeinden (Nachfragestandort).

4.2 Auswertung der Krankenhauserreichbarkeit

4.2.1 Ergebnisse zur Versorgung der Bevölkerung nach Tracer-Diagnose

A) Plötzlicher Kreislaufstillstand

Das Ergebnis aus der Enhanced 2-Step Floating Catchment Area Methode wird für die Erreichbarkeit von Kliniken mit einer Eignung zur Behandlung von Patienten mit einem plötzlichen Kreislaufstillstand und einer erforderlichen Reanimation in nachfolgender Abb. 22 dargestellt. Liegt der Spatial Accessibility Index bei 0, so sind Krankenhäuser von entsprechenden Gemeinden ausgehend nicht innerhalb von 30 Minuten erreichbar. Ist eine RTW-Fahrzeit von maximal 30 Minuten möglich, so sind die Gemeinden grün abgestuft eingefärbt, wobei gilt, je dunkler das Grün, umso besser der Erreichbarkeitsindex. Insgesamt

werden 88 Krankenhäuser als geeignet zur Behandlung reanimationspflichtiger Patienten klassifiziert.

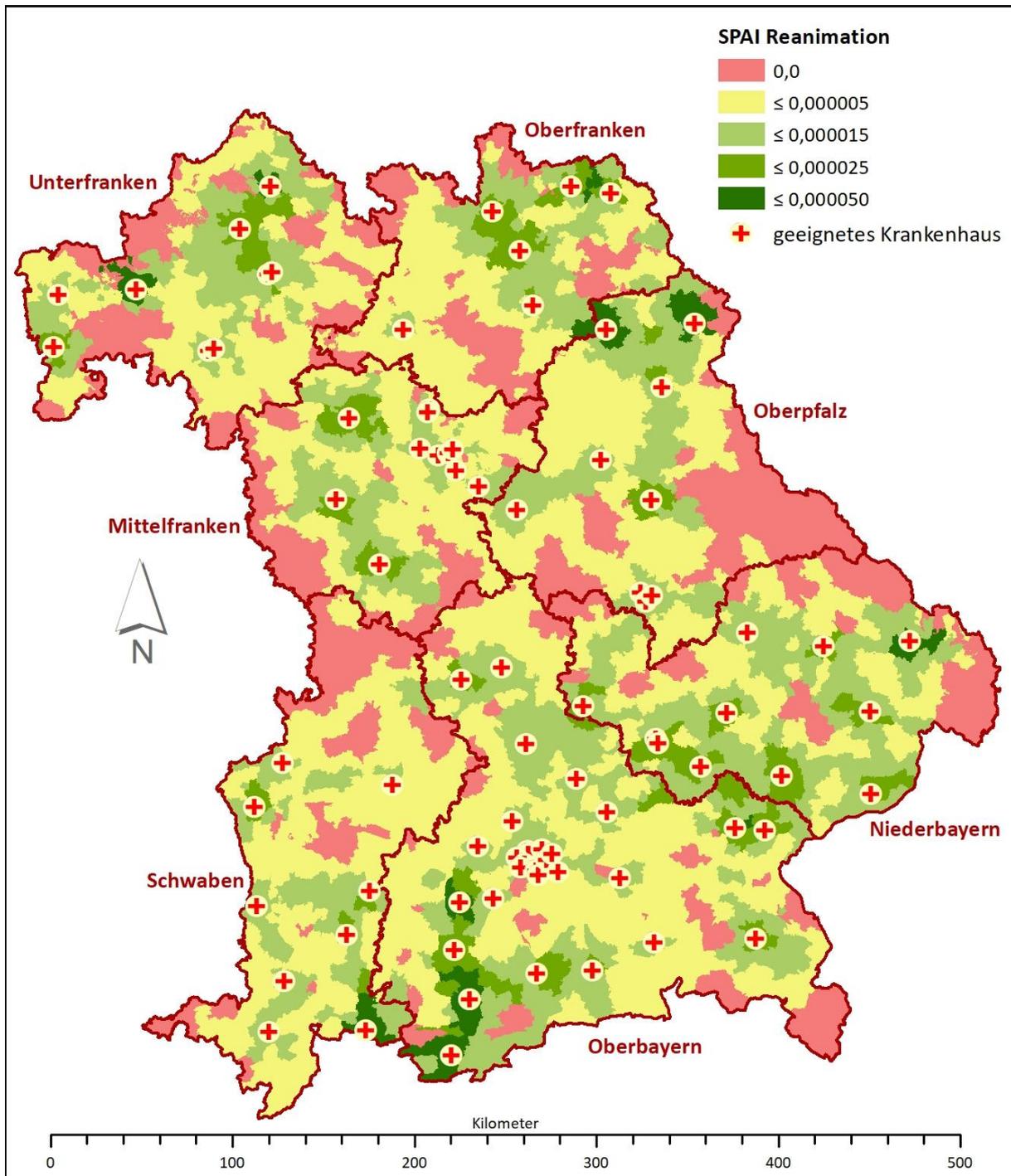


Abb. 22: Spatial Accessibility Index (SPAI) bei Reanimation

Bayernweit können fünf Regionen mit dem höchsten Grad an Erreichbarkeit identifiziert werden. Trotz der weit verteilten Klinikstandorte in Unterfranken und Oberpfalz ist der Erreichbarkeitsindex um die Klinik Main-Spessart in Lohr am Main (0,000028) und dem Krankenhaus Kemnath (0,000026) sehr hoch. Dies lässt darauf schließen, dass die Bevölkerungszahlen in entsprechenden Gemeinden im Verhältnis zum Angebot der Kliniken verhältnismäßig gering sind. Somit gibt es eine geringere Anzahl an potenziellen Patienten,

die die Klinikkapazitäten im Einzugsgebiet nutzen würden. Mit 0,00005 weisen die Gemeinden Leupoldsgrün und Köditz in Oberfranken den höchsten SPAI auf. Diese Gemeinden liegen in der Erreichbarkeitszone bis 10 Minuten bezogen jeweils auf die Sana Klinik in Hof und dem Klinikum Naila. Kurze Fahrzeiten begünstigen hier den hohen Grad an Erreichbarkeit. Ebenfalls kurze Fahrzeiten und eine höhere Dichte an geeigneten Kliniken weisen Gemeinden im Südosten Oberbayerns auf. Hier sind hohe SPAI-Werte von 0,000033 in Garmisch-Partenkirchen bis zu 0,000045 in Spatzenhäusern, zwischen Murnau, mit der Berufsgenossenschaftlichen Unfallklinik und dem Klinikum Garmisch-Murnau, und Weilheim, mit dem Klinikum Weilheim, gelegen, zu verzeichnen. Insgesamt lässt sich für Niederbayern flächendeckend einen guten Erreichbarkeitsindex von Kliniken konstatieren, besonders im Bayrischen Wald im Einzugsgebiet des Kreiskrankenhauses Grafenau. 50,3% der Gemeinden Niederbayerns zeigen einen SPAI von mehr als 0,000015. In der Landeshauptstadt München sind mit 13 die meisten geeigneten Kliniken in geringer Entfernung zueinander angesiedelt. Dennoch liegt hier der SPAI bei lediglich 0,000008. Dies ist auf den geringen R-Wert der Kliniken zurückzuführen, der durch die hohe Einwohnerzahl von bis zu 2.449.824 bedingt ist.

Die größten Schwachstellen in der Erreichbarkeit finden sich vor allem im Norden des Regierungsbezirks Schwaben, an der Grenze zur Tschechischen Republik in der Oberpfalz und im Süden von Berchtesgaden in Oberbayern. Hier lassen sich strukturelle Defizite identifizieren, die zu einer höheren RTW-Fahrzeit in die nächste geeignete Klinik führen.

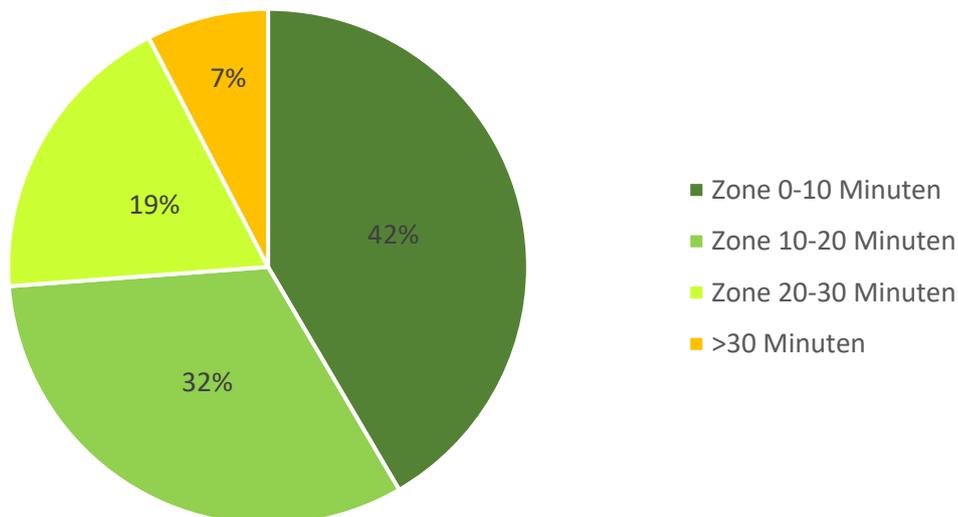


Abb. 23: Verteilung der Bevölkerung nach Erreichbarkeit der nächstgelegenen Klinik bei Reanimation

Mit 93% ist es für einen Großteil der Bevölkerung bei laufender Reanimation möglich das nächste geeignete Krankenaus zu erreichen. Für 42% liegt diese Klinik in einer Entfernung von maximal 10 Minuten. Lediglich 7% der bayrischen Bevölkerung wird mit dem Rettungswagen in über einer halbe Stunde in die nächste geeignete Klinik transportiert. Dies betrifft somit 1.013.098 Einwohner.

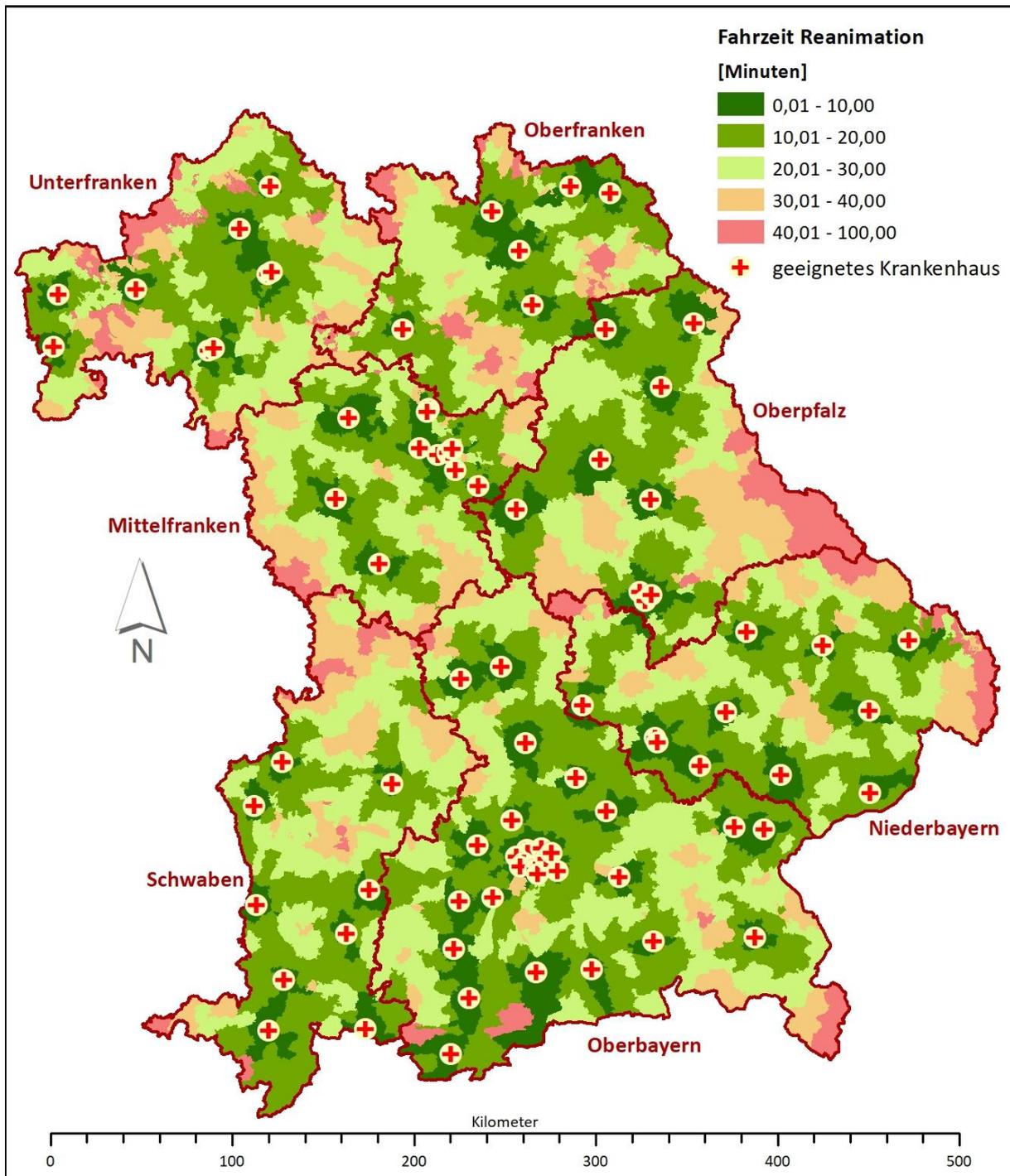


Abb. 24: Fahrzeiten von Gemeinden in die nächste geeignete Klinik bei Reanimation

Regionen mit den höchsten Fahrzeiten von über 40 Minuten können im Grenzbereich zur Tschechischen Republik im Osten der Oberpfalz (Gemeinden mit Fahrzeiten von bis zu 80 Minuten), sowie in Grenzgemeinden nach Baden-Württemberg im Westen von Mittelfranken und nach Thüringen und Hessen im Norden von Ober- und Unterfranken festgestellt werden. Längere Fahrzeiten von bis zu 69 Minuten treten im Forstgebiet im östlichen Oberfranken an der Grenze zur Oberpfalz auf.

In nachfolgender Abb. 25 wird der Anteil der Bevölkerung aufgezeigt, der die nächstgelegene Klinik nach Minuten-Entfernung erreicht, differenziert nach den siedlungsstrukturellen Raumtypen.

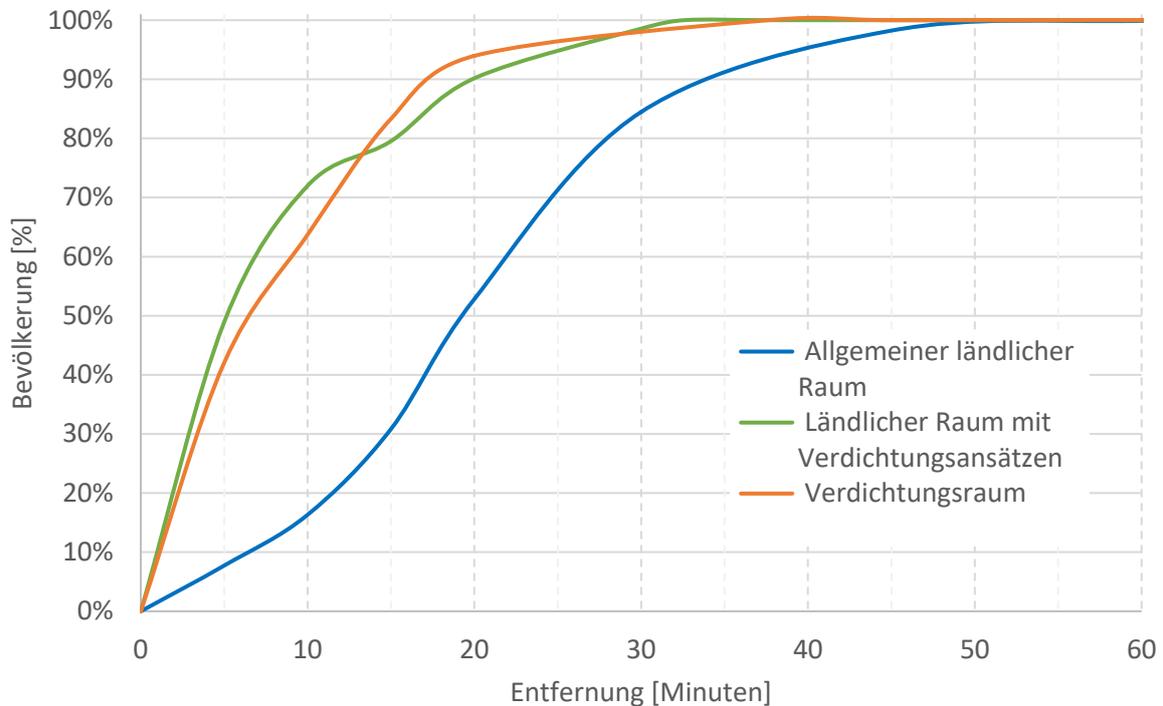


Abb. 25: Bevölkerung nach Entfernung zum nächstgelegenen geeigneten Krankenhaus bei Reanimation

In 30 Minuten Fahrzeit gelangen im ländlichen Raum 84% der Patienten mit dem Rettungswagen in die nächstgelegene Klinik, hingegen im ländlichen Raum mit Verdichtungsansätzen 99%. Innerhalb von knapp 38 Minuten werden Kliniken von 100% der Einwohner im Verdichtungsraum erreicht, im Übergangsgebieten liegt dieser Wert bei 33 Minuten. Etwa 98% der Einwohner benötigen 45 Minuten im ländlichen Raum, um in die nächste Klinik mit dem Rettungswagen gefahren zu werden, sämtliche Einwohner in 79 Minuten. Einen Überblick über die Verteilung der Raumtypen findet sich im Anhang IX: Karte Landesentwicklungsprogramm.

Differenziert nach Raumtypen werden nachfolgend Lage- und Streuungsparameter der gerouteten Fahrzeiten von Rettungswägen in Form eines Boxplots dargestellt.

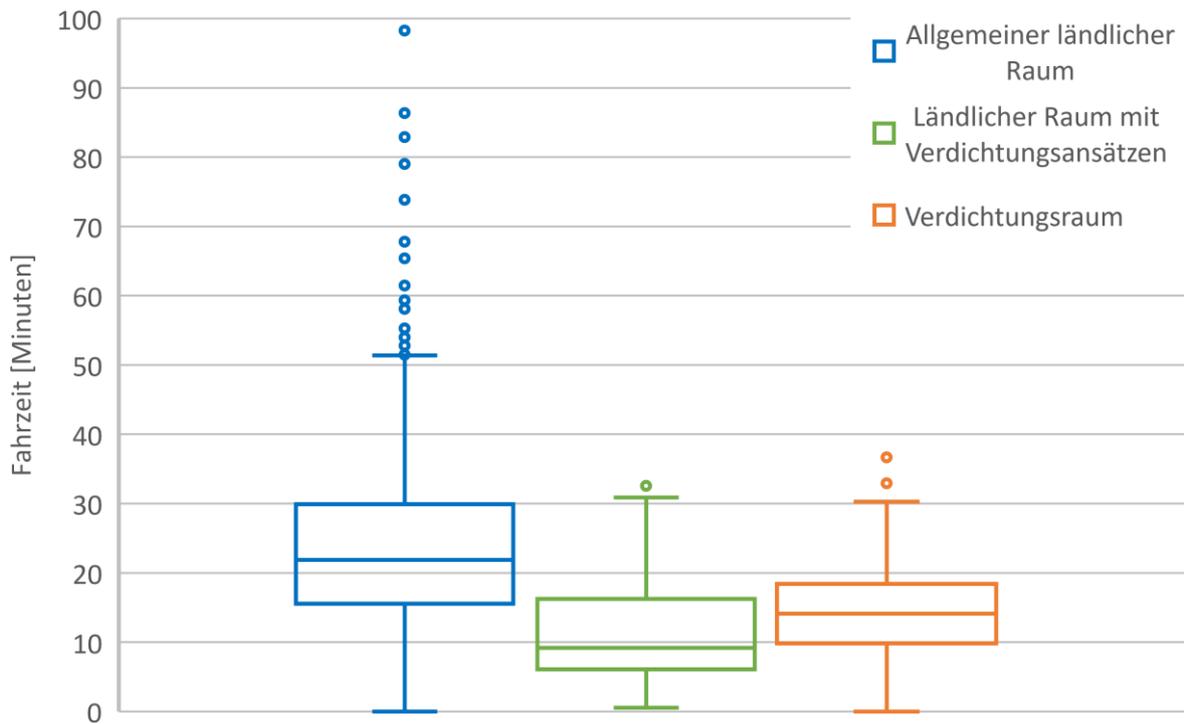


Abb. 26: Verteilung der Fahrzeiten zum nächstgelegenen Krankenhaus bei Reanimation

Es ist ersichtlich, dass die Fahrzeiten im ländlichen Raum stärker streuen als in den beiden weiteren Raumtypen, mit einer höheren Anzahl an Ausreißern. Der Median liegt hier bei 21,9 Minuten, im Vergleich dazu im Verdichtungsraum bei 14,2 Minuten und im Übergangsbereich bei 9,2 Minuten. Die niedrigsten Werte für das erste (6,1 Minuten) und dritte Quartil (16,3 Minuten) weisen Fahrzeiten von Gemeinden im ländlichen Raum mit Verdichtungsansätzen ausgehend auf.

B) Schlaganfall

Zur Behandlung von Patienten mit dem Verdacht auf Schlaganfall werden 90 Kliniken als geeignet betrachtet.

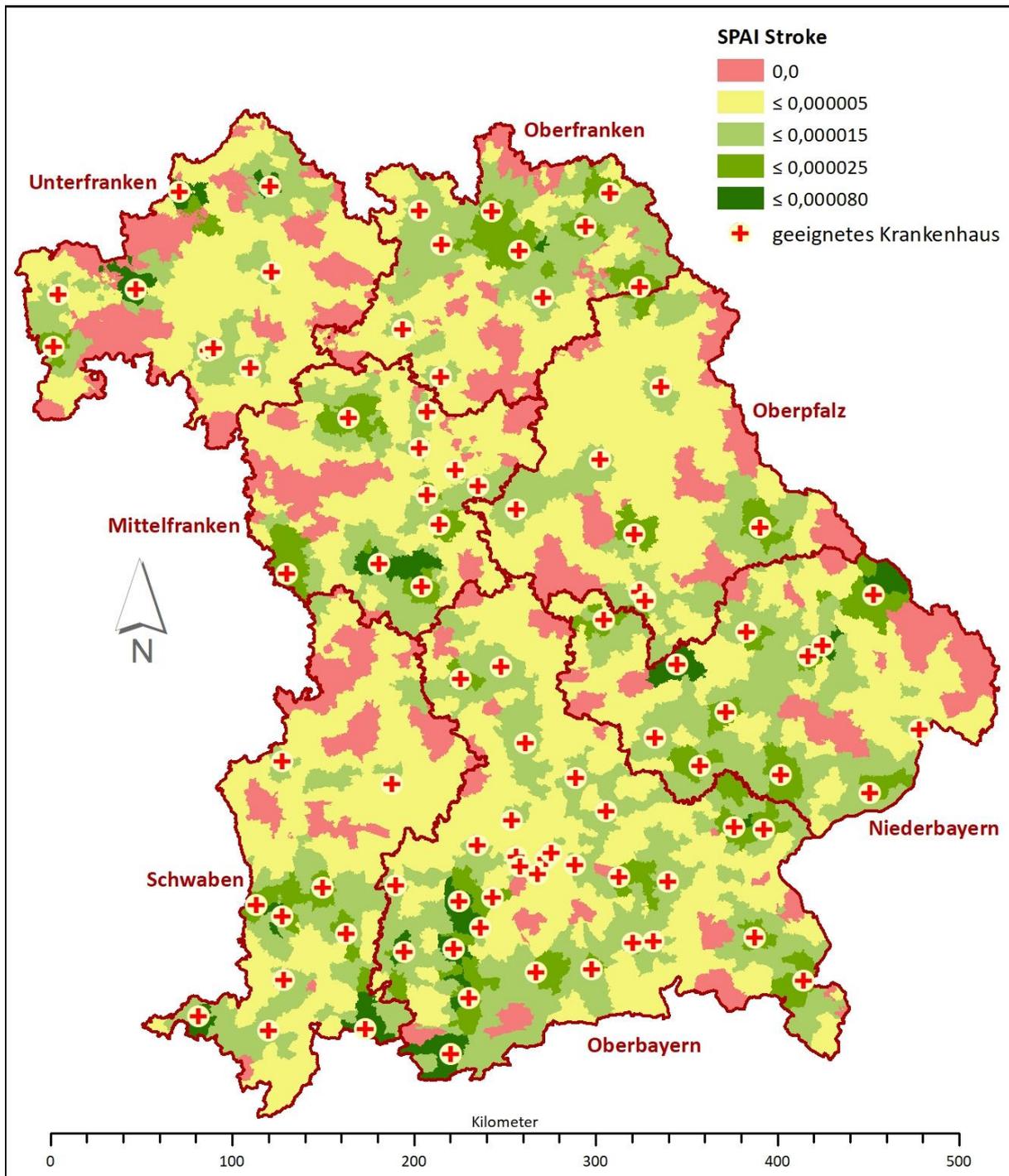


Abb. 27: Spatial Accessibility Index (SPAI) bei Schlaganfall

Der Erreichbarkeitsindex bei Schlaganfall zeigt die besten Ergebnisse in Niederbayern in der Gemeinde Mallersdorf-Pfaffenberg und den Gemeinden im näheren Umfeld der Arberlandklinik Zwiesel im Bayrischen Wald. Sehr gute Index-Werte sind im Norden Unterfrankens im Bereich der Klinik Bad Brückenau in Lohr am Main sowie im Süden Mittelfrankens im Einzugsbereich der Kliniken Altmühlfranken Gunzenhausen und Weißenburg zu konstatieren. Auch der Südwesten Oberbayerns und Gemeinden im Osten Oberfrankens weisen hohe Index-Werte auf. Defizite in der Erreichbarkeit finden sich in der Oberpfalz und Niederbayern an der Grenze zur Tschechischen Republik sowie in weiteren

Bereichen in Unterfranken. Zentralbayrische Gemeinden mit einem SPAI-Wert von 0 liegen nicht zusammenhängend als Inseln zwischen weiter entfernt befindlichen Krankenhäusern.

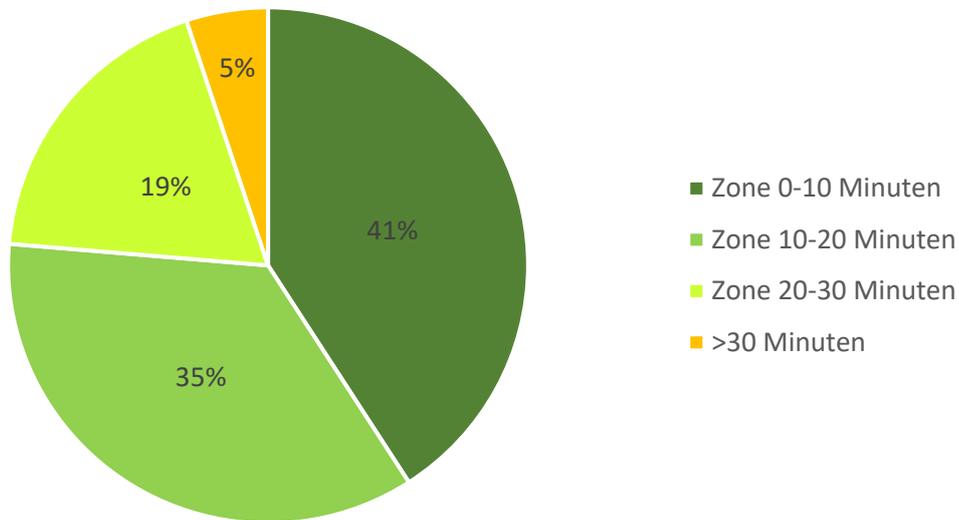


Abb. 28: Verteilung der Bevölkerung nach Erreichbarkeit der nächstgelegenen Klinik bei Schlaganfall

Mit einer Fahrzeit von über 30 Minuten erreichen 5% (679.548) der Einwohner das nächstgelegene Krankenhaus. Hingegen liegen 12.671.060 Einwohner innerhalb eines Einzugsgebiets von 30 Minuten, insgesamt 41% sogar in einem Fahrzeitradius von maximal 10 Minuten.

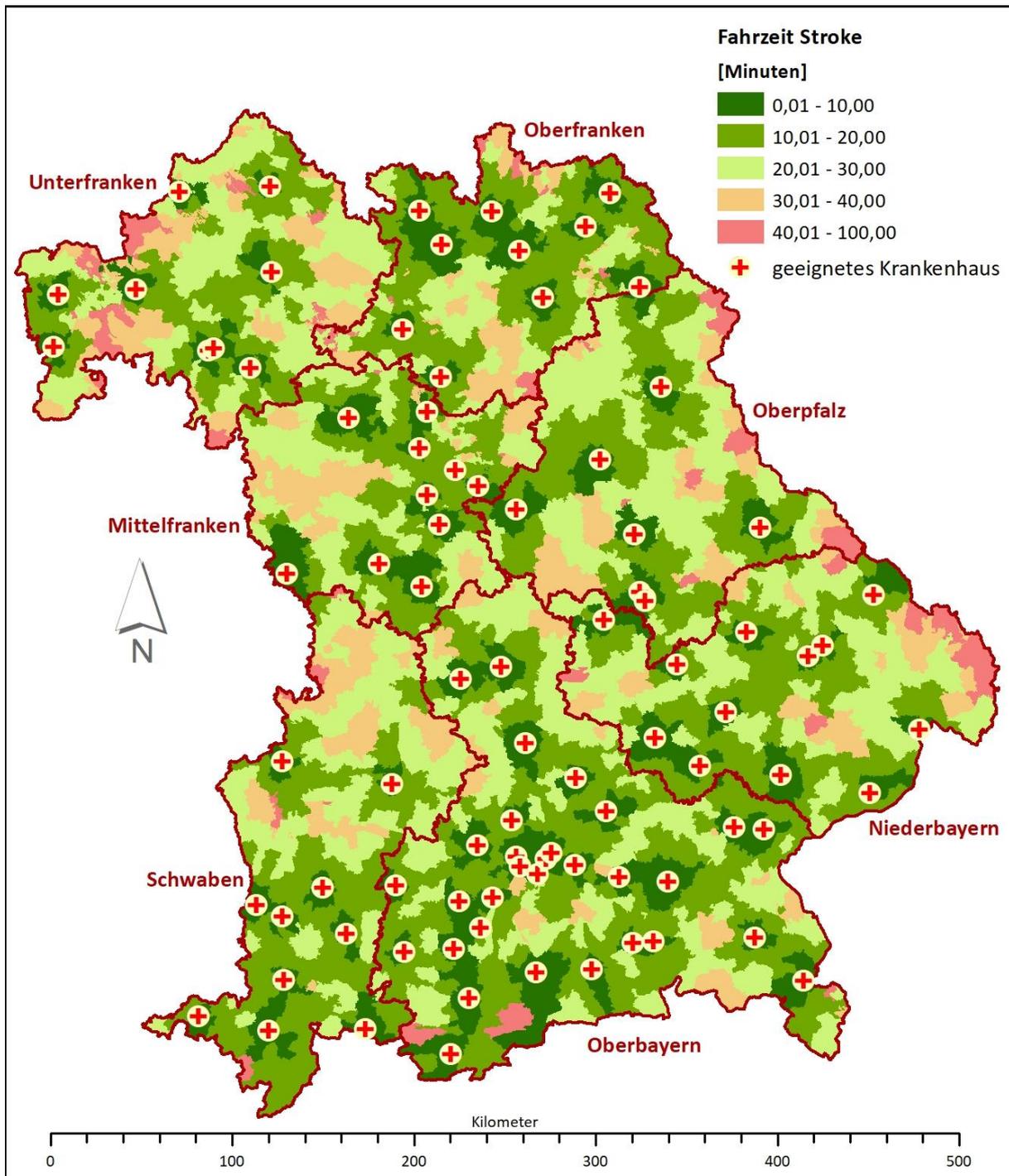


Abb. 29: Fahrzeiten von Gemeinden in die nächste geeignete Klinik bei Schlaganfall

Fahrzeiten von bis zu 90 Minuten in die nächstgelegene geeignete Klinik zeigen Gemeinden im niederbayerischen Bereich des Bayrischen Waldes sowie im unterfränkischen Spessart. Dadurch, dass viele geeignete Krankenhäuser regional verteilt zu finden sind, können in weiten Teilen Bayerns kurze Fahrzeiten konstatiert werden. Mit wenigen Ausnahmen wird flächendeckend von den Gemeinden Oberbayerns und Oberfrankens ausgehend die nächstgelegene Klinik innerhalb von einer halben Stunde erreicht.

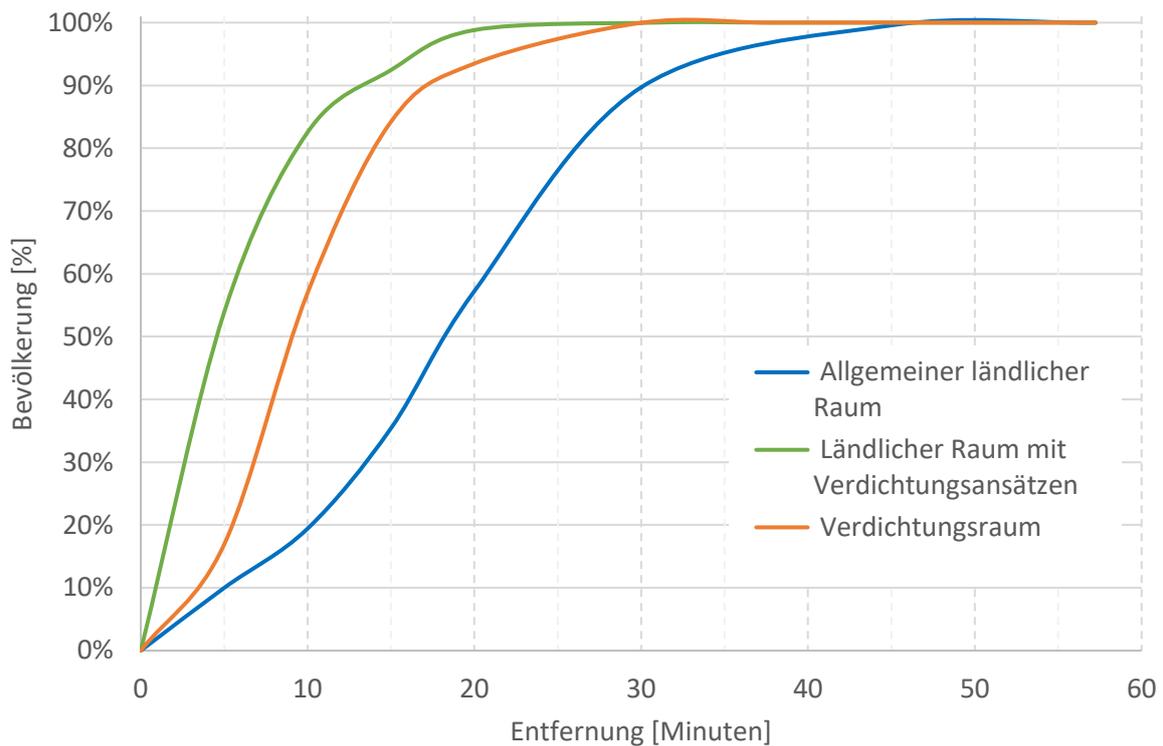


Abb. 30: Bevölkerung nach Entfernung zum nächstgelegenen geeigneten Krankenhaus bei Schlaganfall

Alle Einwohner von Gemeinden, die im ländlichen Raum mit Verdichtungsansätzen liegen, erreichen die nächstgelegene Klinik innerhalb von knapp 31 Minuten. In etwa 38 Minuten erreichen auch die Bewohner von Verdichtungsräumen das nächstgelegene Krankenhaus. Lediglich im ländlichen Raum beträgt die Fahrzeit bis zu 45 Minuten.

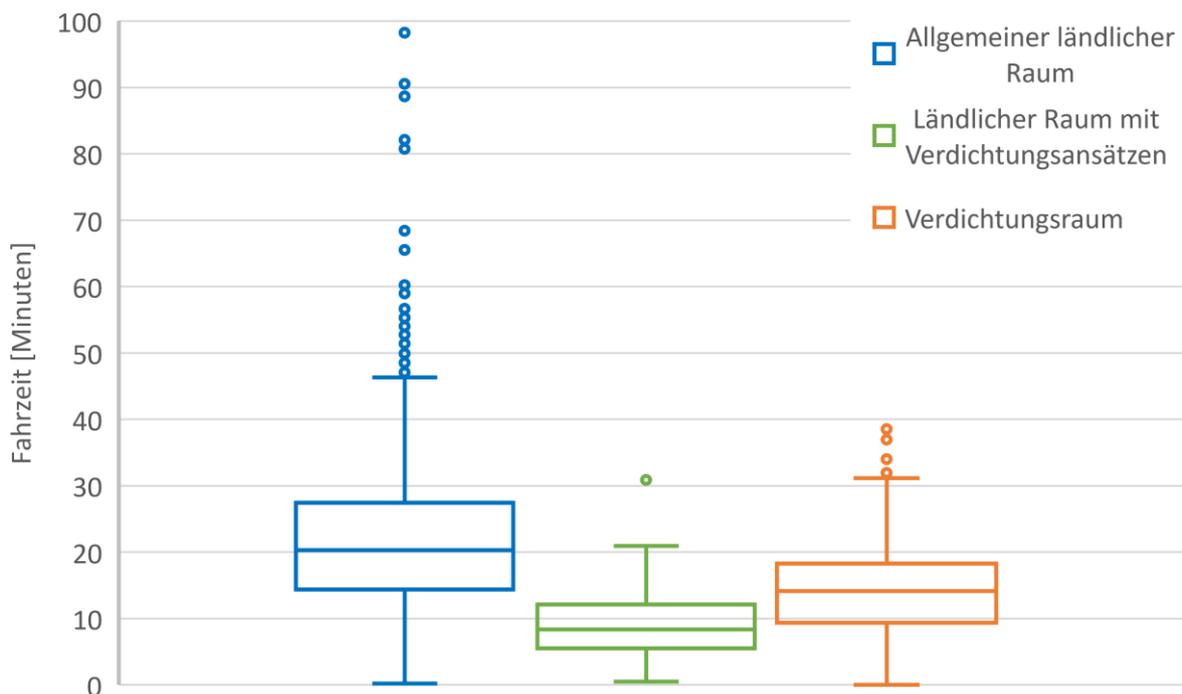


Abb. 31: Verteilung der Fahrzeiten zum nächstgelegenen Krankenhaus bei Schlaganfall

Im Median beträgt die Fahrzeit im ländlichen Raum 20,3 Minuten und liegt damit weit über entsprechendem Wert im ländlichen Raum mit Verdichtungsansätzen (8,4 Minuten) und im Verdichtungsraum (14,2 Minuten). 75% der Einwohner im ländlichen Raum mit Verdichtungsansätzen erreichen die nächste geeignete Klinik innerhalb von 12,1 Minuten, aber auch im ländlichen Raum liegt das 3. Quartil mit 27,4 Minuten unter dem kritischen Schwellenwert von 30 Minuten.

C) Schweres Schädel-Hirn-Trauma

Mit lediglich 38 Krankenhäusern sind, verglichen mit den anderen Tracer-Diagnosen, die wenigsten Kliniken zur Behandlung eines schweren Schädel-Hirn-Traumas als geeignet eingestuft.

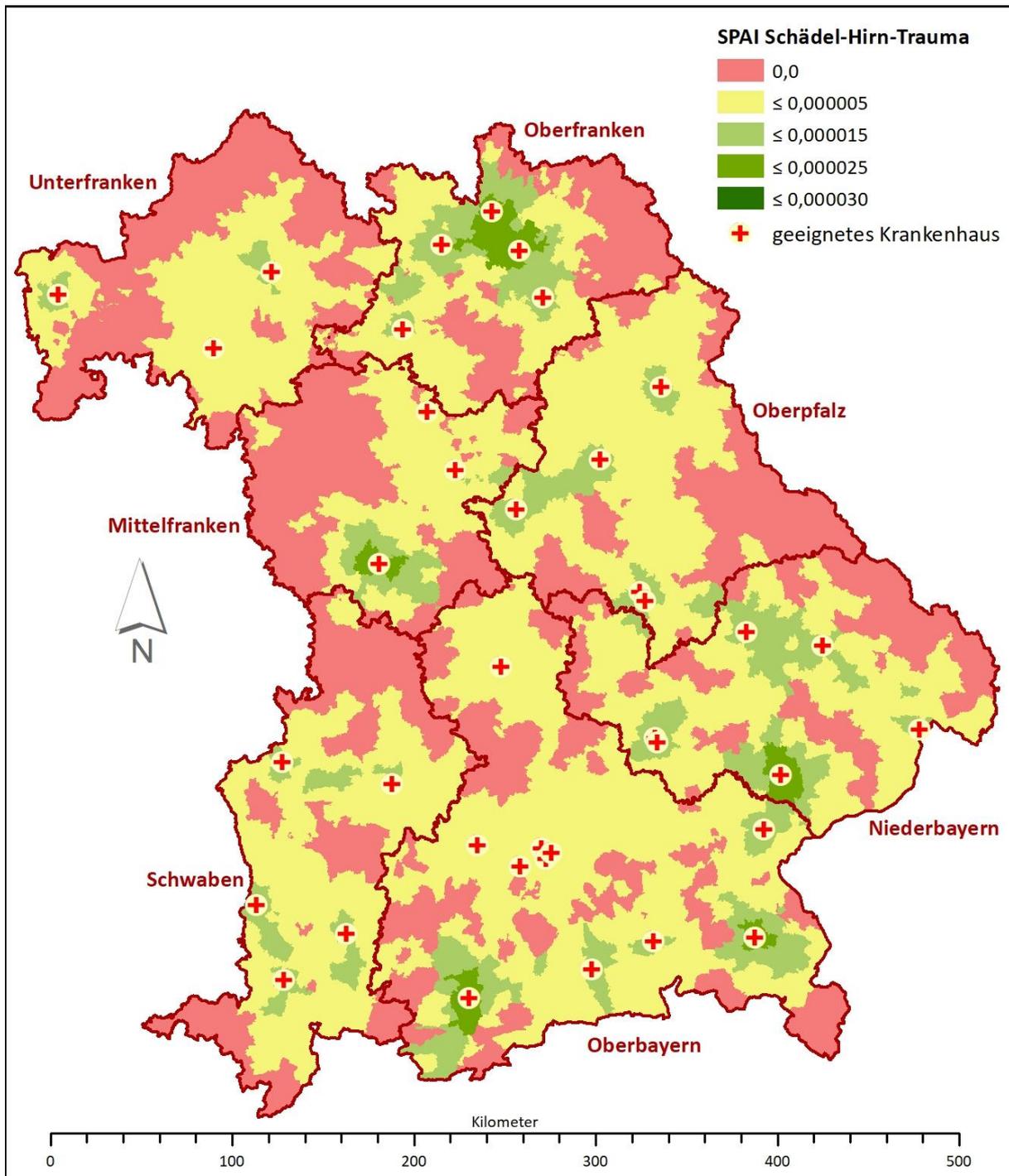


Abb. 32: Spatial Accessibility Index (SPAI) bei schwerem Schädel-Hirn-Trauma

Mit 0,000023 werden die höchsten Index-Werte in den Einzugsbereichen des Klinikums Kulmbach und der HELIOS Frankenwaldklinik in Kronach in Oberfranken sowie im Überschneidungsbereich der Einzugsgebiete der Rottal-Inn Klinik Eggenfelden und der Kreisklinik Altötting in Niederbayern berechnet. In jedem Regierungsbezirk findet sich eine hohe Anzahl an Regionen, die einen SPAI-Wert von 0 aufweisen, besonders in Mittelfranken sowie im Norden Schwabens und Unterfrankens und im Grenzbereich zur Tschechischen Republik.

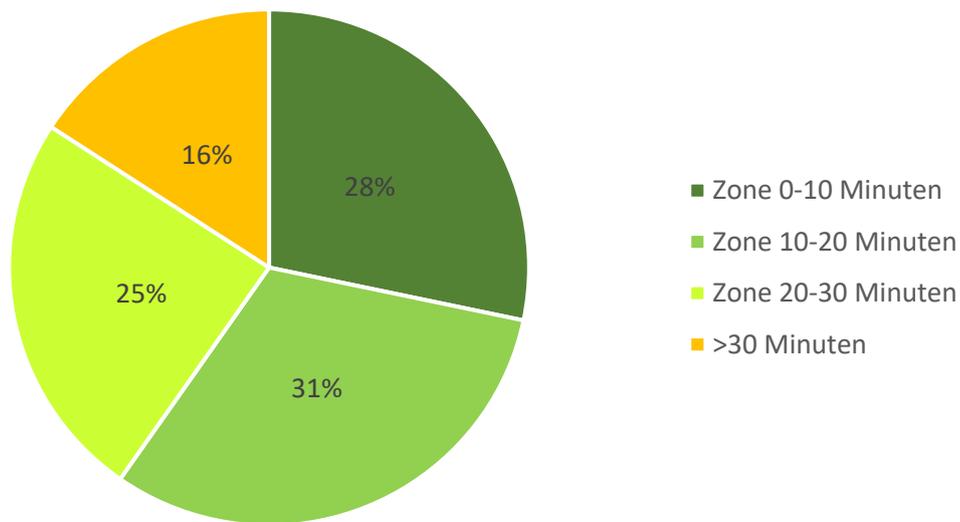


Abb. 33: Verteilung der Bevölkerung nach Erreichbarkeit der nächstgelegenen Klinik bei schwerem Schädel-Hirn-Trauma

Die verringerte Erreichbarkeit zeigt sich auch in der Verteilung der Bevölkerung nach Erreichbarkeitszone. 16% und damit 2.112.673 Einwohner können es nicht in die nächstgelegene geeignete Klinik innerhalb von 30 Minuten schaffen. Bis 10 Minuten Fahrzeit mit einem Rettungswagen benötigen 3.776.271 Einwohner, bis 20 Minuten 4.194.807 Einwohner und bis 30 Minuten 3.266.857 Einwohner in die nächste Klinik.

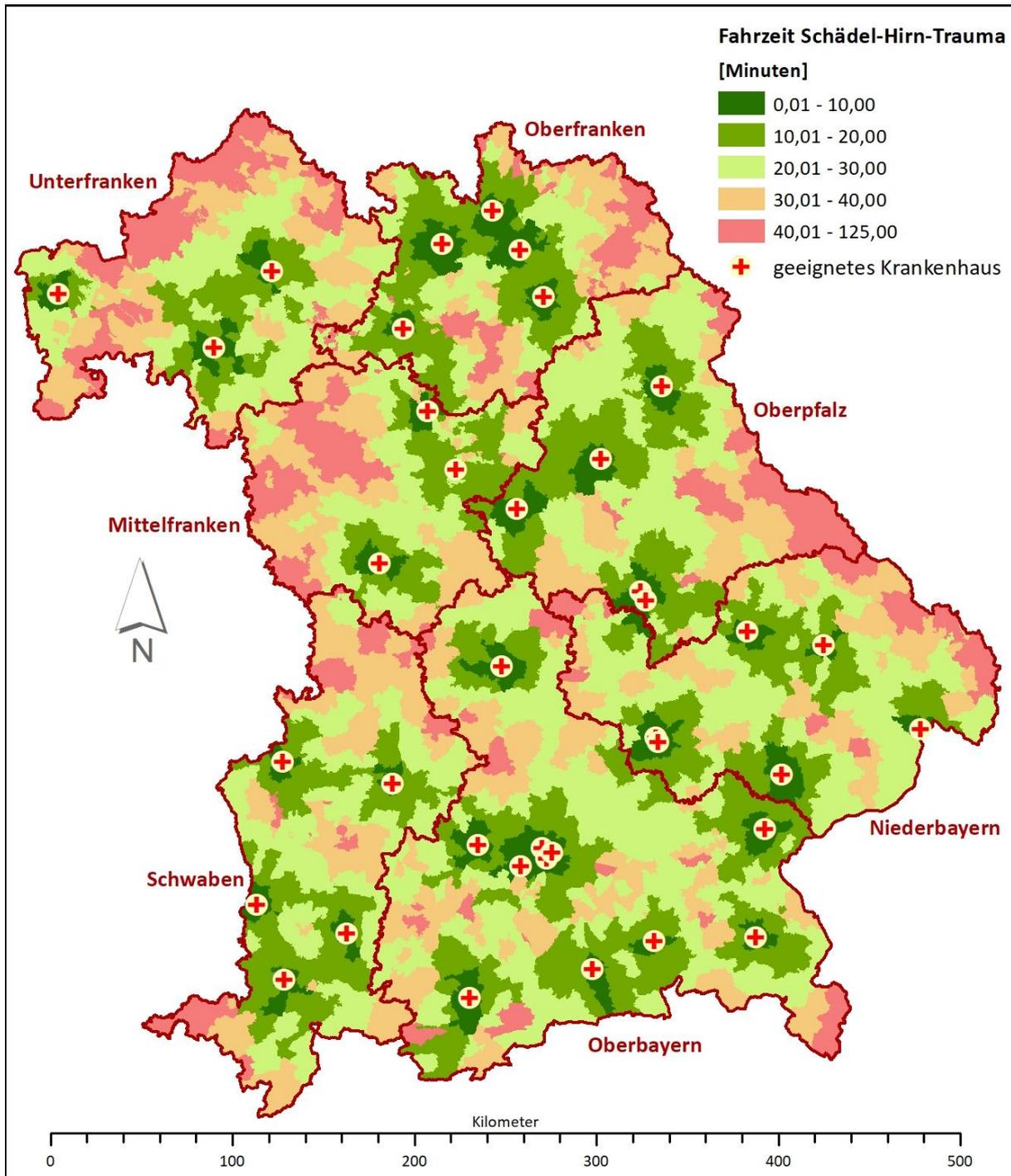


Abb. 34: Fahrzeiten von Gemeinden in die nächste geeignete Klinik bei schwerem Schädel-Hirn-Trauma

Fahrzeitdefizite lassen sich vor allem in Unterfranken im Grenzbereich zu Hessen im Naturpark Bayerische Rhön feststellen. Hier müssen Rettungswägen mit Patienten, die ein mögliches schweres Schädel-Hirn-Trauma erlitten haben, bis zu 120 Minuten in die nächste geeignete Klinik fahren. Auch an der Grenze zur Tschechischen Republik sind lange Fahrzeiten von bis zu 90 Minuten zu konstatieren. Die kürzesten Fahrzeiten finden sich in der Stadt München und seinem Umland mit einer Verfügbarkeit von fünf geeigneten Krankenhäusern sowie in Oberfranken.

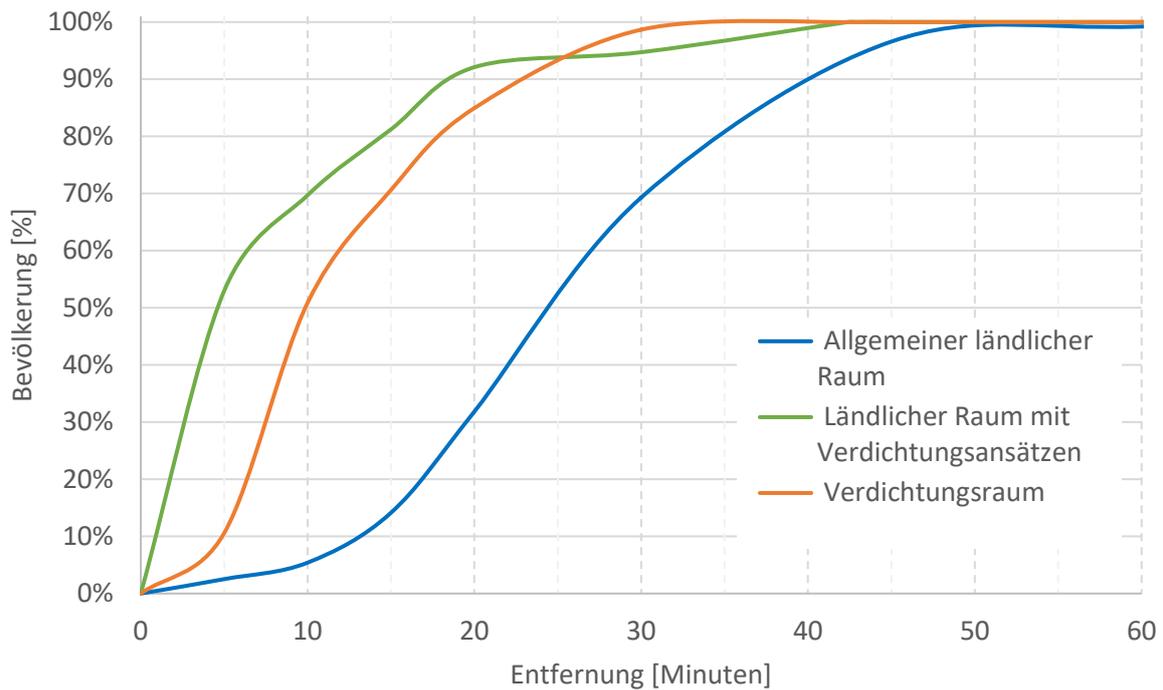


Abb. 35: Bevölkerung nach Entfernung zum nächstgelegenen geeigneten Krankenhaus bei schwerem Schädel-Hirn-Trauma

Die verlängerten Fahrzeiten zeigen sich auch in Abb. 35 für den ländlichen Raum. Erst nach 83 Minuten erreichen 100% der Bevölkerung die nächstgelegene geeignete Klinik. Etwa 70% müssen eine Fahrzeit von 30 Minuten hinnehmen, wohingegen im ländlichen Raum mit Verdichtungsansätzen und im Verdichtungsraum hier bis zu 99% der Einwohner das nächste Krankenhaus erreichen. Bei knapp über 42 Minuten gelangen hier alle Einwohner in die nächste Klinik. Im ländlichen Raum mit Verdichtungsansätzen haben bereits 53% der Bevölkerung innerhalb von 5 Minuten Zugang zu einer geeigneten Klinik.

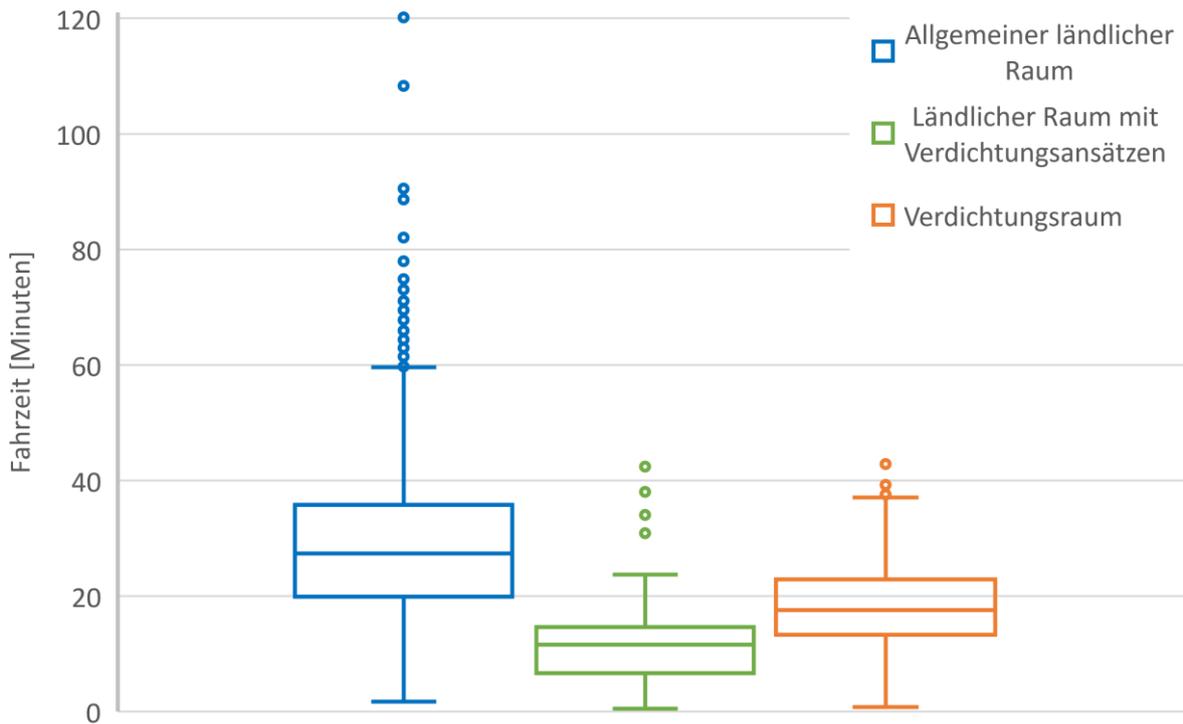


Abb. 36: Verteilung der Fahrzeiten zum nächstgelegenen Krankenhaus bei schwerem Schädel-Hirn-Trauma

Den niedrigsten Median der Fahrzeiten mit 11,6 Minuten weisen Gemeinden des ländlichen Raums mit Verdichtungsansätzen auf, im Verdichtungsraum liegt der Median bei 17,6 Minuten und im ländlichen Raum bei 27,4 Minuten. 75% der Gemeinden im ländlichen Raum weisen eine Fahrzeit von bis zu 35,8 Minuten auf, bei weniger als der Hälfte (14,6 Minuten) liegt das dritte Quartil im ländlichem Raum mit Verdichtungsansätzen.

D) Polytrauma

Für 108 Kliniken kann eine Eignung zur Behandlung von polytraumatisch verletzten Personen festgestellt werden.

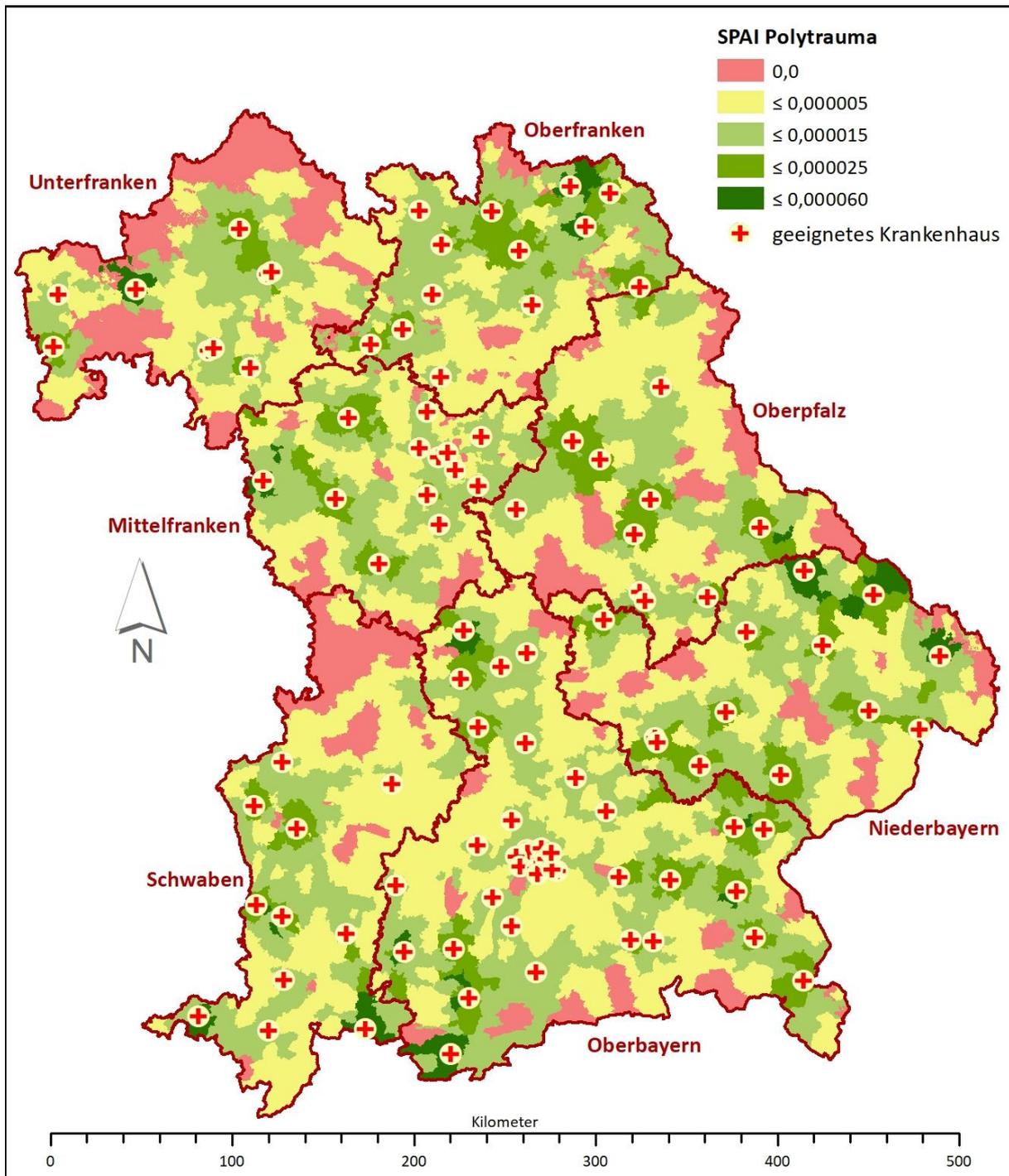


Abb. 37: Spatial Accessibility Index (SPAI) bei Polytrauma

Die größten zusammenhängenden Regionen mit Defiziten im Spatial Accessibility Index zeigen sich in Unterfranken an der Grenze zu Hessen und Thüringen. Knapp 34% der Gemeinden erreichen hier keine geeignete Klinik innerhalb von 30 Minuten, darunter fallen ein Drittel unter gemeindefreies Gebiet. Ein weiteres größeres Gebiet, das Erreichbarkeitsdefizite aufweist, liegt im Norden Schwabens um Nördlingen und Oettingen.

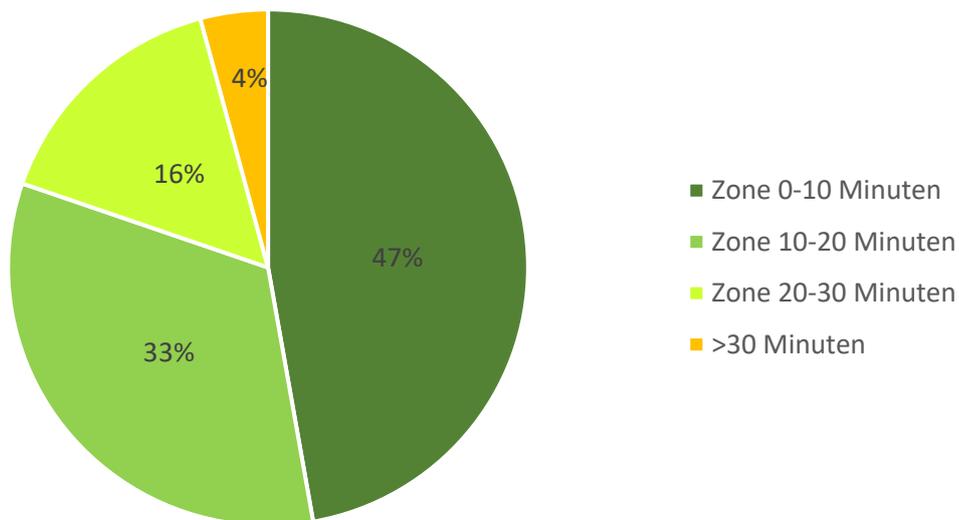


Abb. 38: Verteilung der Bevölkerung nach Erreichbarkeit der nächstgelegenen Klinik bei Polytrauma

In Krankenhäuser, die geeignet sind polytraumatisch verletzte Personen zu behandeln, können 47% der Einwohner innerhalb von 10 Minuten mit dem Rettungswagen eingeliefert werden. Dies umfasst 6.304.580 Personen. Gut ein Drittel (4.413.134 Einwohner) benötigen 10 bis 20 Minuten in die nächste Klinik. Lediglich 560.834 Einwohner befinden sich außerhalb der 30 Minuten Fahrzeitgrenze.

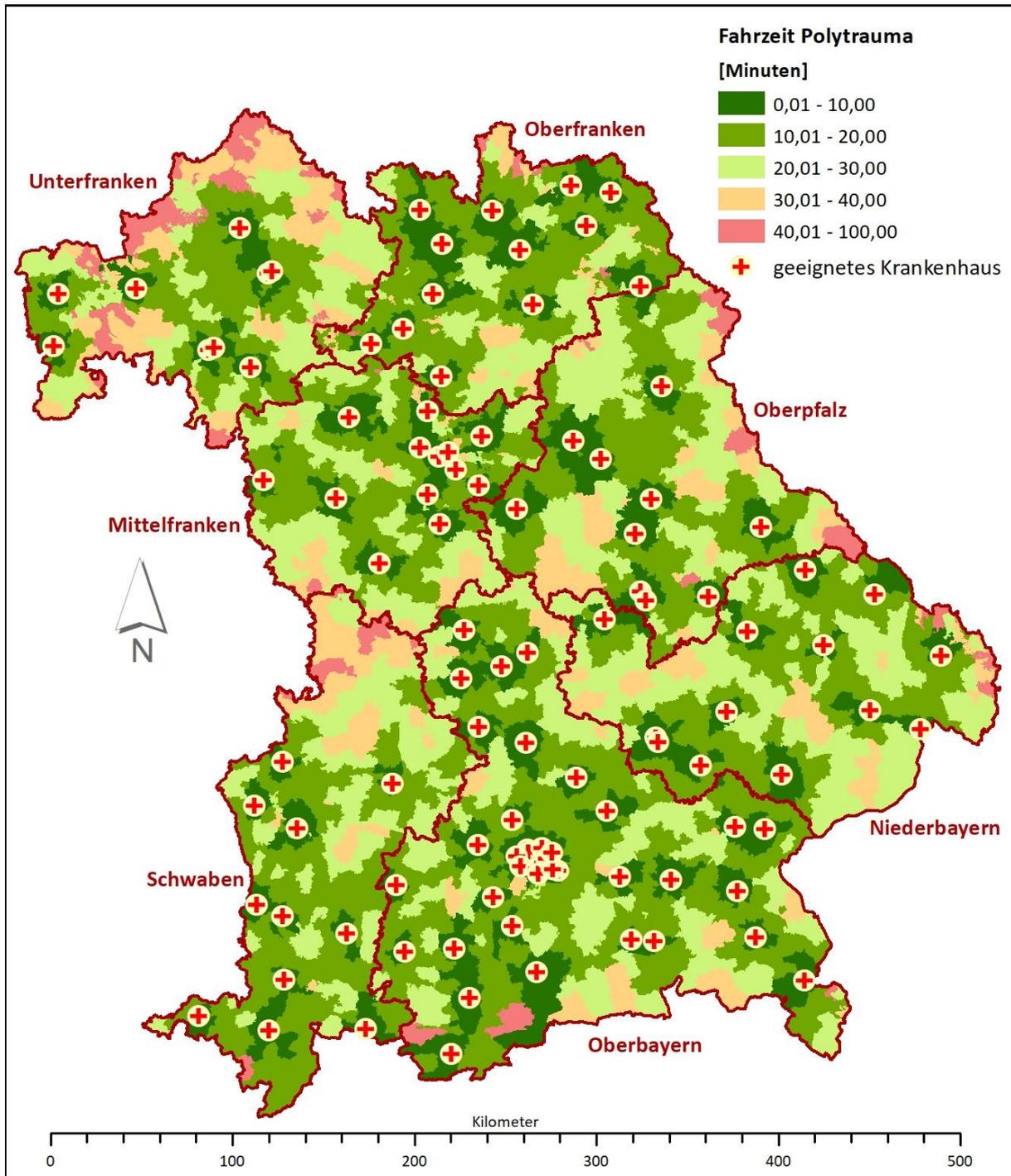


Abb. 39: Fahrzeiten von Gemeinden in die nächste geeignete Klinik bei Polytrauma

Die längsten Fahrzeiten weisen Gemeinden in Unterfranken im Grenzbereich zu Hessen auf. Hier kann die nächste geeignete Klinik in gerade mal 98 Minuten angefahren werden. Da es sich hier aber um gemeindefreies Gebiet handelt, sind keine Einwohner mit Hauptwohnsitz zu verzeichnen. Ebenso im Bayrischen Wald in Niederbayern zeigen sich hohe Fahrzeiten von bis zu 75 Minuten, aber auch hier sind die betroffenen Gemeinden als gemeindefreies Gebiet einzuordnen.

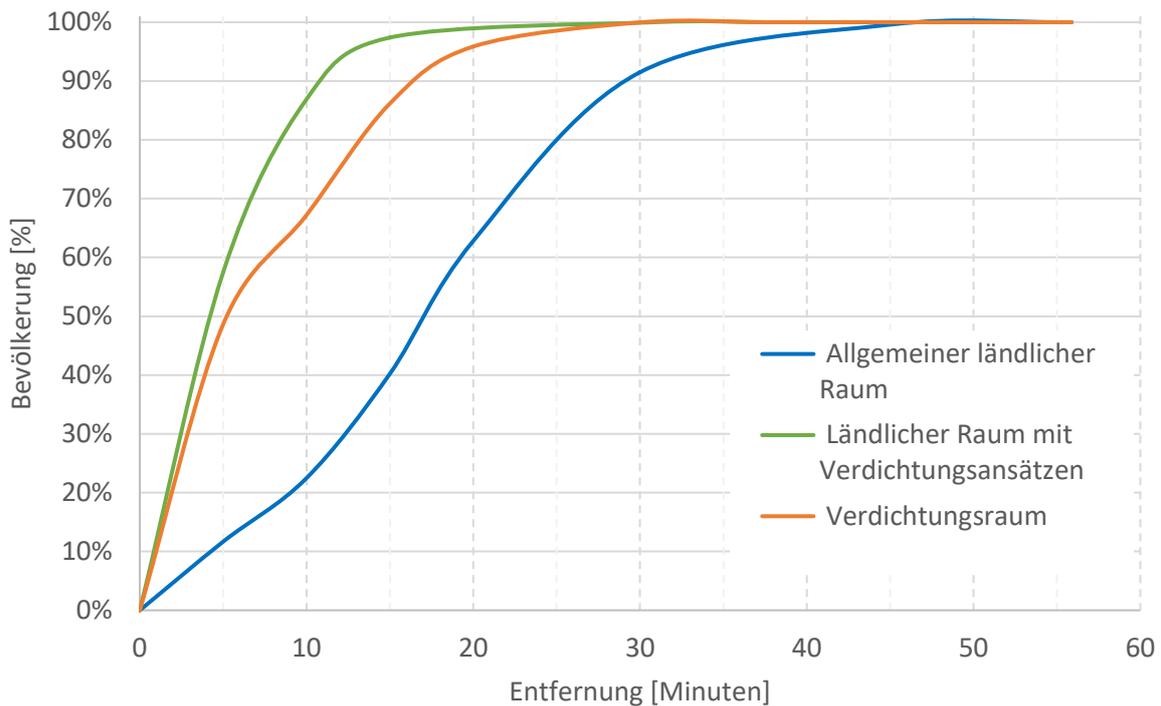


Abb. 40: Bevölkerung nach Entfernung zum nächstgelegenen geeigneten Krankenhaus bei Polytrauma

Die längsten Fahrzeiten in die nächste geeignete Klinik müssen Bewohner des ländlichen Raums auf sich nehmen. So benötigt hier die Hälfte der Bevölkerung etwa 17 Minuten, im Vergleich dazu im ländlichen Raum mit Verdichtungsansätzen knapp über 3 Minuten und im Verdichtungsraum etwa 5 Minuten. Diese Diskrepanz nimmt weiter zu mit seinem höchsten Wert in der Erreichbarkeit der Gesamtbevölkerung. Nach knapp 56 Minuten gelangen sämtliche Einwohner im ländlichen Raum in die nächste geeignete Klinik, hingegen im Verdichtungsraum nach knapp 38 Minuten und im Übergangsbereich bereits nach 31 Minuten.

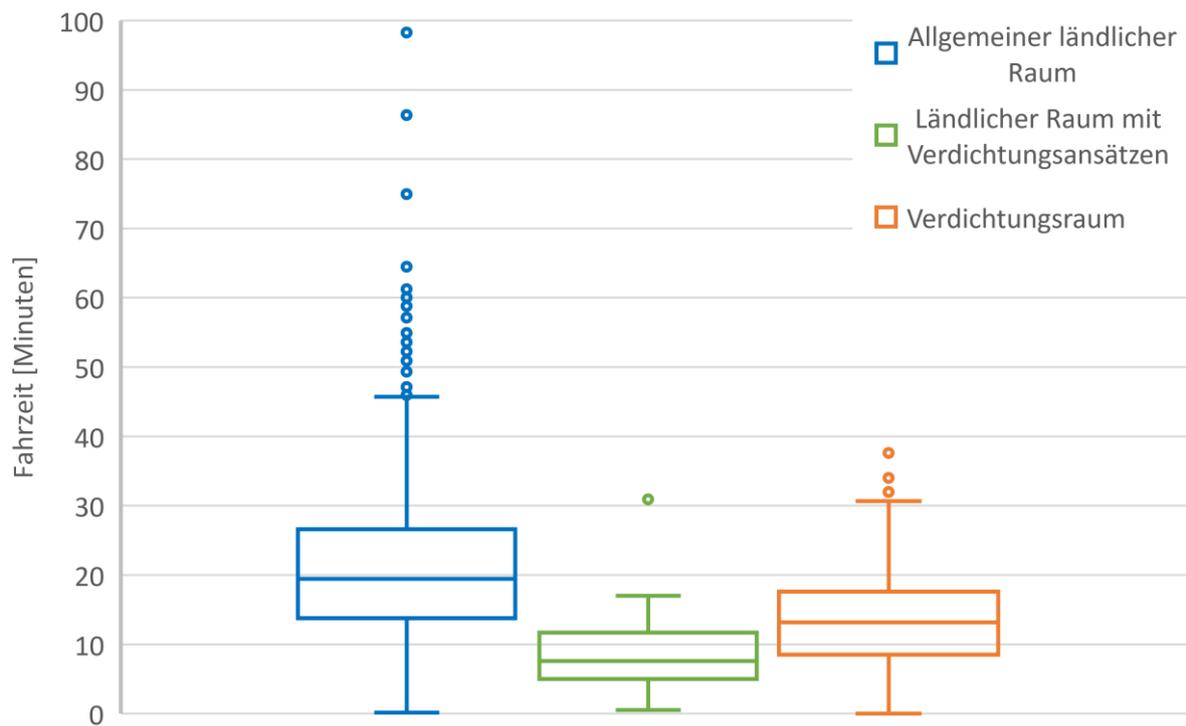


Abb. 41: Verteilung der Fahrzeiten zum nächstgelegenen Krankenhaus bei Polytrauma

Der Median der Fahrzeiten im ländlichen Raum liegt bei 19,4 Minuten und damit mehr als doppelt so hoch, als im ländlichen Raum mit Verdichtungsansätzen (7,6 Minuten). Die Hälfte der Gemeinden erreicht das nächstgelegene Krankenhaus in 13,2 Minuten.

E) Sepsis

Die meisten geeigneten Krankenhäuser zur Behandlung einer Tracer-Diagnose finden sich mit 109 Kliniken bei Sepsis.

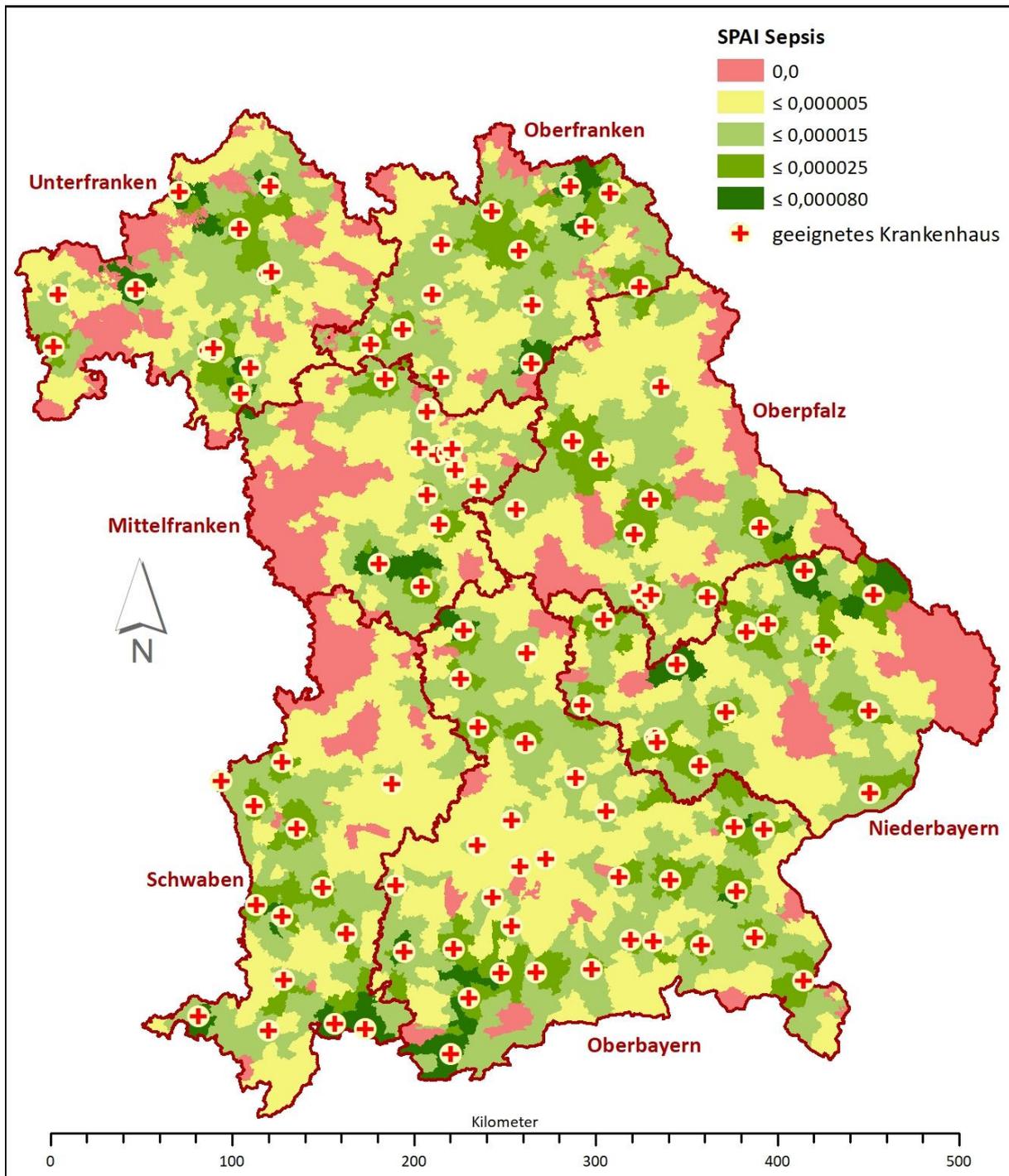


Abb. 42: Spatial Accessibility Index (SPAI) bei Sepsis

Dadurch, dass viele Kliniken zur Behandlung von septischen Patienten geeignet sind und diese eine gute räumliche Verteilung aufweisen, ist auch die Abdeckung des Zugänglichkeitsindex sehr hoch. Besonders in Oberbayern, von 519 ergibt sich lediglich für 19 Gemeinden ein SPAI von 0, und davon liegen 10 in gemeindefreiem Gebiet. Ebenso zeigt die Berechnung eine gute Abdeckung in Oberfranken. Lediglich 3% der bewohnten Gemeinden zeigen einen SPAI von 0. Der höchste Index-Wert von 0,00008 findet sich in Niederbayern in Mallersdorf-Pfaffenhofen, aber auch Gemeinden im Bayrischen Wald, wie Zwiesel und Lindberg, zeigen mit 0,000047 einen hohen Erreichbarkeitsindex. Trotzdem haben größere Regionen im niederbayrischen

Osten sowie zentrale Gemeinden keinen Zugang zu einer geeigneten Klinik innerhalb von 30 Minuten. Ähnliche Verhältnisse zeigen sich in Unterfranken. Geroda (0,000067) und Bad Brückenau (0,000063) weisen sehr hohe Index-Werte auf, dennoch finden sich größere Bereiche mit einem Wert von 0. Knapp 42% davon liegen allerdings in gemeindefreien Gebieten ohne Einwohner mit Hauptwohnsitz. Mittelfranken und das nördliche Schwaben zeigen auch größere zusammenhängende Bereiche mit einem Defizit in der Erreichbarkeit.

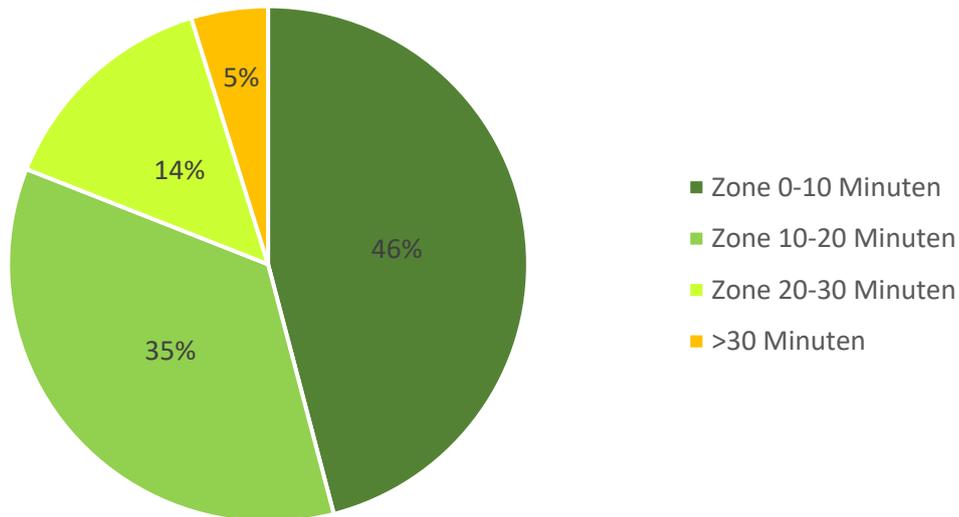


Abb. 43: Verteilung der Bevölkerung nach Erreichbarkeit der nächstgelegenen Klinik bei Sepsis

Die gute Verteilung der Krankenhäuser im Landesgebiet zeigt sich auch in den Fahrzeiten, die die Bewohner benötigen, um die nächstgelegene Klinik zu erreichen. So weisen 46% und damit 6.131.962 Personen eine Anfahrtszeit von maximal 10 Minuten auf. Im Gegensatz dazu können 638.239 Einwohner nicht in einem geeigneten Krankenhaus innerhalb von 30 Minuten sein, insofern sie durch einen Rettungswagen transportiert werden.

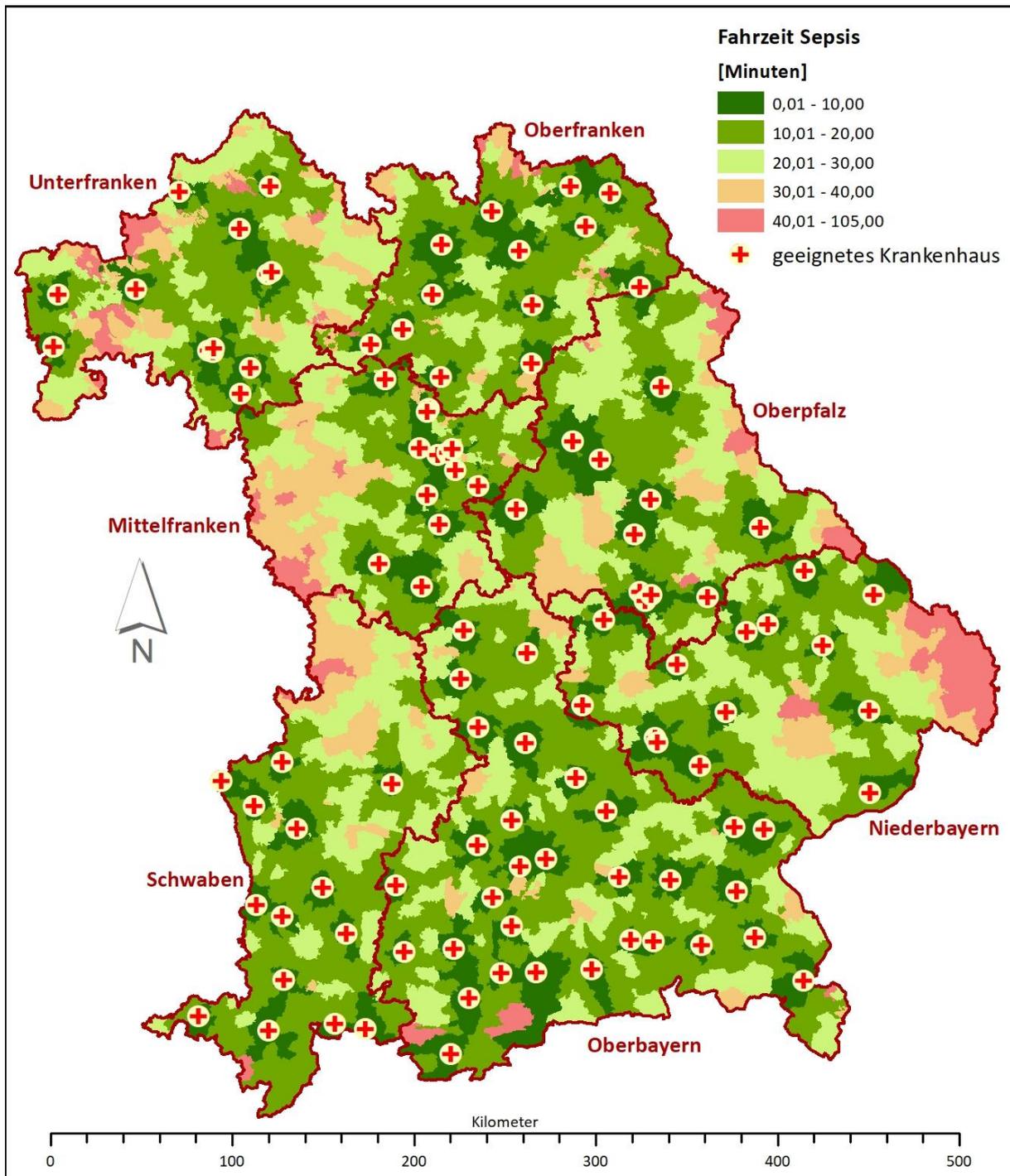


Abb. 44: Fahrzeiten von Gemeinden in die nächste geeignete Klinik bei Sepsis

Obgleich viele Krankenhäuser zur Behandlung von Sepsis geeignet sind und eine gute Verteilung im Flächenland Bayern aufweisen, sind einige wenige Gemeinden zu finden, die länger Fahrzeiten von über 40 Minuten aufweisen. Diese befinden sich im äußersten Osten Niederbayerns, wobei der überwiegende Teil der Gemeinden, für die sich eine Fahrzeit von sogar über 60 Minuten berechnet, in gemeindefreiem Gebiet liegt.

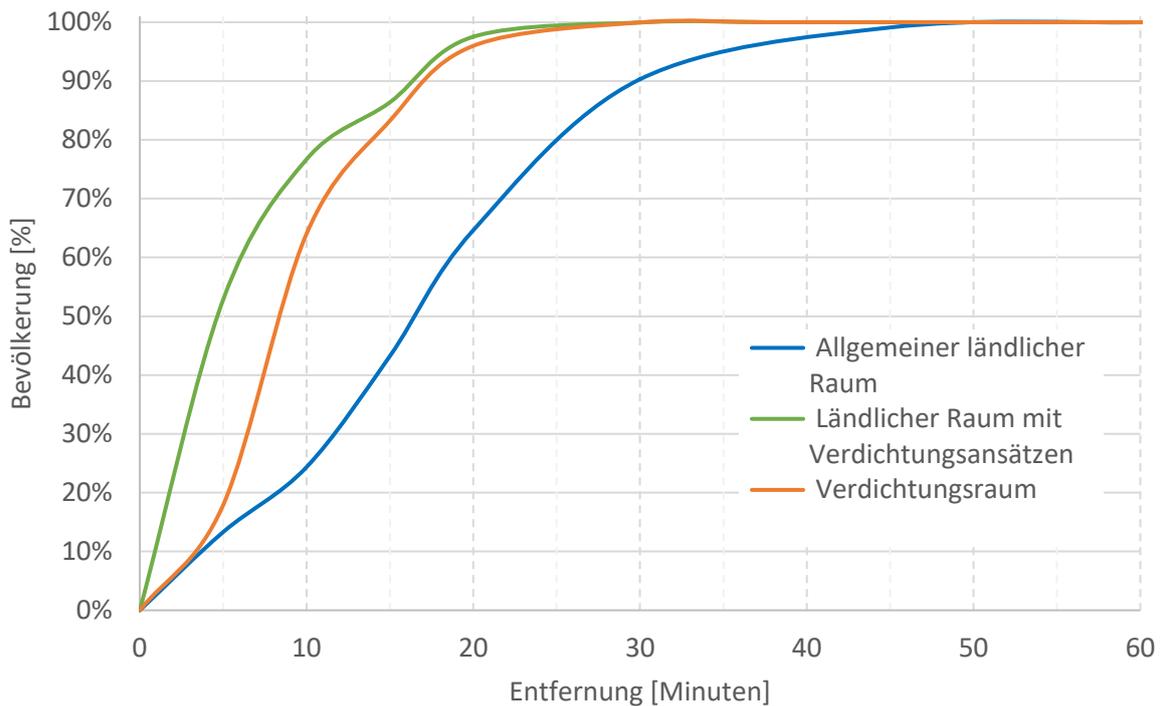


Abb. 45: Bevölkerung nach Entfernung zum nächstgelegenen geeigneten Krankenhaus bei Sepsis

Bei kurzen Fahrzeiten zeigt sich die Divergenz der Erreichbarkeiten, so gelangen mehr als die Hälfte der Bewohner des ländlichen Raums mit Verdichtungsansätzen in die nächste geeignete Klinik, wobei entsprechender Wert im ländlichen Raum (13%) und im Verdichtungsraum (18%) wesentlich niedriger ausfällt. Mit zunehmender Minuten-Entfernung nähern sich die Werte für den ländlichen Raum mit Verdichtungsansätzen und für den Verdichtungsraum an, sodass die 100%-Marke bei 30 Minuten, bzw. bei 38 Minuten erreicht wird. Einwohner des ländlichen Raums müssen bis zu 70 Minuten fahren, damit jeder in der nächsten geeigneten Klinik ankommt.

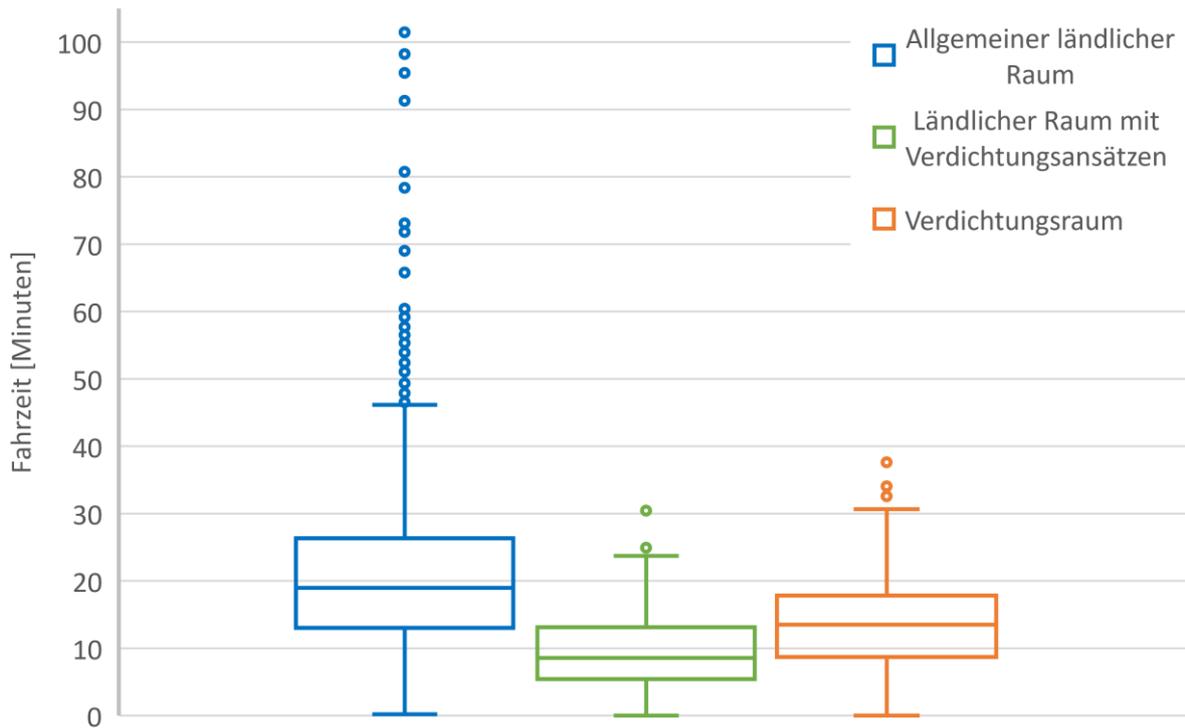


Abb. 46: Verteilung der Fahrzeiten zum nächstgelegenen Krankenhaus bei Sepsis

Eine große Varianz der Fahrzeiten zeigt sich im ländlichen Raum, von bis zu 101 Minuten, wobei der Median bei knapp 19 Minuten liegt. Wesentlich geringer fällt der Median bei den Fahrzeiten im ländlichen Raum mit Verdichtungsansätzen (8,6 Minuten) aus, im Verdichtungsraum beträgt der Median 13,5 Minuten.

F) ST-Hebungsinfarkt

Insgesamt werden 73 Krankenhäuser zur Behandlung von Patienten mit ST-Hebungsinfarkt als geeignet eingestuft.

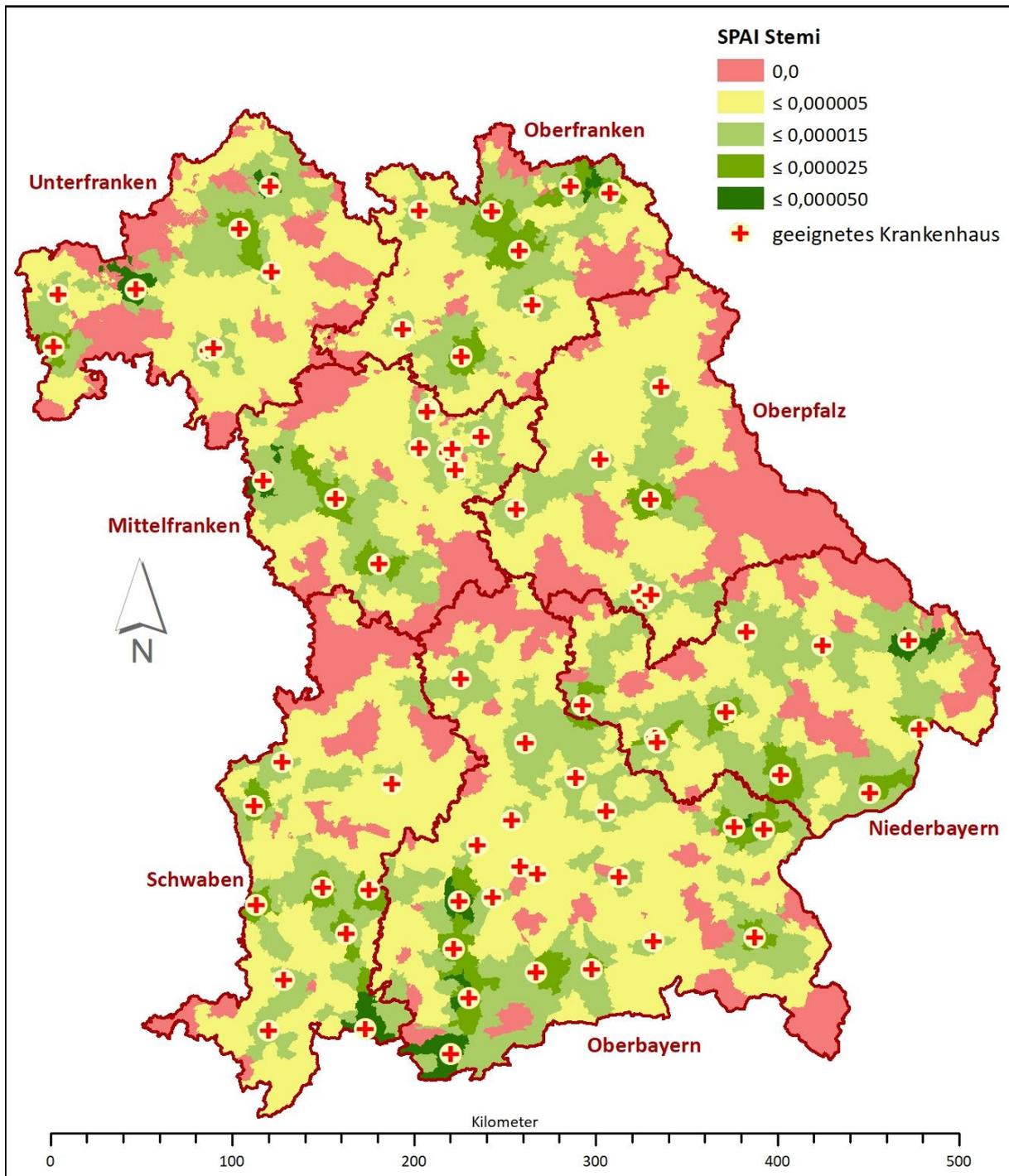


Abb. 47: Spatial Accessibility Index (SPAI) bei ST-Hebungsinfarkt

Eine größere Region mit Defiziten in der Zugänglichkeit zu einer Klinik, welche Herzinfarkt-Patienten behandeln kann, findet sich im Grenzbereich zwischen den Regierungsbezirken Oberbayern, Mittelfranken und Schwaben. Von Riedenburg in Niederbayern bis Nördlingen in Schwaben an der Grenze zu Baden-Württemberg ist ein großes zusammenhängendes Gebiet, mit einem Index-Wert von 0. Auch in der Oberpfalz sind defizitäre Gemeinden zu verzeichnen. Bei etwa 30% der Gemeinden beträgt der SPAI 0.

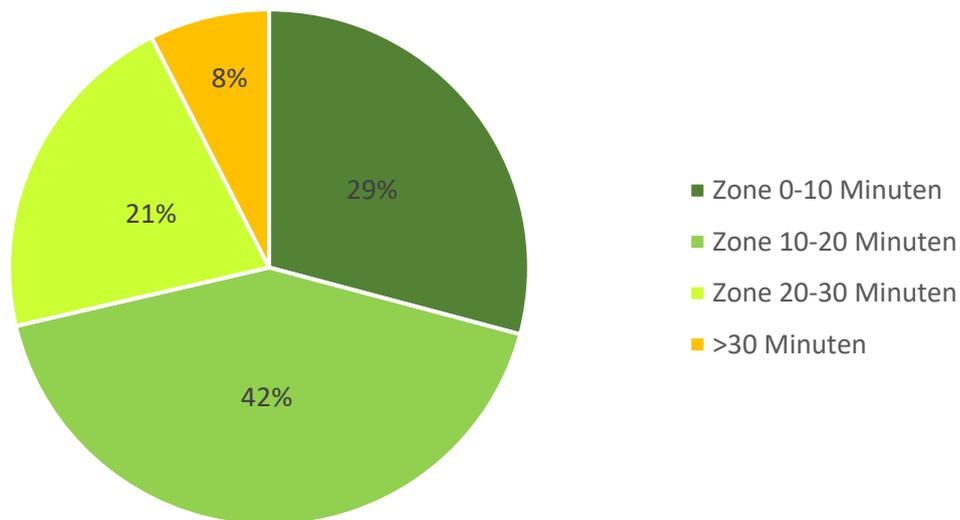


Abb. 48: Verteilung der Bevölkerung nach Erreichbarkeit der nächstgelegenen Klinik bei ST-Hebungsinfarkt

Mit 5.632.583, benötigt der größte Teil der Bevölkerung (42%) in das nächste geeignete Krankenhaus 10 bis 20 Minuten. 3.894.096 Einwohner schaffen es innerhalb von 10 Minuten, sofern diese in einem Rettungswagen eingeliefert werden. Eine Fahrdauer von über 30 Minuten müssen 997.073 Personen auf sich nehmen.

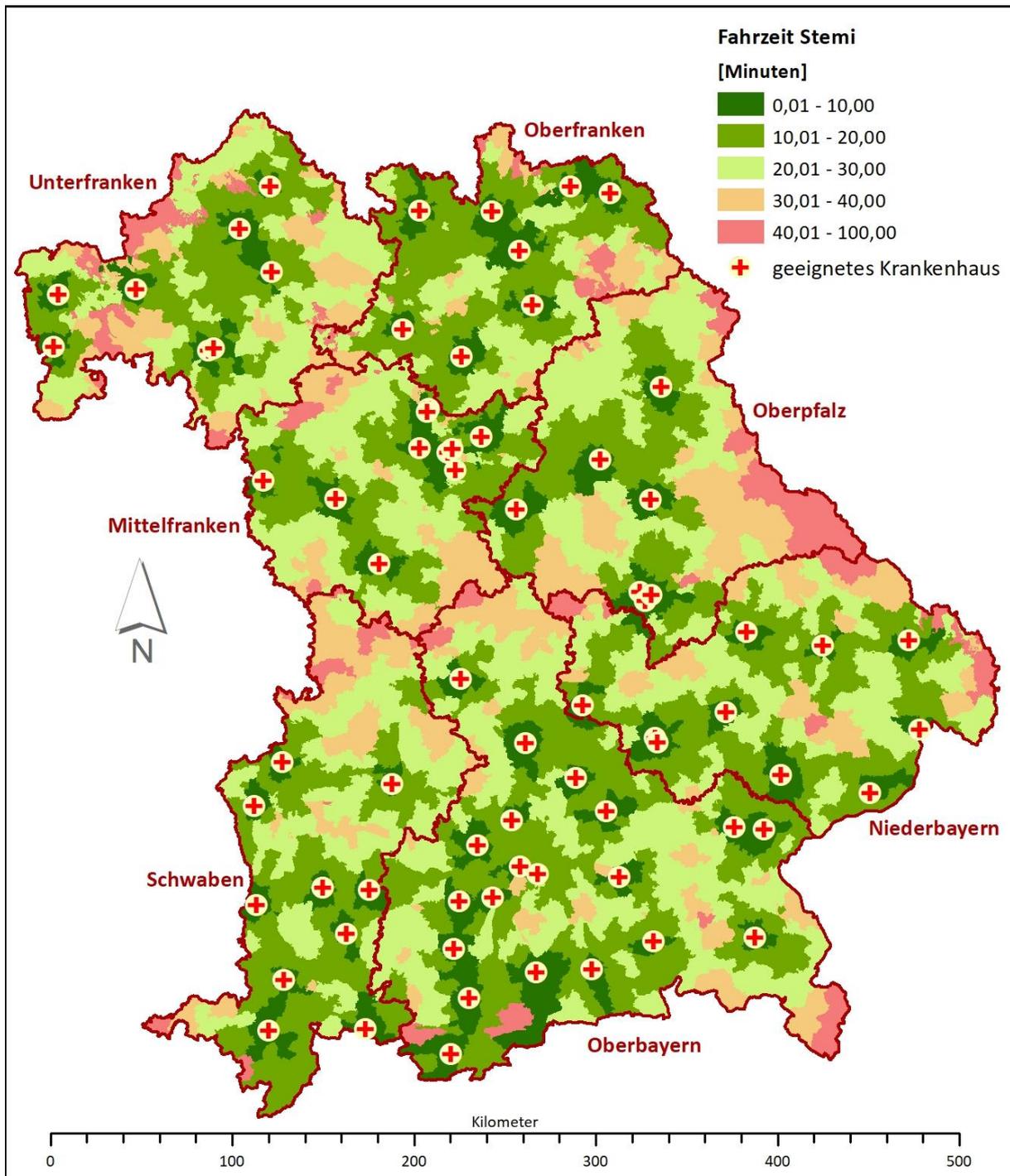


Abb. 49: Fahrzeiten von Gemeinden in die nächste geeignete Klinik bei ST-Hebungsinfarkt

Ein großer Bereich mit Defiziten in der Erreichbarkeit zeigt sich in der Oberpfalz, besonders im Grenzbereich zur Tschechischen Republik. Knapp 43% der Gemeinden zeigen eine Fahrzeit von über 30 Minuten, in gut einem Drittel davon sogar über 40 Minuten, in die nächstgelegene geeignete Klinik. Auch für die Region um Berchtesgaden berechnen sich hohe Fahrzeiten, von zum Teil über 40 Minuten bis hin zu knapp 80 Minuten im gemeindefreien Gebiet.

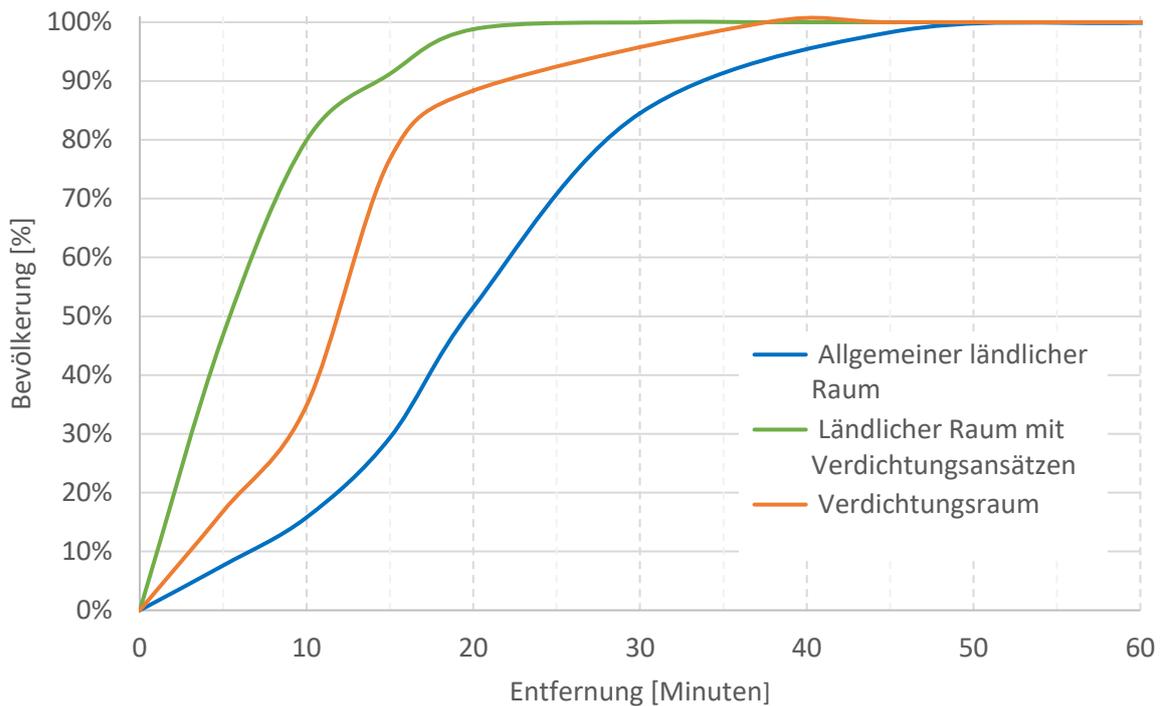


Abb. 50: Bevölkerung nach Entfernung zum nächstgelegenen geeigneten Krankenhaus bei ST-Hebungsinfarkt

Im Vergleich zu den anderen Raumtypen, erreichen die Bewohner des ländlichen Raums mit Verdichtungsansätzen in kürzer Fahrzeit die nächste geeignete Klinik. Hier schaffen es bis zu 99% der Einwohner bereits in 20 Minuten, wohingegen im Verdichtungsraum 88% und im ländlichen Raum gerade einmal 52% dieselbe Fahrzeit benötigen. Mehr als 31 Minuten fahren Rettungswägen im Übergangsbereich nicht, um alle Bewohner in die nächste Klinik zu transportieren. Im Verdichtungsraum liegt das Maximum bei knapp 38 Minuten und im ländlichen Raum bei 79 Minuten.

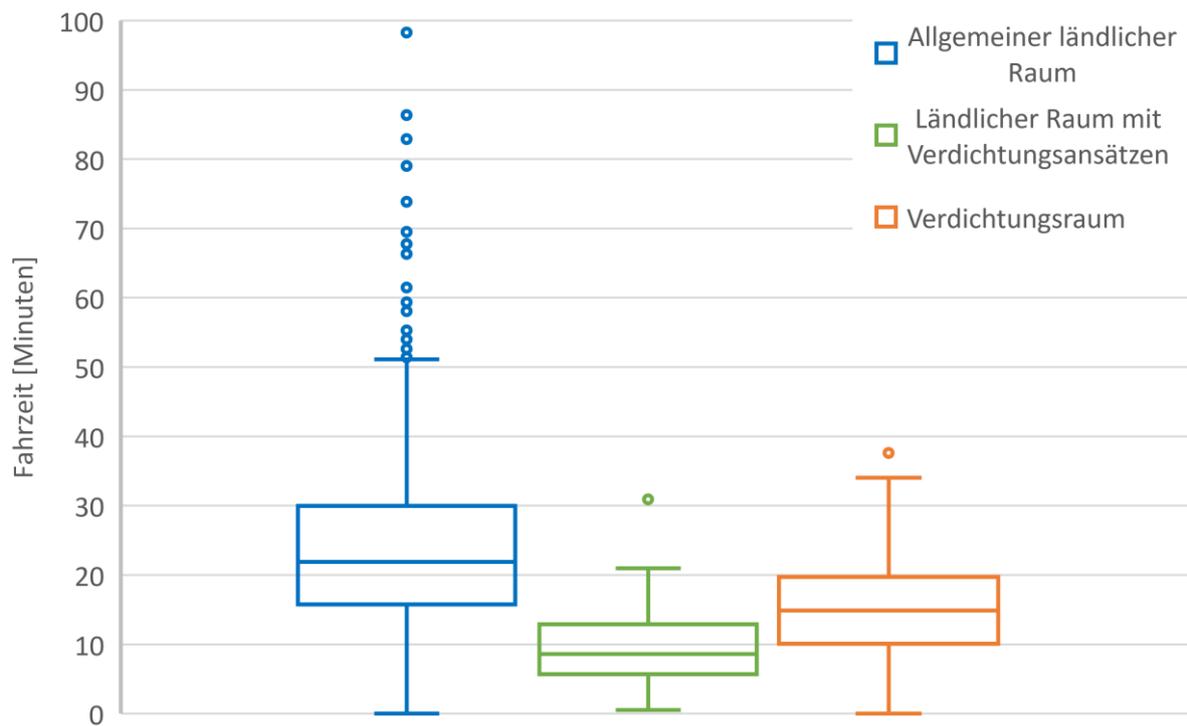


Abb. 51: Verteilung der Fahrzeiten zum nächstgelegenen Krankenhaus bei ST-Hebungsinfarkt

Im Median fährt der Rettungswagen im ländlichen Raum 22 Minuten in die nächste geeignete Klinik, wohingegen im ländlichen Raum mit Verdichtungsansätzen lediglich 9 Minuten benötigt werden. Im Verdichtungsraum beträgt der Median der Fahrzeit knapp 15 Minuten.

4.2.2 Vergleich der Ergebnisse

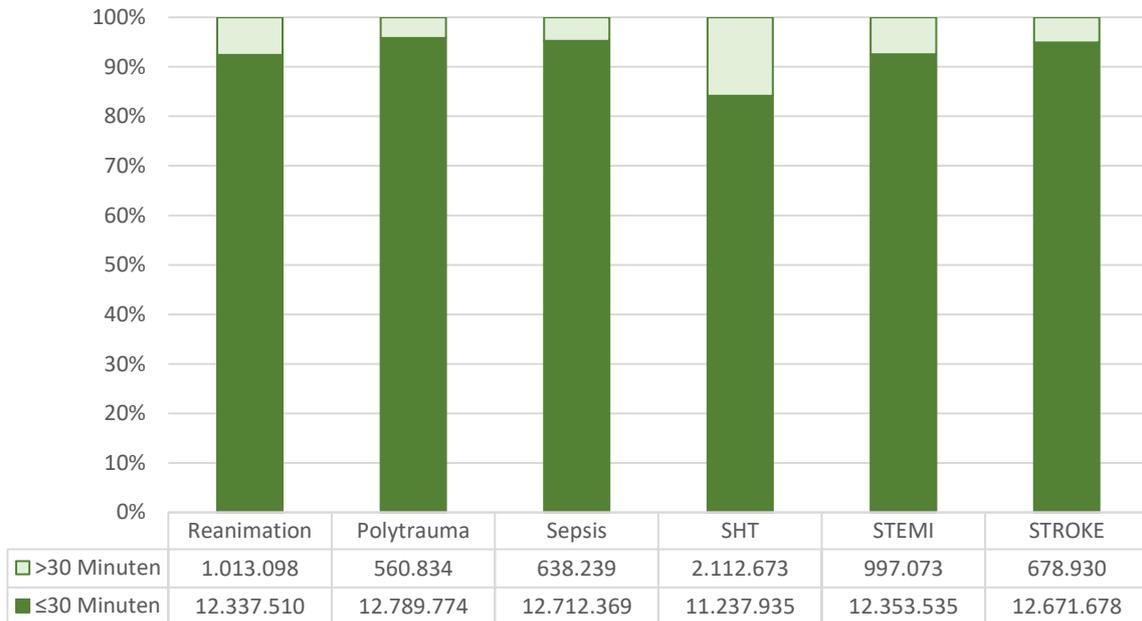


Abb. 52: Anzahl der Einwohner (Gesamtzahl 13.350.608) mit Erreichbarkeit von Kliniken innerhalb und über 30 Minuten nach Tracer-Eignung

Bei schwerem Schädel-Hirn-Trauma schafft es der niedrigste Anteil der Bevölkerung (84%) innerhalb von 30 Minuten in die nächste geeignete Klinik. Über zwei Millionen Einwohner würden demnach mit dem Rettungswagen mehr als 30 Minuten fahren. Die beste Erreichbarkeit innerhalb von 30 Minuten kann für Patienten mit Verdacht auf Polytrauma, Sepsis und Schlaganfall konstatiert werden. Bei plötzlichen Kreislaufstillstand und Herzinfarkt benötigen jeweils knapp 8% über 30 Minuten Fahrzeit.

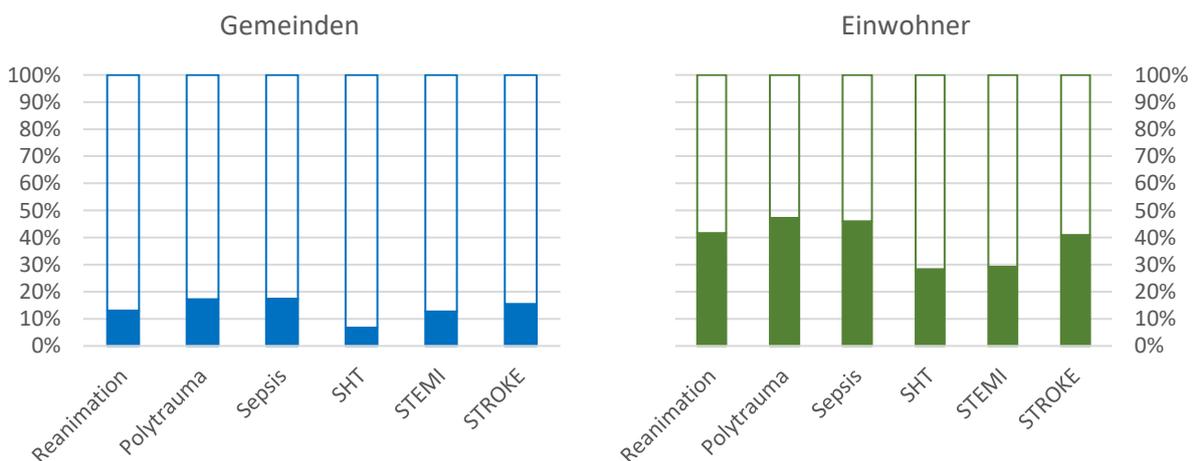


Abb. 53: Erreichbarkeit von Kliniken in Zone 0 - 10 Minuten nach Tracer-Eignung

Eine Fahrzeit von maximal 10 Minuten weisen knapp unter 30% der Einwohner bei schwerem Schädel-Hirn-Trauma (3.776.271) und Herzinfarkt (3.894.096) auf. Mit jeweils über 6.000.000 Einwohner erreichen über 45%, der höchste berechnete Anteil, das nächste geeignete Krankenhaus.

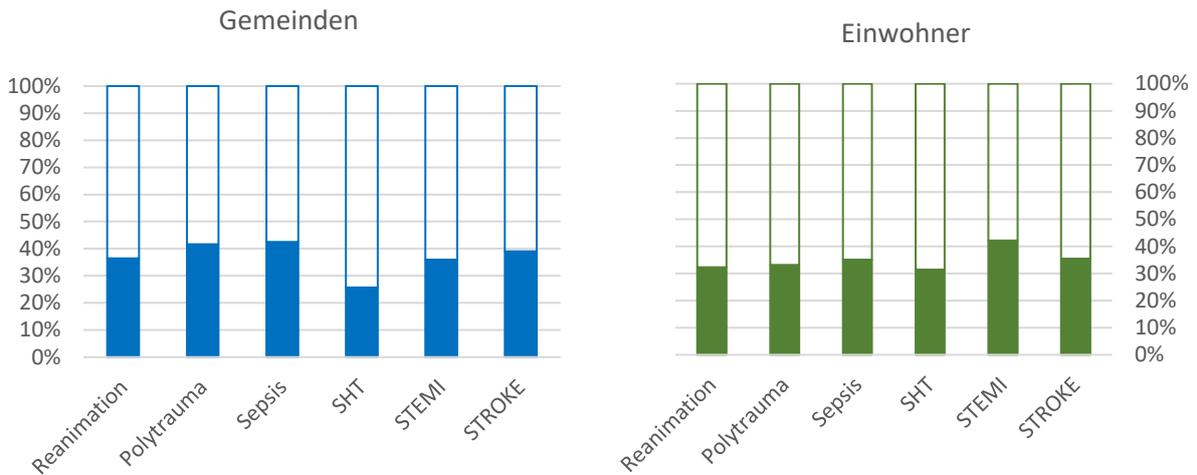


Abb. 54: Erreichbarkeit von Kliniken in Zone 10 - 20 Minuten nach Tracer-Eignung

10 bis 20 Minuten benötigen 31% bis 35% der Bevölkerung bei sämtlichen Tracer-Diagnosen. Lediglich im Falle eines Herzinfarkts beträgt der Anteil 42% und umfasst damit 5.632.583 Personen.

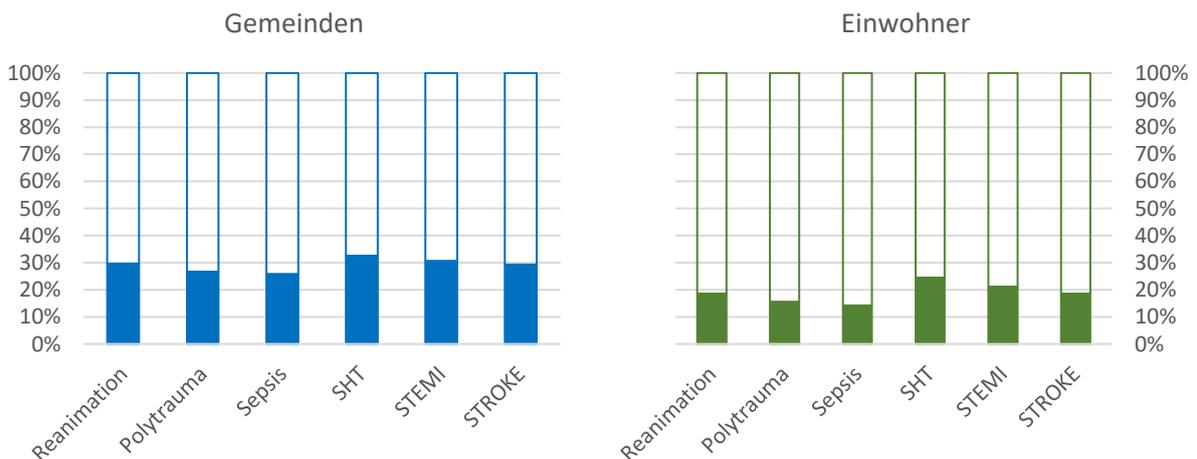


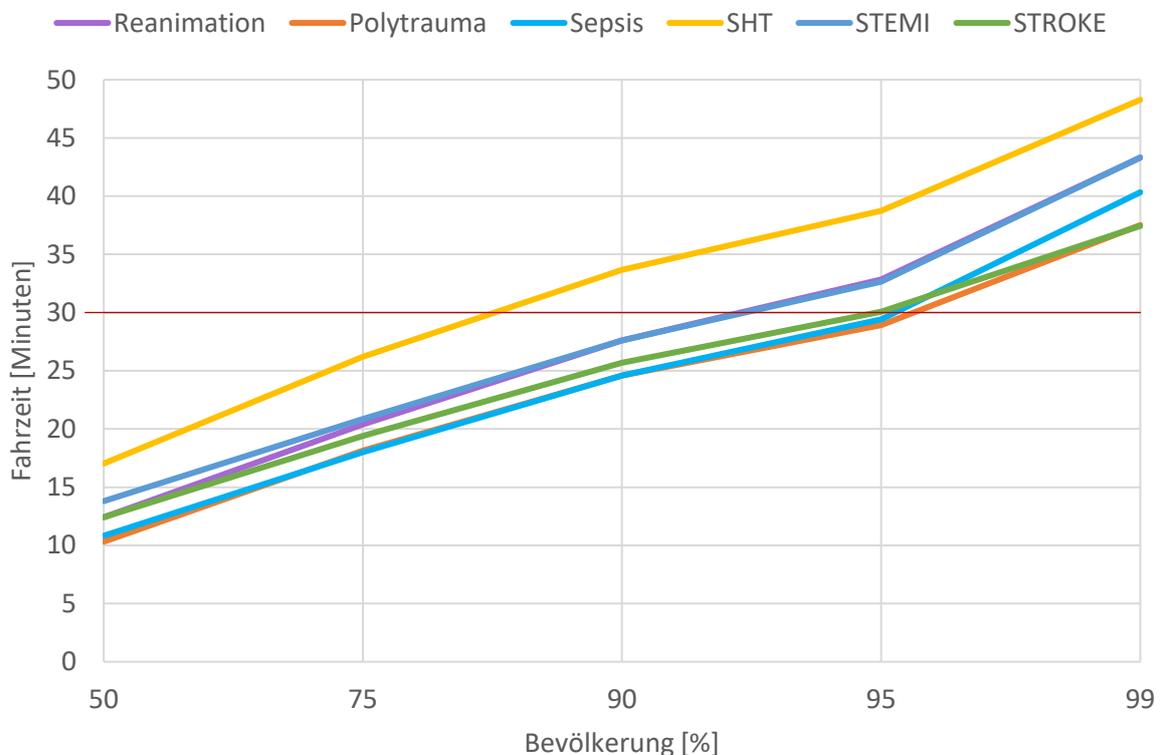
Abb. 55: Erreichbarkeit von Kliniken in Zone 20 - 30 Minuten nach Tracer-Eignung

Bei einem schweren Schädel-Hirn-Trauma und bei Herzinfarkt wird eine Fahrzeit von 20 bis 30 Minuten für über 20% der Bevölkerung berechnet. Der Anteil für alle übrigen Tracer-Diagnosen liegt zwischen 14,2% bei Sepsis und 18,6% bei Reanimation und Schlaganfall.

Tab. 30: Kumulierter Bevölkerungsanteil nach Fahrzeitzonen zur nächstgelegenen Klinik

Fahrzeitzonen	kumulierter Bevölkerungsanteil					
	Reanimation [%]	Polytrauma [%]	Sepsis [%]	SHT [%]	STEMI [%]	STROKE [%]
0-10	41,59	47,22	45,93	28,29	29,17	40,89
10-20	73,81	80,28	81,02	59,71	71,36	76,34
20-30	92,41	95,80	95,22	84,18	92,53	94,91
>30	100,00	100,00	100,00	100,00	100,00	100,00

Nach maximal 20 Minuten erreichen über 80% der polytraumatisierten oder septischen Patienten die nächstgelegene für die jeweilige Tracer-Diagnose geeignete Klinik. Im Gegensatz dazu liegt der Anteil bei einem schweren Schädel-Hirn-Trauma bei knapp unter 60%.

**Abb. 56: Fahrzeit in die nächstgelegene Klinik nach Bevölkerungsperzentilen**

Um in die nächstgelegene geeignete Klinik zu fahren, muss die Hälfte der Bevölkerung bei Verdacht auf Polytrauma und Sepsis jeweils knapp über 10 Minuten mit dem Rettungswagen transportiert werden. Davon weicht der Wert für schweres Schädel-Hirn-Trauma erheblich ab, hier beträgt die Fahrzeit für 50% der Bevölkerung über 17 Minuten. Für diese Tracer-Diagnose sind im Allgemeinen wesentlich höhere Transportzeiten zu konstatieren. So müssen 95% der Einwohner fast 39 Minuten ins nächste Krankenhaus zurücklegen, bei plötzlichem Kreislaufstillstand und Herzinfarkt knapp 33 Minuten. Die kürzesten Fahrzeiten für die bayrische Bevölkerung berechnen sich für Polytrauma.

Tab. 31: Erreichbarkeit von Kliniken in 30 Minuten Fahrzeit

Anzahl erreichbarer Kliniken	Bevölkerungsanteil der mind. die genannte Anzahl an Kliniken erreicht					
	Reanimation [%]	Polytrauma [%]	Sepsis [%]	SHT [%]	STEMI [%]	STROKE [%]
1	92,41%	95,80%	95,22%	84,18%	92,53%	94,91%
2	73,58%	84,00%	84,20%	49,57%	71,84%	78,27%
3	52,94%	64,26%	65,96%	23,86%	47,51%	58,34%
4	37,53%	47,87%	52,29%	16,04%	32,25%	41,87%
5	32,15%	37,63%	39,62%	14,19%	23,97%	30,74%
6	28,44%	31,26%	29,08%	0,00%	9,39%	24,68%
7	27,06%	27,81%	11,04%	0,00%	7,43%	17,76%
8	25,99%	26,39%	9,01%	0,00%	1,25%	15,45%
9	19,73%	24,44%	8,01%	0,00%	0,87%	12,96%
10	17,78%	19,18%	7,24%	0,00%	0,00%	1,31%

Knapp 50% der Einwohner mit Verdacht auf ein schweres Schädel-Hirn-Trauma liegt in einem 30-Minuten Einzugsgebiet von maximal zwei geeigneten Krankenhäusern, hingegen bei Polytraum und Sepsis 84% der Einwohner. Bis zu fünf Kliniken können für einen Teil der Bevölkerung bei allen Tracer-Diagnosen erreicht werden. Fast 18% der Patienten mit einem plötzlichen Kreislaufstillstand unter Reanimation, sind im Umkreis von 30 Minuten in zehn geeigneten Krankenhäusern.

5 Diskussion

5.1 Beantwortung des Forschungsziels

Durch die Auswertung der Ergebnisse lassen sich Versorgungs- und Erreichbarkeitsdefizite von medizinischen Einrichtungen in Bayern beobachten. Erwartungsgemäß berechnet sich eine bessere Erreichbarkeit von Krankenhäusern in Verdichtungsräumen und in Bereichen mit Verdichtungsansätzen im Vergleich zu ländlichen Gebieten. Die Heterogenität zwischen Stadt (Verdichtungsraum und ländlicher Raum mit Verdichtungsansätzen) und dem allgemein ländlichen Raum zeigt sich vor allem am geringeren Versorgungsangebot für die Größe der Fläche und der Anzahl der Einwohner.

Tab. 32: Gegenüberstellung von Kennwerten für Stadt und Land

	Anzahl		
	Gemeinden	Einwohner	Kliniken
Stadt	381	6.943.119	82
Land	1.848	6.407.489	66

Dies erklärt auch die hohe Varianz der Fahrzeiten im ländlichen Raum, der Gemeinden mit Krankenhausstandorten umfasst sowie gemeindefreie Gebiete, ohne Einwohner mit Hauptwohnsitz, aber den längsten Fahrzeiten in ein Krankenhaus. Dennoch finden sich hier die höchsten Werte des SPAI. Kurze Fahrzeiten im Einzugsbereich um einen Klinikstandort und eine geringere Bevölkerungszahl, die potenziell das Klinikangebot nutzen können im Vergleich zu Stadtgebieten, erzeugen einen höheren Zugänglichkeitsindex.

Insgesamt zeigen sich 227 Gemeinden (davon 89 im gemeindefreien Gebiet), die bei keiner Tracer-Diagnose ein Krankenhaus innerhalb von 30 Minuten erreichen. Fünf dieser Gemeinden befinden sich im Verdichtungsraum, wobei bis auf Geiselbach in Unterfranken im Grenzbereich zu Hessen allesamt gemeindefreies Gebiet darstellen, und eine Gemeinde im ländlichen Raum mit Verdichtungsansätzen. Die restlichen 221 Gemeinden können dem ländlichen Raum zugeordnet werden.

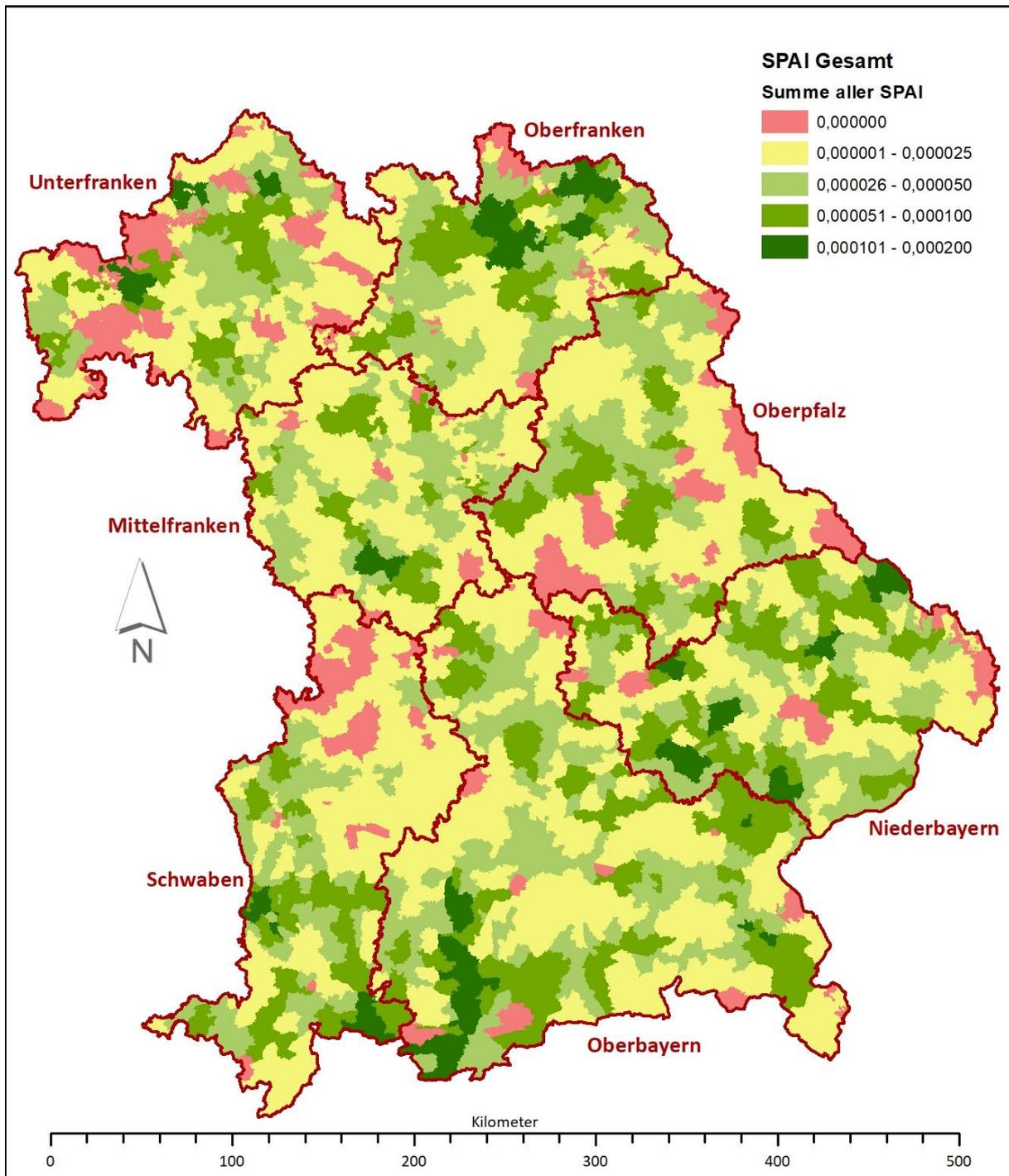


Abb. 57: Summe des Spatial Accessibility Index über alle Tracer-Diagnosen

Insgesamt, für alle Tracer-Diagnosen zeigen Gemeinden des ländlichen Raums mit Verdichtungsansätzen die kürzeren Fahrzeiten in die nächste geeignete Klinik. Gemeinden diesen Raumtyps sind größtenteils kreisfreien Städten mit dem näheren Umland zuzuordnen und weisen eine geringere räumliche Ausdehnung auf als die Gemeinden im Verdichtungsraum. Daher könnte sich die schnellere Erreichbarkeit von Kliniken erklären. Dies könnte allerdings auch auf eine Überschätzung der Fahrgeschwindigkeiten für Bundes- und Landstraßen aus dem Routingprofil hindeuten. Dennoch erscheinen die Ergebnisse plausibel.

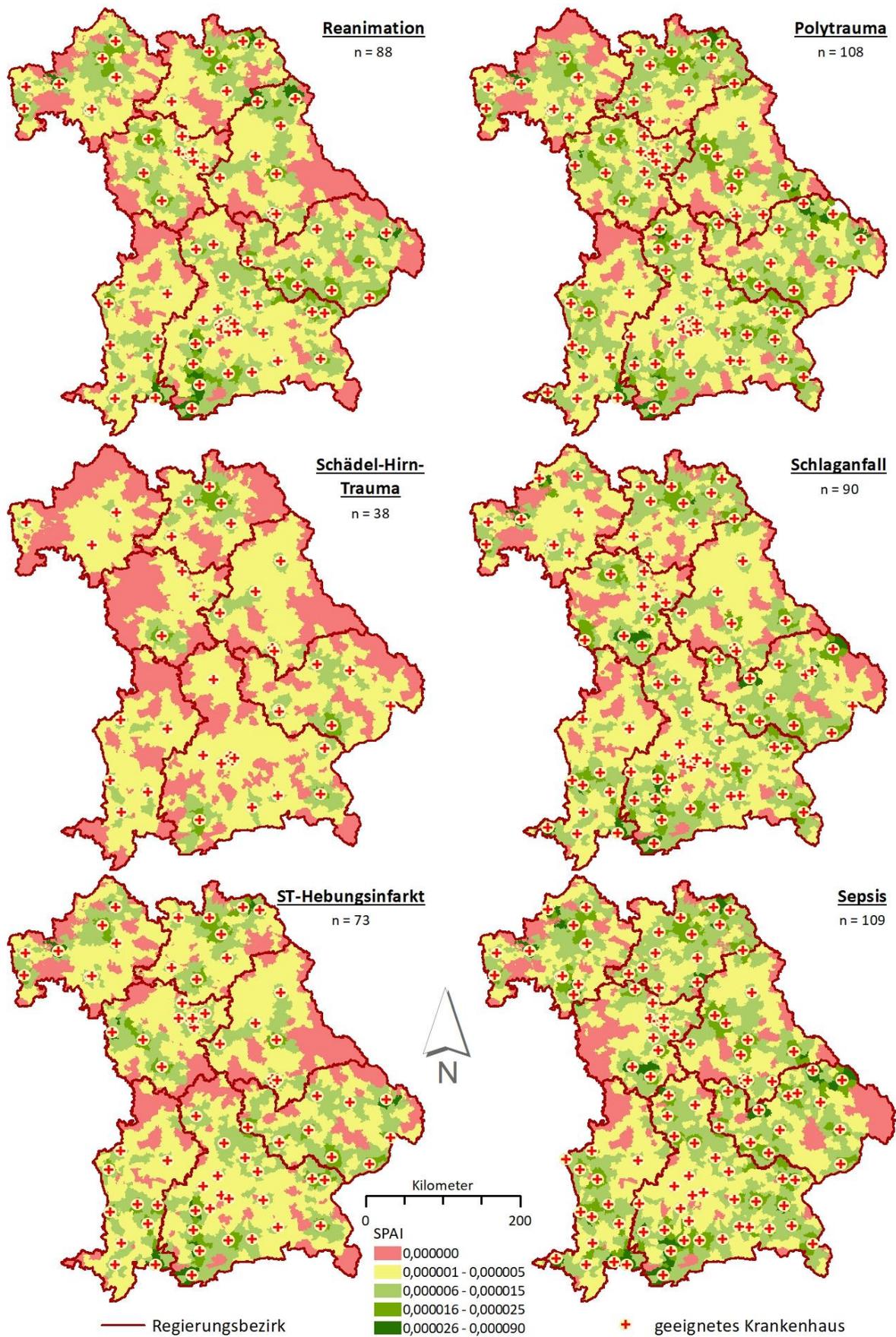


Abb. 58: Spatial Accessibility Index je Tracer-Diagnose

Größere zusammenhängende Gebiete mit erkennbaren Defiziten in der Verfügbarkeit von geeigneten klinischen Einrichtungen zur Behandlung von Patienten unter Reanimation finden sich an der oberpfälzer Grenze zur Tschechischen Republik und im Bayrischen Wald im östlichen Niederbayern. Aber auch an der westlichen Grenze Bayerns zu Baden-Württemberg in Mittelfranken und im nördlichen Schwaben. In dieser Region könnte ein besserer SPAI erwartet werden, wenn potenzielle Ressourcen im Nachbarbundesland mitberücksichtigt werden würden. Daher besteht die Möglichkeit, dass hier aus der isolierten Betrachtungsweise Bayerns ein eventueller Grenzeffekt resultiert. Auch im Landesinneren finden sich Inseln mit Fahrzeiten von über einer halben Stunde in die nächste geeignete Klinik, vor allem am Berührungspunkt von Niederbayern, Oberpfalz und Oberbayern im Raum Riedenburg sowie ein größeres Gebiet in Oberfranken. Besonders durch die wenigen geeigneten Krankenhäuser in der Oberpfalz und Oberfranken entstehen Erreichbarkeitsdefizite in den genannten Regionen.

Längste Fahrzeiten im Falle eines Schlaganfalls berechnen sich in vornehmlich für gemeindefreie Gebiete, die aber keine Einwohner mit Hauptwohnsitz aufweisen. Das Auftreten von Schlaganfallpatienten in diesen Regionen ist auf Besucher und Touristen zurückzuführen, die saisonal entsprechende Gebiete aufsuchen. Somit besteht eine geringere Wahrscheinlichkeit, dass eine entsprechende Diagnose auftritt, im Vergleich zu dicht besiedelten Gebieten. Gerade einmal 5% der Einwohner müssen über 30 Minuten Fahrzeit in Kauf nehmen, wobei die niedrige Zahl auch auf hohe Fahrzeiten zu gemeindefreien Gebieten ohne Einwohner zurückzuführen ist. Der Grenzbereich zur Tschechischen Republik zeigt erhebliche Erreichbarkeitsdefizite, die strukturell bedingt sind. Da hier keine länderübergreifende Disposition stattfindet kann nicht von einem Grenzeffekt ausgegangen werden. Die Leitstelle müsste mit tschechischen Kliniken telefonischen Kontakt aufnehmen, um die Verfügbarkeit der Notaufnahme abzufragen. Dies kostet unter Umständen viel Zeit. Viele bayrische Leitstellen nutzen IVENA eHealth, ein Krankenhaus-Versorgungsnachweis in Echtzeit, in welchem die Verfügbarkeit benötigter Krankenhausressourcen, wie Notaufnahme, Stroke-Unit oder Herzkatheterlabor, minutengenau abgerufen werden können.

Im Falle eines schweren Schädel-Hirn-Traumas sind nur wenige Krankenhäuser als geeignet eingestuft, daher sind lange Fahrzeiten in vielen Regionen zu erwarten. Die kürzesten Fahrzeiten sind in der Stadt und im Umland von München zu finden, allerdings weist der Zugänglichkeitsindex hier lediglich einen sehr niedrigen Wert (weniger als 0,000005 trotz der Eignung von fünf Kliniken) auf, was auf die sehr hohe Einwohnerzahlen und damit potenzielle Patienten zurückzuführen ist. Mit vielen Einwohnern steigt die Wahrscheinlichkeit für ein Erhöhtes Auftreten einer Tracer-Diagnose und somit steigt die Nachfrage für die angebotenen Krankenhauskapazitäten. Mögliche Grenzeffekte zeigen sich zu Baden-Württemberg, Hessen und Thüringen. Dazu müsste die Eignung der Kliniken in den Nachbarbundesländern ermittelt werden. Einen weiteren Grenzeffekt könnte sich bei Lindau an der Südwest-Spitze Schabens mit Fahrzeiten von mehr als einer Stunde in die nächste geeignete Klinik zeigen.

Die größten zusammenhängenden Defizite in der Zugänglichkeit zu einer Klinik bei Polytrauma sind in Unterfranken zu konstatieren, bedingt durch die geringe Anzahl an geeigneten Krankenhäusern in diesem Regierungsbezirk. In möglichen weiterführenden Analysen müsste die Interaktion von bayrischen Leitstellen mit Kliniken in den umliegenden Bundesländern betrachtet werden, um einen möglichen Grenzeffekt auszuschließen.

Für Polytrauma sowie für Sepsis zeigt sich die höchste Abdeckung des SPAI durch die meisten als geeignet zu betrachtenden Krankenhäuser. Lediglich in Mittelfranken und Schwaben zeigen sich größere zusammenhängende Gebiete mit Defiziten in der Versorgung bei Sepsis. Da die geeigneten Kliniken im Osten Mittelfrankens, bzw. im Süden Schwabens konzentriert zu finden sind, zeigen sich unterversorgte Regionen in diesen Regierungsbezirken, die durch Einbeziehung von Kliniken in Baden-Württemberg verbesserte Index-Werte aufweisen könnten.

Über alle Tracer-Diagnosen hinweg zeigen sich Regionen, die Defizite in der Versorgung aufweisen. Insgesamt weisen in Mittelfranken und Nordschwaben im grenznahen Bereich zu Baden-Württemberg hohe Fahrzeiten in geeignete Krankenhäuser auf, die aber verzerrte Werte darstellen können durch einen potenziellen Grenzeffekt. Auch in Unterfranken werden Kliniken nicht innerhalb von 30 Minuten erreicht. Entsprechende Gemeinden liegen zum einen am Rand des Regierungsbezirks bei Hessen, zum anderen im Naturpark Spessart. Trotz der Nähe zum Klinikum Main-Spessart in Lohr am Main berechnen sich hier lange Fahrzeiten von über 40 Minuten. Dies ist darauf zurückzuführen, dass Gemeinden im Forstgebiet durch wenig Straßen verbunden sind und daher lange Umwege gefahren werden müssen, um entsprechende Einwohner zu erreichen. Strukturelle Defizite lassen sich in der östlichen Oberpfalz und im südlichen Bayerischen Wald feststellen. Vor allem die Sana Klinik in Cham und die Arberlandklinik in Zwiesel sichern hier die Versorgung der Bevölkerung, jedoch können beide Krankenhäuser als nicht geeignet zur Behandlung von Herzinfarkt, schwerem Schädel-Hirn-Trauma und plötzlichem Kreislaufstillstand mit Reanimation betrachtet werden. Des Weiteren fällt ein strukturschwaches Gebiet westlich von Regensburg, um Hemau und Dietfurt a.d. Altmühl in der südwestlichen Oberpfalz auf sowie im zentralen Niederbayern. Gerade in Niederbayern erweisen sich viele Krankenhäuser als geeignet zur Behandlung von Tracer-Diagnosen, dennoch gibt es Gemeinden, mit Fahrzeiten von mehr als einer halben Stunde. Auffallend sind auch zwei Gemeinden im Süden Oberbayerns, für welche sich Fahrzeiten von über 40 Minuten berechnen bei sämtlichen Tracer-Diagnosen. Die Jachenau und der Ettaler Forst liegen vermeintlich nahe an einer hohen Dichte an geeigneten Kliniken, dennoch sind sie bezogen auf die Erreichbarkeit sehr abgelegen. Nur wenige und eher Nebenstraßen führen in die Täler der entsprechenden Gebiete in den bayrischen Voralpen.

5.2 Bewertung der Ergebnisse

Mit den Ergebnissen zur Zugänglichkeit zu Kliniken nach Tracer-Eignung können Erkenntnisse über strukturelle Probleme in der bayrischen Krankenhauslandschaft gewonnen werden und Erreichbarkeitsdefizite aufgedeckt werden. Dies erlaubt es Empfehlungen abzuleiten, insbesondere für die Dispositionsstrategie der Integrierten Leitstellen. Treten Einsätze mit Verdacht auf eine Tracer-Diagnose in Regionen auf, für welche ein Erreichbarkeitsdefizit konstatiert wurde, könnte von Seiten der Integrierten Leitstelle primär ein luftgestütztes Rettungsmittel, ein sogenannter Rettungshubschrauber alarmiert werden, um deren Zeitvorteil gegenüber bodengebundenen Rettungsmitteln zu nutzen.

Bisher ist es nicht möglich Krankenhäuser dazu zu verpflichten entsprechende Kapazitäten und Ressourcen zur Behandlung von Tracer-Diagnosen bereitzustellen, um die erkannten strukturellen Defizite zu beheben. Krankenhäuser sind zum größten Teil privat organisiert und müssen in der eigenen Planung wirtschaftliche Aspekte berücksichtigen.

5.3 Eignung der gewählten Methoden

Die vorliegende Arbeit zeigt, dass die Zielsetzung mit Hilfe der angewandten Methoden erfüllt werden kann. Dennoch sind einige Limitationen in der Umsetzung zu beachten. Relevant für die Betrachtung sind zwar die Transportfahrten von den Gemeinden ins nächste geeignete Krankenhaus, die Basis für die Berechnung des Geschwindigkeitsprofils stellen aber Einsatzfahrten von Rettungswägen zum Einsatzort dar. Da die Fahrten vom Einsatzort zum Krankenhaus eine zu unsichere Dokumentation aufweist und lediglich eine eingeschränkte Datenmenge umfasst, wurden diese für die Abschätzung des Geschwindigkeitsprofils nicht herangezogen. Es ist aber davon auszugehen, dass sich ein geringfügig anderes Fahrprofil bei Fahrten mit Patienten im Rettungswagen ergeben. Das in dieser Arbeit geschätzte Profil stellt grundsätzlich eine Annäherung an Fahrten bei einem Notfalleinsatz, mit Blaulicht und Martinshorn sowie Sonderrechten bei Vorfahrt und Ampeln, dar.

Für sonstige Straßen, Zufahrtsstraßen und Feldwege ist keine Schätzung der Geschwindigkeiten möglich, da diese Straßenkategorien bei nur wenigen Einsatzfahrten genutzt wurden. Es finden sich jeweils eine bzw. zwei Fahrten im Routing über das OpenStreetMap Straßennetz mit PKW-Profil, daher werden entsprechende Geschwindigkeiten aus der Literatur angenommen. Es bestünde auch die Möglichkeit diese Kategorien mit der Kategorie XXX, die der nicht gefundenen Kanten entspricht, zusammenzufassen. Interessant wäre auch bei der Erstellung des Geschwindigkeitsprofil die Erweiterung der Straßenkategorien um die Steigung der Straßensegmente und der Kurvigkeit. Dabei ist das Problem zu beachten, dass die Anzahl der Straßenkategorien abgewogen werden muss. Eine ausreichende Anzahl an Einsatzfahrten ist erforderlich, die entsprechende Abschnitte befahren, um ein verlässliches Geschwindigkeitsprofil für die einzelnen Kategorien über die lineare Regression zu erhalten.

In der Nutzung der OSRM Routing-Engine zeigte sich eine Einschränkung in der Berechnung einer $n \times m$ -Matrix. Matrixberechnung ist zwar möglich, allerdings erweist sich diese in OSRM bei der Anzahl der zugrunde gelegten Relationen als zu begrenzt, als dass es sich zur Berechnung der Distanzmatrix zwischen Gemeinden und Krankenhäuser eignet. Als Lösung wurden die Einzelrelationen geroutet, aus einer Datenbasis aus dem kartesischen Produkt der entsprechenden Basistabellen.

Die Methode der Enhanced 2-Step Floating Catchment Areas erweist sich als geeignet, um Zugänglichkeitsdefizite zu geeigneten medizinischen Einrichtungen aufzudecken. In einer Weiterführung der Forschungsarbeit würde es sich anbieten die Methode noch weiter zu verfeinern, indem zum Beispiel eine Altersgruppen abhängige Gewichtung je nach Tracer-Diagnose umgesetzt wird. Die Wahrscheinlichkeit z.B. einen Herzinfarkt zu erleiden ist erhöht für Männer und Frauen ab 60 Jahren, somit könnte in diesem Fall speziell die Einwohnergruppe älter als 60 Jahre berücksichtigt werden. Wie in der Beschreibung der Krankenhauslandschaft aufgeführt, sind Kliniken nach Versorgungsstufen gegliedert. Hier würde es sich anbieten die hierarchische Eigenschaft der Einrichtungen in die Priorisierung von Krankenhäusern bei der Untersuchung der räumlichen Erreichbarkeit mit einzubeziehen. Einrichtungen, insbesondere Krankenhäuser unterscheiden sich in der Anzahl der Bettenkapazitäten, Personalschlüssel und Möglichkeiten zur Anschlussbehandlung, je nach Versorgungsstufe. Bei Auftreten eines Tracer-Ereignisses kann nicht ausgeschlossen werden, dass es sich immer um ein isoliertes Ereignis handelt, beispielsweise eine zunächst aufgestellte chirurgische Indikation kann zusätzlich auch eine neurologische Intervention implizieren. In

höher klassifizierten Kliniken sind mehr spezialisierte Fachabteilungen und Experten verfügbar. Die Bewertung der Krankenhäuser als geeignet bezieht sich auf die minimale Ausstattung und die minimale Ausbildung zur Überlebenssicherung des Patienten bei Tracer-Diagnose. Es können daher alle Kliniken anhand der Kriterien als geeignet zur Behandlung von Tracer-Diagnosen angesehen werden und sind daher für die Erstversorgung essenziell. Zur Weiterbehandlung kann es aber zu Patiententransporten von Kliniken niedriger Versorgungsstufe zu Kliniken mit höherer Versorgungsstufe kommen, sogenannte Sekundärtransporte oder Verlegungsfahrten. Daher sollte in der Entscheidung für ein Krankenhaus, bei gleicher Erreichbarkeit, Krankenhäuser mit einer höherwertigeren Klassifizierung bevorzugt werden.

Aus den Ergebnissen kann konstatiert werden, dass die E2SFCA-Methode verwendet werden kann, um die Unterschiede in der Zugänglichkeit von Gesundheitseinrichtungen aufzuzeigen. Unter der Annahme, dass die Unterschiede in der Erreichbarkeit ihre Ursache in den unterschiedlichen Fahrzeiten haben, können Muster über die Angebots-Nachfrage-Relationen aufgezeigt werden.

Eine weitere Limitierung in der Methodik bezieht sich auf die Bevölkerungszahlen. Es werden lediglich Einwohner berücksichtigt, deren Wohnsitz in der entsprechenden Gemeinde liegt. Dabei ist in vielen Regionen ein hoher Berufspendlerstrom zu erwarten, wodurch es eine tageszeitliche Personenverschiebung in Zentren mit hohem Arbeitsplatzangebot gibt. Dazu kommen noch Ein- und Auspendler im Grenzbereich Bayerns. Des Weiteren weisen gemeindefreie Gebiete zwar keine dauerhaften Bewohner mit Hauptwohnsitz auf, wie sie für die Auswertungen herangezogen werden, dennoch können sich saisonal Personen in entsprechenden Gebieten aufhalten, seien es Forstarbeiter oder erholungssuchende Touristen. In der Bevölkerung sind auch Kinder enthalten. Für diese gelten aber andere Anforderungen an die Behandlung von Tracer-Diagnosen. Ein entsprechender Anforderungskatalog eigens für Kinder findet sich im Eckpunktepapier. Die Aussagen zu den Ergebnissen für Erwachsene lassen sich somit nicht ohne Weiteres auf Kinder übertragen. Insgesamt sind weniger und andere Krankenhäuser geeignet zur Behandlung von Kindern, im Vergleich zu Erwachsene Patienten.

Weiterführende Auswertungen und eine Verfeinerung der Methoden könnten zusätzliche Erkenntnisse ermöglichen. Um regionale Versorgungs- und Erreichbarkeitsdefizite aufzudecken, erscheinen die gewählten Methoden und Daten aber einen großen Erfolg zu versprechen.

6 Ausblick

Ab 2023 erfolgt die Einführung eines neuen Leitstellensystems IGNIS-Plus der Firma Sopra Steria für alle 26 bayrischen integrierten Leitstellen zur Koordinierung und Bearbeitung der Feuerwehr- und Rettungsdiensteinsätze. Die Einführung soll in den nächsten 3 Jahren abgeschlossen sein. Vielleicht ist es mit Hilfe der neuen Datenerfassung in den Leitstellen möglich mehr Informationen über den Fahrweg zu erhalten. Andernfalls könnte in einer dieser Arbeit nachfolgenden Studie die Anfrage an den Rettungsdienstausschuss gestellt werden, Befahrungen in unterschiedlichen repräsentativen Regionen Bayerns mit einem Rettungswagen mit Sondersignal durchzuführen und über GPS zu tracken. Zum Zwecke der Standortplanung für Rettungsdienststrukturen wird dies bereits realisiert, allerdings in nicht repräsentativem Umfang.

Für eine mögliche Weiterführung dieser Arbeit wäre es notwendig die Eignung zur Behandlung von Tracer-Diagnosen für Kliniken auch außerhalb von Bayern zu erheben. Die Erkenntnisse aus dieser Arbeit lassen sich durch weiterführende Forschungsvorhaben ergänzen. Bisher hat sich die Arbeit mit den Umständen zur Rettung von Erwachsenen mit möglicher Tracer-Diagnose befasst. Die verwendeten Methoden lassen sich aber auch auf erkrankte Kinder übertragen, bei welchen eine Tracer-Diagnose festgestellt wurde. Da die Behandlung von Kindern andere Ressourcen benötigt, seien es entsprechende Behandlungsgeräte und ein spezielles Training der Ärzte für die pädiatrische Behandlung in kind spezifischen Einrichtungen, zeigt sich eine andere Krankenhauslandschaft in Bayern.

Die Inanspruchnahme des Rettungsdienstes ist in den letzten Jahren angestiegen und die Kapazitäten der Notaufnahmen sind immer öfter überlastet. Dies führt dazu, dass Notaufnahmen für die Annahme von sogar Notfallpatienten zum Teil nicht mehr zur Verfügung stehen. Damit einher geht eine Verlängerung der Transportwege in eine geeignete medizinische Einrichtung. Dies würde die jetzt schon bestehenden Erreichbarkeitsdefizite verschärfen und erhebliche Auswirkungen auf die Versorgung der Bevölkerung nach sich ziehen. Umso wichtiger ist es die bisher bestehenden strukturellen Defizite zu identifizieren und Überlegungen anstellen, wie diese verbessert werden können. Somit zeigt sich der Bereich der Krankenhausplanung und deren Zugänglichkeit als ein relevanter Forschungsgegenstand, der einen weiteren Forschungsbedarf aufweist.

Literaturverzeichnis

- Ahlmeyer, Florian/Wittowsky, Dirk (2018). Was brauchen wir in ländlichen Räumen? Erreichbarkeitsmodellierung als strategischer Ansatz der regionalen Standort- und Verkehrsplanung. *Raumforschung und Raumordnung | Spatial Research and Planning* 76 (6), 531–550. <https://doi.org/10.1007/s13147-018-0558-8>.
- Bast, Hannah/Delling, Daniel/Goldberg, Andrew/Müller-Hannemann, Matthias/Pajor, Thomas/Sanders, Peter/Wagner, Dorothea/Werneck, Renato F. (2016). Route Planning in Transportation Networks. In: *Algorithm Engineering*. Springer, Cham, 19–80.
- Bayeff-Filloff, M./Pawlak, C. L./Prückner, S. (2014). Vernetzte Notfallversorgung von Krankenhäusern verschiedener Versorgungsstufen. *Notfall + Rettungsmedizin* 17 (3), 202–208. <https://doi.org/10.1007/s10049-013-1788-5>.
- Bayerisches Staatsministerium des Innern, für Sport und Integration (Hg.) (2014). Richtlinie für Funkrufnamen und operativ-taktische Adressen (OPTA) der nichtpolizeilichen Behörden und Organisationen mit Sicherheitsaufgaben (npol. BOS) in Bayern. Online verfügbar unter https://www.innenministerium.bayern.de/assets/stmi/sus/feuerwehr/id2_17c_03_03_npol_bos_funkrufnamen_20141106.pdf (abgerufen am 09.02.2023).
- Beckmann, Norbert/Kriegel, Hans-Peter/Schneider, Ralf/Seeger, Bernhard (1990). The R*-tree: an efficient and robust access method for points and rectangles. In: Hector Garcia-Molina/H. V. Jagadish (Hg.). *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data, May 23-25, Atlantic City, NJ, the 1990 ACM SIGMOD international conference, Atlantic City, New Jersey, United States, 5/23/1990 - 5/26/1990*. New York, The Association for Computing Machinery, 322–331.
- Bracht, Hendrik/Hafner, Sebastian/Weiß, Manfred (2019). Sepsis-Update: Definition und Epidemiologie. *AINS - Anästhesiologie · Intensivmedizin · Notfallmedizin · Schmerztherapie* 54 (1), 10–20. <https://doi.org/10.1055/a-0625-5492>.
- Dai, Dajun/Wang, Fahui (2011). Geographic disparities in accessibility to food stores in southwest Mississippi. *Environment and Planning B: Planning and Design* 38 (4), 659–677. <https://doi.org/10.1068/b36149>.
- Destatis (2022). Grunddaten der Krankenhäuser 2020. 22. Aufl. Wiesbaden. Fachserie 12 Reihe 6.1.1. Online verfügbar unter https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Gesundheit/Krankenhaeuser/Publikationen/Downloads-Krankenhaeuser/grunddaten-krankenhaeuser-2120611207004.pdf;jsessionid=768924A6A04935ABA9196ACCAD435897.live731?__blob=publicationFile (abgerufen am 01.11.2022).
- Edlich, Richard F./Wish, John R. (2004). Maryland State Police Aviation Division. A model emergency medical system for our nation. *Journal of Long-Term Effects of Medical Implants* 14 (5), 401–414. <https://doi.org/10.1615/JLongTermEffMedImplants.v14.i5.60>.
- Eindhoven University of Technology (Hrsg.) (2023). Algorithms, Geometry & Applications. Online verfügbar unter <https://alga.win.tue.nl/tutorials/openstreetmap/> (abgerufen am 02.02.2023).
- Fischer, M./Kehrberger, E./Marung, H./Moecke, H./Prückner, S./Trentzsch, H./Urban, B. (2016). Eckpunktepapier 2016 zur notfallmedizinischen Versorgung der Bevölkerung in der Prähospitalphase und in der Klinik. *Notfall & Rettungsmedizin* 19 (5), 387–395. <https://doi.org/10.1007/s10049-016-0187-0>.
- Freyssenge, J./Renard, F./Schott, A. M./Derex, L./Nighoghossian, N./Tazarourte, K./El Khoury, C. (2018). Measurement of the potential geographic accessibility from call to definitive care for

- patient with acute stroke. *International Journal of Health Geographics* 17 (1).
<https://doi.org/10.1186/s12942-018-0121-4>.
- Geisberger, Robert/Sanders, Peter/Schultes, Dominik/Delling, Daniel (2008). Contraction Hierarchies: Faster and Simpler Hierarchical Routing in Road Networks. In: WEA 2008: Experimental Algorithms, International Workshop on Experimental and Efficient Algorithms, Provincetown, MA, USA. 5038. Aufl. Springer, Berlin, Heidelberg, 319–333.
- GIS-OPS UG (Hrsg.) (2020). Open Source Routing Engines And Algorithms - An Overview. Online verfügbar unter <https://gis-ops.com/open-source-routing-engines-and-algorithms-an-overview/#user-content-osrm> (abgerufen am 14.01.2023).
- Guagliardo, Mark F. (2004). Spatial accessibility of primary care: concepts, methods and challenges. *International Journal of Health Geographics* 3 (1), 3. <https://doi.org/10.1186/1476-072X-3-3>.
- Haynes, Robin/Jones, Andrew P./Sauerzapf, Violet/Zhao, Hongxin (2006). Validation of travel times to hospital estimated by GIS. *International Journal of Health Geographics* 5 (1).
<https://doi.org/10.1186/1476-072X-5-40>.
- Jánošíková, Ľudmila/Kvet, Marek/Jankovič, Peter/Gábrišová, Lýdia (2019). An optimization and simulation approach to emergency stations relocation. *Central European Journal of Operations Research* 27 (3), 737–758. <https://doi.org/10.1007/s10100-019-00612-5>.
- Jia, Peng/Wang, Fahui/Xierali, Imam M. (2017). Using a Huff-Based Model to Delineate Hospital Service Areas. *The Professional Geographer* 69 (4), 522–530.
<https://doi.org/10.1080/00330124.2016.1266950>.
- Khan, Abdullah A. (1992). An integrated approach to measuring potential spatial access to health care services. *Socio-Economic Planning Sciences* 26 (4), 275–287.
[https://doi.org/10.1016/0038-0121\(92\)90004-O](https://doi.org/10.1016/0038-0121(92)90004-O).
- Kim, Yeeun/Byon, Young-Ji/Yeo, Hwasoo (2018). Enhancing healthcare accessibility measurements using GIS: A case study in Seoul, Korea. *PLoS ONE* 13 (2), e0193013.
<https://doi.org/10.1371/journal.pone.0193013>.
- Kleber, C./Lefering, R./Kleber, A. J./Buschmann, C. T./Bail, H. J./Schaser, K. D./Haas, N. P. (2013). Rettungszeit und Überleben von Schwerverletzten in Deutschland. *Der Unfallchirurg* 116 (4), 345–350. <https://doi.org/10.1007/s00113-011-2132-5>.
- Koster, R. W./Baubin, M. A./Bossart, L. L./Caballero, A./Cassan, P./Castrén, M./Granja, C./Handley, A. J./Monsieurs, K. G./Perkins, G. D./Raffay, V./Sandroni, C. (2010). Basismaßnahmen zur Wiederbelebung Erwachsener und Verwendung automatisierter externer Defibrillatoren: Sektion 2 der Leitlinien zur Reanimation 2010 des European Resuscitation Council. *Notfall & Rettungsmedizin* 13 (7), 523–542. <https://doi.org/10.1007/s10049-010-1368-x>.
- Köther, H. L./Gorschlüter, P./Milz, H.-P. (2009). Interhospitaltransfer und Krankenhausfinanzierung. *Notfall + Rettungsmedizin* 12 (5), 372–378. <https://doi.org/10.1007/s10049-009-1160-y>.
- Lerner, E. B./Moscato, R. M. (2001). The golden hour: scientific fact or medical "urban legend"? *Academic emergency medicine : official journal of the Society for Academic Emergency Medicine* 8 (7), 758–760. <https://doi.org/10.1111/j.1553-2712.2001.tb00201.x>.
- Li, Chuanyao/Wang, Junren (2022). A hierarchical two-step floating catchment area analysis for high-tier hospital accessibility in an urban agglomeration region. *Journal of Transport Geography* 102, 103369. <https://doi.org/10.1016/j.jtrangeo.2022.103369>.
- Little, Wendalyn K. (2010). Golden Hour or Golden Opportunity: Early Management of Pediatric Trauma. *Clinical Pediatric Emergency Medicine* 11 (1), 4–9.
<https://doi.org/10.1016/j.cpem.2009.12.005>.

- Love, D./Lindquist, P. (1995). The geographical accessibility of hospitals to the aged: a geographic information systems analysis within Illinois. *Health Services Research* 29 (6), 629–651. Online verfügbar unter <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1070036/>.
- Luo, Wei/Qi, Yi (2009). An enhanced two-step floating catchment area (E2SFCA) method for measuring spatial accessibility to primary care physicians_. *Health & Place* 15 (4), 1100–1107. <https://doi.org/10.1016/j.healthplace.2009.06.002>.
- Luo, Wei/Wang, Fahui (2003). Measures of Spatial Accessibility to Healthcare in a GIS Environment: Synthesis and a Case Study in Chicago Region. *Environment and planning. B, Planning & design* 30 (6), 865–884. <https://doi.org/10.1068/b29120>.
- Minghini, Marco/Frassinelli, Francesco (2019). OpenStreetMap history for intrinsic quality assessment: Is OSM up-to-date? *Open Geospatial Data, Software and Standards* 4 (1), 1–17. <https://doi.org/10.1186/s40965-019-0067-x>.
- Mullen, Michael T./Wiebe, Douglas J./Bowman, Ariel/Wolff, Catherine S./Albright, Karen C./Roy, Jason/Balcer, Laura J./Branas, Charles C./Carr, Brendan G. (2014). Disparities in accessibility of certified primary stroke centers. *Stroke* 45 (11), 3381–3388. <https://doi.org/10.1161/STROKEAHA.114.006021>.
- OpenStreetMap Wiki contributors (2023). Elements. Online verfügbar unter <https://wiki.openstreetmap.org/wiki/Elements> (abgerufen am 03.02.2023).
- OpenStreetMap Wiki contributors (2023). Key:highway. Online verfügbar unter <https://wiki.openstreetmap.org/wiki/Key:highway> (abgerufen am 06.02.2023).
- OpenStreetMap Wiki contributors (2023). PBF Format. Online verfügbar unter https://wiki.openstreetmap.org/wiki/PBF_Format (abgerufen am 03.02.2023).
- OpenStreetMap Wiki (Hrsg.) (2023). OpenStreetMap Statistics. Online verfügbar unter https://planet.openstreetmap.org/statistics/data_stats.html (abgerufen am 03.02.2023).
- Pedigo, Ashley S./Odoi, Agricola (2010). Investigation of disparities in geographic accessibility to emergency stroke and myocardial infarction care in East tennessee using geographic information systems and network analysis. *Annals of epidemiology* 20 (12), 924–930. <https://doi.org/10.1016/j.annepidem.2010.06.013>.
- Radke, John/Mu, Lan (2000). Spatial Decompositions, Modeling and Mapping Service Regions to Predict Access to Social Programs. *Annals of GIS* 6 (2), 105–112. <https://doi.org/10.1080/10824000009480538>.
- Ramm, Frederik/Topf, Jochen (2010). OpenStreetMap. Die freie Weltkarte nutzen und mitgestalten. 3. Aufl. Berlin, Lehmanns.
- Rettungsdienstauschuss Bayern (2019). Strukturelle Voraussetzungen von Akutkliniken zur Behandlung von Tracer-Diagnosen. Empfehlung 02/5 - 2019 vom 13.03.2019 des Rettungsdienstauschusses Bayern. Online verfügbar unter https://www.aelrd-bayern.de/images/stories/pdf/rda/6.7_Tracerdiagnosen_.pdf (abgerufen am 18.12.2022).
- Röther, J./Dichgans, M./Bohn, A./Fischer, M./Kehrberger, E./Moecke, H./Prückner, S./Trentzsch, H./Urban, B./Marung, H. (2017). Tracerdiagnose Schlaganfall. *Der Nervenarzt* 88 (2), 180–182. <https://doi.org/10.1007/s00115-016-0236-4>.
- Schlechtriemen, T./Stratmann, D./Altemeyer, K. H. (2002). Qualitätsmanagement im Rettungsdienst - Luftrettung: Konzepte für die Zukunft. *Notfall & Rettungsmedizin* 5 (1), 47–53. <https://doi.org/10.1007/s10049-001-0408-y>.

- Schmailzl, Kurt J./Hölschermann, Frank/Schwalm, Torsten/Wilke, Enrico (2003). Das Akute Koronarsyndrom: ST-Hebungsinfarkt. *Wiener Medizinische Wochenschrift* 153 (19-20), 434–449. <https://doi.org/10.1007/s10354-003-0031-2>.
- Shannon, Gary W./Spurlock, Carl W./Gladin, Steven T./Skinner, James L. (1975). A Method For Evaluating The Geographic Accessibility Of Health Services*. *The Professional Geographer* 27 (1), 30–36. <https://doi.org/10.1111/j.0033-0124.1975.00030.x>.
- StMGP (2022). Krankenhausplan des Freistaates Bayern zum Stand 01.01.2022. 47. Fortschreibung. Bayerisches Staatsministerium für Gesundheit und Pflege. Online verfügbar unter https://www.stmgp.bayern.de/wp-content/uploads/2022/06/20220101_bayerischer-krankenhausplan.pdf (abgerufen am 07.10.2022).
- Teasdale, G./Jennett, B. (1974). Assessment of coma and impaired consciousness. A practical scale. *The Lancet* 2 (7872), 81–84. [https://doi.org/10.1016/S0140-6736\(74\)91639-0](https://doi.org/10.1016/S0140-6736(74)91639-0).
- Tobler, W.R. (1970). A Computer Movie Simulating Urban Growth in the Detroit Region. *Economic Geography* (46), 234–240. Online verfügbar unter https://www.jstor.org/stable/pdf/143141.pdf?casa_token=LWPHrE9ZlikAAAAA:A9w4BrI98OI1msTZBarbD3MgoiQkQH5oBfoffJKOty8TMmy4QA_8GFIHI-Nuijpk3IS0QHDKwxk-D2zv5mOEKq3V6HqiPYuVAsJ49a5wJd_J06zINJ5 (abgerufen am 15.03.2023).
- Valhalla Organization (Hrsg.) (2023a). Administrative data - Valhalla Docs. Online verfügbar unter <https://valhalla.github.io/valhalla/mjolnir/admins/> (abgerufen am 09.01.2023).
- Valhalla Organization (Hrsg.) (2023b). Overview of how routes are computed - Valhalla Docs. Online verfügbar unter https://valhalla.github.io/valhalla/route_overview/ (abgerufen am 09.01.2023).
- Valhalla Organization (Hrsg.) (2023c). Valhalla Docs. Online verfügbar unter <https://valhalla.github.io/valhalla/> (abgerufen am 09.01.2023).
- Velden, Lisa/Herzog, Melanie/Riedl, Wolfgang F. (2014a). Der A*-Algorithmus. Technische Universität München. Online verfügbar unter https://algorithms.discrete.ma.tum.de/graph-algorithms/spp-a-star/index_de.html (abgerufen am 11.01.2023).
- Velden, Lisa/Herzog, Melanie/Riedl, Wolfgang F. (2014b). Der Dijkstra-Algorithmus. Technische Universität München. Online verfügbar unter https://algorithms.discrete.ma.tum.de/graph-algorithms/spp-dijkstra/index_de.html (abgerufen am 11.01.2023).
- Wan, Neng/Zou, Bin/Sternberg, Troy (2012). A three-step floating catchment area method for analyzing spatial access to health services. *International Journal of Geographical Information Science* 26 (6), 1073–1089. <https://doi.org/10.1080/13658816.2011.624987>.
- Wang, Fahui (2012). Measurement, Optimization, and Impact of Health Care Accessibility: A Methodological Review. *Annals of the Association of American Geographers* 102 (5), 1104–1112. <https://doi.org/10.1080/00045608.2012.657146>.

Anhang

Anhang I: car_con.lua

```
-- Car profile
api_version = 4

Set = require('lib/set')
Sequence = require('lib/sequence')
Handlers = require("lib/way_handlers")
Relations = require("lib/relations")
find_access_tag = require("lib/access").find_access_tag
limit = require("lib/maxspeed").limit
Utils = require("lib/utils")
Measure = require("lib/measure")

function setup()
    return {
        properties = {
            max_speed_for_map_matching      = 180/3.6, -- 180km/h -> m/s
            weight_name                     = 'routability',
            process_call_tagless_node       = false,
            u_turn_penalty                   = 20,
            continue_straight_at_waypoint   = true,
            use_turn_restrictions            = true,
            left_hand_driving                = false,
            traffic_light_penalty           = 2,
        },

        default_mode                        = mode.driving,
        default_speed                       = 10,
        oneway_handling                     = true,
        side_road_multiplier                 = 0.8,
        turn_penalty                         = 7.5,
        speed_reduction                     = 0.8,
        turn_bias                           = 1.075,
        cardinal_directions                  = false,

        vehicle_height = 2.5,
        vehicle_width = 1.9,

        suffix_list = {
            'N', 'NE', 'E', 'SE', 'S', 'SW', 'W', 'NW', 'North', 'South', 'West', 'East', 'Nor', 'Sou', 'We', 'Ea'
        },

        barrier_whitelist = Set {
            'cattle_grid',
            'border_control',
            'toll_booth',
            'sally_port',
            'gate',
            'lift_gate',
            'no',
            'entrance'
        },

        access_tag_whitelist = Set {
            'yes',
            'motorcar',
            'motor_vehicle',
            'vehicle',
            'permissive',
            'designated',
            'hov'
        },

        access_tag_blacklist = Set {
            -- 'no', -- Anpassung Baustellen
            'agricultural',
            'forestry',
            'emergency',
            'psv',
            'customers',
            'private',
        }
    }
end
```

```
'delivery',
'destination'
},
restricted_access_tag_list = Set {
'private',
'delivery',
'destination',
'customers',
},
access_tags_hierarchy = Sequence {
'motorcar',
'motor_vehicle',
'vehicle',
'access'
},
service_tag_forbidden = Set {
'emergency_access'
},
restrictions = Sequence {
'motorcar', 'motor_vehicle', 'vehicle'
},
classes = Sequence {
'toll', 'motorway', 'ferry', 'restricted', 'tunnel'
},
excludable = Sequence {
Set {'toll'}, Set {'motorway'}, Set {'ferry'}
},
avoid = Set {
'area',
-- 'toll',
'reversible',
'impassable',
'hov_lanes',
'steps',
-- 'construction', Anpassung Baustellen
'proposed'
},
--Standardgeschwindigkeiten aus OSM
speeds = Sequence {
highway = {
motorway = 90,
motorway_link = 45,
trunk = 85,
trunk_link = 40,
primary = 65,
primary_link = 30,
secondary = 55,
secondary_link = 25,
tertiary = 40,
tertiary_link = 20,
unclassified = 25,
residential = 25,
living_street = 10,
service = 15
}
},
service_penalties = {
alley = 0.5,
parking = 0.5,
parking_aisle = 0.5,
driveway = 0.5,
["drive-through"] = 0.5,
["drive-thru"] = 0.5
},
restricted_highway_whitelist = Set {
'motorway',
'motorway_link',
```

```
'trunk',
'trunk_link',
'primary',
'primary_link',
'secondary',
'secondary_link',
'tertiary',
'tertiary_link',
'residential',
'living_street',
'unclassified'
},
construction_whitelist = Set {
'no', 'widening', 'minor',
},
route_speeds = {
  ferry = 5,
  shuttle_train = 10
},
bridge_speeds = {
  movable = 5
},
surface_speeds = {
  asphalt = nil,
  concrete = nil,
  ["concrete:plates"] = nil,
  ["concrete:lanes"] = nil,
  paved = nil,
  cement = 80,
  compacted = 80,
  fine_gravel = 80,
  paving_stones = 60,
  metal = 60,
  bricks = 60,
  grass = 40,
  wood = 40,
  sett = 40,
  grass_paver = 40,
  gravel = 40,
  unpaved = 40,
  ground = 40,
  dirt = 40,
  pebblestone = 40,
  tartan = 40,
  cobblestone = 30,
  clay = 30,
  earth = 20,
  stone = 20,
  rocky = 20,
  sand = 20,
  mud = 10
},
tracktype_speeds = {
  grade1 = 60,
  grade2 = 40,
  grade3 = 30,
  grade4 = 25,
  grade5 = 20
},
smoothness_speeds = {
  intermediate = 80,
  bad = 40,
  very_bad = 20,
  horrible = 10,
  very_horrible = 5,
  impassable = 0
},
maxspeed_table_default = {
  urban = 50,
```

```

    rural = 90,
    trunk = 110,
    motorway = 130
  },
  maxspeed_table = {
    ["at:rural"] = 100,
    ["at:trunk"] = 100,
    ["be:motorway"] = 120,
    ["by:urban"] = 60,
    ["by:motorway"] = 110,
    ["ch:rural"] = 80,
    ["ch:trunk"] = 100,
    ["ch:motorway"] = 120,
    ["cz:trunk"] = 0,
    ["cz:motorway"] = 0,
    ["de:living_street"] = 7,
    ["de:rural"] = 100,
    ["de:motorway"] = 0,
    ["dk:rural"] = 80,
    ["gb:ns1_single"] = (60*1609)/1000,
    ["gb:ns1_dual"] = (70*1609)/1000,
    ["gb:motorway"] = (70*1609)/1000,
    ["nl:rural"] = 80,
    ["nl:trunk"] = 100,
    ["no:rural"] = 80,
    ["no:motorway"] = 110,
    ["pl:rural"] = 100,
    ["pl:trunk"] = 120,
    ["pl:motorway"] = 140,
    ["ro:trunk"] = 100,
    ["ru:living_street"] = 20,
    ["ru:urban"] = 60,
    ["ru:motorway"] = 110,
    ["uk:ns1_single"] = (60*1609)/1000,
    ["uk:ns1_dual"] = (70*1609)/1000,
    ["uk:motorway"] = (70*1609)/1000,
    ["za:urban"] = 60,
    ["za:rural"] = 100,
    ["none"] = 140
  },
  relation_types = Sequence {
    "route"
  },
  highway_turn_classification = {
  },
  access_turn_classification = {
  }
}
end

function process_node(profile, node, result, relations)
  local access = find_access_tag(node, profile.access_tags_hierarchy)
  if access then
    if profile.access_tag_blacklist[access] and not profile.restricted_access_tag_list[access] then
      result.barrier = true
    end
  else
    local barrier = node:get_value_by_key("barrier")
    if barrier then
      local bollard = node:get_value_by_key("bollard")
      local rising_bollard = bollard and "rising" == bollard
      local restricted_by_height = false

      if barrier == "height_restricter" then
        local maxheight = Measure.get_max_height(node:get_value_by_key("maxheight"), node)
        restricted_by_height = maxheight and maxheight < profile.vehicle_height
      end

      local kerb = node:get_value_by_key("kerb")
      local highway = node:get_value_by_key("highway")
      local flat_kerb = kerb and ("lowered" == kerb or "flush" == kerb)
      local highway_crossing_kerb = barrier == "kerb" and highway and highway == "crossing"
    end
  end
end

```

```

        if not profile.barrier_whitelist[barrier]
            and not rising_bollard
            and not flat_kerb
            and not highway_crossing_kerb
            or restricted_by_height then
            result.barrier = true
        end
    end
end

local tag = node:get_value_by_key("highway")
if "traffic_signals" == tag then
    result.traffic_lights = true
end
end

function in_list (val, tab)
    for index, value in ipairs(tab) do
        if value == val then
            return true
        end
    end
    return false
end

function process_way(profile, way, result, relations)
    local data = {
        highway = way:get_value_by_key('highway'),
        bridge = way:get_value_by_key('bridge'),
        route = way:get_value_by_key('route'),
        construction = way:get_value_by_key('construction'), -- Anpassung Baustellen
    }
    -- Anpassung Baustellen
    if data.construction ~= nil and data.construction ~= 'minor' then
        data.highway = data.construction
        data.construction = nil
    end

    if (not data.highway or data.highway == '') and (not data.route or data.route == '')
    then
        return
    end
    -- Anpassung: pruefen ob Strassenkategorie unter den gultigen ist
    elseif not in_list(data.highway, {
        'residential',
        'unclassified',
        'tertiary',
        'secondary',
        'primary',
        'living_street',
        'motorway',
        'pedestrian',
        'motorway_link',
        'primary_link',
        'trunk',
        'trunk_link',
        'secondary_link',
        'road',
        'tertiary_link',
        -- 'construction', Construction darf hier als Wert nicht mehr auftauchen
        'yes',
        'mini_roundabout',
        'turning_loop',
        'turning_circle'
    })
    then
        return
    end

    handlers = Sequence {
        WayHandlers.default_mode,
        WayHandlers.blocked_ways,
        WayHandlers.avoid_ways,
        WayHandlers.handle_height,
        WayHandlers.handle_width,
        WayHandlers.access,
    }
end

```

```

WayHandlers.oneway,
WayHandlers.destinations,
WayHandlers.ferries,
WayHandlers.movables,
WayHandlers.service,
WayHandlers.hov,
WayHandlers.speed,
WayHandlers.maxspeed,
WayHandlers.surface,
WayHandlers.penalties,
WayHandlers.classes,
WayHandlers.turn_lanes,
WayHandlers.classification,
WayHandlers.roundabouts,
WayHandlers.startpoint,
WayHandlers.driving_side,
WayHandlers.names,
WayHandlers.weights,
WayHandlers.way_classification_for_turn
}

WayHandlers.run(profile, way, result, data, handlers, relations)
if profile.cardinal_directions then
    Relations.process_way_refs(way, relations, result)
end
end

function process_turn(profile, turn).
    local turn_penalty = profile.turn_penalty
    local turn_bias = turn.is_left_hand_driving and 1. / profile.turn_bias or profile.turn_bias

    if turn.has_traffic_light then
        turn.duration = profile.properties.traffic_light_penalty
    end

    if turn.number_of_roads > 2 or turn.source_mode ~= turn.target_mode or turn.is_u_turn then
        if turn.angle >= 0 then
            turn.duration = turn.duration + turn_penalty / (1 + math.exp( -((13 / turn_bias)
* turn.angle/180 - 6.5*turn_bias)))
        else
            turn.duration = turn.duration + turn_penalty / (1 + math.exp( -((13 * turn_bias) * -
turn.angle/180 - 6.5/turn_bias)))
        end

        if turn.is_u_turn then
            turn.duration = turn.duration + profile.properties.u_turn_penalty
        end
    end

    if profile.properties.weight_name == 'routability' then
        if not turn.source_restricted and turn.target_restricted then
            turn.weight = constants.max_turn_weight
        end
    end
end
end

return {
    setup = setup,
    process_way = process_way,
    process_node = process_node,
    process_turn = process_turn
}

```

Anhang II: osm2sqlite.py

```

# -*- coding: utf-8 -*-
"""
Schreibt die OSM-Daten in eine SQLite-Tabelle,
diese wird später genutzt, um die Anteile der verschiedenen Straßenkategorien zu ermitteln
"""
import os, sys, csv, time
import xml.etree.ElementTree as etree
import sqlite3 as lite

```

```

def create_connection(db_file):
    dbconnect = None
    try:
        dbconnect = lite.connect(db_file)
        return dbconnect
    except Error as e:
        print(e)
    return dbconnect

os.chdir(r'C:\directory_path\OSM_DATA')
database = r"C:\file_path\database_name.sqlite"
dbconnect = create_connection(database)
cursor = dbconnect.cursor()
if dbconnect is not None:
    sql_command = """
    CREATE TABLE IF NOT EXISTS EDGES (
        node_von INTEGER,
        node_nach INTEGER,
        id INTEGER,
        isarea INTEGER,
        osm_highway TEXT, -- Attribut highway aus OSM
        osm_maxspeed TEXT, -- Attribut maxspeed aus OSM
        kategorie TEXT
    );"""
    cursor.execute(sql_command)
else:
    print("Fehler! Datenbankverbindung fehlgeschlagen.")

insert_edge = """INSERT INTO EDGES (node_von, node_nach, id, isarea, osm_highway, osm_maxspeed,
kategorie)
VALUES ("{node_von}", "{node_nach}", "{id}", "{isarea}", "{osm_highway}", "{osm_maxspeed}",
"{kategorie}");"""

kat_gueltig = ['residential',
'unclassified',
'tertiary',
'secondary',
'primary',
'living_street',
'motorway',
'pedestrian',
'motorway_link',
'primary_link',
'trunk',
'trunk_link',
'secondary_link',
'road',
'tertiary_link',
'yes',
'mini_roundabout',
'turning_loop',
'turning_circle',
'track',
'service'
]

n = 0
for event, elem in etree.iterparse('bayern-latest.osm'):
    if elem.tag=='way':
        id = elem.attrib['id']
        nodes = [nd.attrib['ref'] for nd in elem.findall("nd")] # Liste der IDs der beteiligten Nodes
        edges = map(lambda x, y: (min(int(x), int(y)), max(int(x), int(y))), nodes[0:-1], nodes[1:])

        tags = elem.findall("tag")

        fclass = None
        maxspeed = 'x'
        isArea = 0
        junction = None
        speed_zone = None # Maximalgeschwindigkeit aus zone:traffic-Attribut
        speed_maxspeed = None # Maximalgeschwindigkeit aus maxspeed-Attribut
        speed_source = None # Maximalgeschwindigkeit aus source:maxspeed-Attribut
        construction = None

        for tag in tags:
            if tag.attrib['k'] == 'highway':
                fclass = tag.attrib['v'].strip()

```

```

# eigentliche Straßenkategorie im Wert des Attributs "construction"
# construction mit dem Attribut minor sind Baustellen, die befahrbar sind
if (tag.attrib['k'] == 'construction') and (tag.attrib['v'] != 'minor'):
    construction = tag.attrib['v'].strip()

if tag.attrib['k'] == 'zone:traffic':
    if tag.attrib['v'] == 'DE:motorway':
        speed_zone = '0' # keine Höchstgeschwindigkeit
    elif tag.attrib['v'] in ['DE:rural']:
        speed_zone = '100' # Höchstgeschwindigkeit 100km/h
    elif tag.attrib['v'] in ['DE:urban', 'AT:urban', 'urban']:
        speed_zone = '50' # Höchstgeschwindigkeit 50km/h
    elif tag.attrib['v'] in ['DE:30', '30', 'DE:speed:30']:
        speed_zone = '30' # Höchstgeschwindigkeit 30km/h

if tag.attrib['k'] == 'zone:maxspeed':
    if tag.attrib['v'] in ['DE:motorway']:
        speed_zone = '0' # keine Höchstgeschwindigkeit
    elif tag.attrib['v'] in ['DE:rural']:
        speed_zone = '100' # Höchstgeschwindigkeit 100km/h
    elif tag.attrib['v'] in ['DE:urban', 'DE:50', 'AT:50']:
        speed_zone = '50' # Höchstgeschwindigkeit 50km/h
    elif tag.attrib['v'] in ['DE:40']:
        speed_zone = '40' # Höchstgeschwindigkeit 40km/h
    elif tag.attrib['v'] in ['DE:30', 'de:30', 'AT:30', '30']:
        speed_zone = '30' # Höchstgeschwindigkeit 30km/h
    elif tag.attrib['v'] in ['DE:20']:
        speed_zone = '20' # Höchstgeschwindigkeit 20km/h
    elif tag.attrib['v'] in ['DE:10']:
        speed_zone = '10' # Höchstgeschwindigkeit 10km/h

if tag.attrib['k'] == 'source:maxspeed':
    if tag.attrib['v'] in ['DE:motorway', 'sign;DE:motorway']:
        speed_source = '0' # keine Höchstgeschwindigkeit
    elif tag.attrib['v'] in ['AT:motorway', '130']:
        speed_source = '130' # Höchstgeschwindigkeit 130km/h
    elif tag.attrib['v'] in ['DE:rural', 'de:rural', 'AT:rural', 'DE:rural;sign', 'rural',
'sign;DE:rural']:
        speed_source = '100' # Höchstgeschwindigkeit 100km/h
    elif tag.attrib['v'] in ['CZ:rural']:
        speed_source = '90' # Höchstgeschwindigkeit 90km/h
    elif tag.attrib['v'] in ['80']:
        speed_source = '80' # Höchstgeschwindigkeit 80km/h
    elif tag.attrib['v'] in ['DE:urban', 'de:urban', 'AT:urban', 'DE:urban', 'CZ:urban',
'urban', '50', 'DE:urban;sign', 'de_urban', 'sign;DE:urban', 'source:maxspeed=DE:urban']:
        speed_source = '50' # Höchstgeschwindigkeit 50km/h
    elif tag.attrib['v'] in ['AT:zone40']:
        speed_source = '40' # Höchstgeschwindigkeit 40km/h
    elif tag.attrib['v'] in ['zone', 'zone30', 'DE:zone', 'DE:zone30', 'de:zone30',
'DE:274[30]', 'AT:zone30', 'AT:zone:30', 'DE:zone:30', 'DE:274-30', 'DE:zone:30;DE:bicycle_road',
'DE:30', 'sign:zone30', 'DE:30zone', 'sign DE:274-30', 'DE:bicycle_road', 'DE:bicycle_zone',
'DE:Fahrradstraße']:
        speed_source = '30' # Höchstgeschwindigkeit 30km/h
    elif tag.attrib['v'] in ['DE:zone:20', 'DE:zone20', 'CZ:living_street']:
        speed_source = '20' # Höchstgeschwindigkeit 20km/h
    elif tag.attrib['v'] in ['DE:zone:10', 'DE:zone10']:
        speed_source = '10' # Höchstgeschwindigkeit 10km/h

if tag.attrib['k'] == 'maxspeed':
    speed_maxspeed = tag.attrib['v'].strip()

if tag.attrib['k'] == 'area':
    if tag.attrib['v'].strip() == 'yes':
        isArea = 1

if tag.attrib['k'] == 'junction':
    junction = tag.attrib['v'].strip()

if speed_maxspeed is not None:
    maxspeed = speed_maxspeed
elif speed_zone is not None:
    maxspeed = speed_zone
elif speed_source is not None:
    maxspeed = speed_source

```

```
if maxspeed.isdigit():
    maxspeed = int(maxspeed)
else:
    maxspeed = -1

# bei Baustellen den Wert des Attributs construction als fclass setzen
if (construction in kat_guelstig) and (fclass == 'construction'):
    fclass = construction

# Klassen zusammenfassen:
# Autobahnen
if fclass == 'motorway':
    if maxspeed > 0 and maxspeed <= 80:
        kat = 'autobahn_80'
    elif maxspeed > 80 and maxspeed <= 100:
        kat = 'autobahn_100'
    elif maxspeed > 100 and maxspeed <= 120:
        kat = 'autobahn_120'
    elif maxspeed > 120:
        kat = 'autobahn_x'
    else:
        kat = 'autobahn_x'
elif fclass == 'motorway_link':
    kat = 'autobahn_link'

# Schnellstraßen
elif fclass == 'trunk':
    if maxspeed > 0 and maxspeed <= 60:
        kat = 'schnell_60'
    elif maxspeed > 60 and maxspeed <= 100:
        kat = 'schnell_100'
    elif maxspeed > 100:
        kat = 'schnell_x'
    else:
        kat = 'schnell_x'

# Fußgängerzone
elif fclass == 'pedestrian':
    kat = 'fussgaenger'

# Bundesstraße
elif fclass == 'primary':
    if maxspeed > 0 and maxspeed <= 50:
        kat = 'bundes_50'
    elif maxspeed > 50 and maxspeed <= 80:
        kat = 'bundes_80'
    elif maxspeed > 80:
        kat = 'bundes_x'
    else:
        kat = 'bundes_x'

# Landes- und Kreisstraßen
elif fclass in ('secondary', 'tertiary'):
    if maxspeed > 0 and maxspeed <= 30:
        kat = 'land_30'
    elif maxspeed > 30 and maxspeed <= 50:
        kat = 'land_50'
    elif maxspeed > 50:
        kat = 'land_x'
    else:
        kat = 'land_x'

# Bundes-/Landes-/Schnellstraßenlinks in einer Klasse
elif fclass in ['primary_link', 'secondary_link', 'tertiary_link', 'trunk_link']:
    kat = 'link'

# Zufahrtswege
elif fclass == 'service':
    kat = 'zufahrt'

# Feldwege
elif fclass == 'track':
    kat = 'feldweg'

# Nebenstraßen
elif fclass == 'unclassified':
    if maxspeed > 0 and maxspeed <= 50:
```

```

        kat = 'neben_50'
    elif maxspeed > 50:
        kat = 'neben_x'
    else:
        kat = 'neben_x'

# Anliegerstraßen
elif fclass == 'residential':
    kat = 'anlieger_50'

# Spielstraße / verkehrsberuhigter Bereich
elif fclass == 'living_street':
    kat = 'spiel'

# nicht explizit definierte Kategorien:
else:
    kat = 'sonstige'

# Kreisverkehr: mini_roundabout = Kreisverkehr mit befahrbarer Mittelinsel
if junction == 'roundabout' or fclass == 'mini_roundabout':
    kat = 'kreisverkehr'

if fclass in kat_gueltig:
    for edge in edges:
        if edge[0] > edge[1]:
            print (edge[0], edge[1])
            sys.exit('Ungültige Reihenfolge!')
        try:
            sql_command = insert_edge.format(node_von=edge[0], node_nach=edge[1], id=id,
isarea=isArea, osm_highway=fclass, osm_maxspeed=maxspeed, kategorie=kat)
            cursor.execute(sql_command)
        except:
            print ('Nicht geschrieben: ')
            print (edge[0], edge[1], id, isArea, fclass, maxspeed)

    n += 1
    if n % 50000 == 0:
        dbconnect.commit()
    elem.clear()
dbconnect.commit()

sql_command = """CREATE INDEX IF NOT EXISTS EDGES_IDX ON EDGES (node_von, node_nach);"""
cursor.execute(sql_command)
dbconnect.close()

```

Anhang III: calc_routes.py

```

# -*- coding: utf-8 -*-
"""
Skript ermittelt die Anteile der verschiedenen Straßentypen bei Routing mit Standard-Profil (car_con.lua)
Die Anteile werden benötigt, um später die Fahrzeiten für verschiedene Straßentypen zu berechnen
"""
import urllib.request, urllib.error, json, os, sys
import sqlite3 as lite
import pandas as pd
import os.path
import numpy as np
from pprint import pprint

class Point(object):
    def __init__(self, x, y):
        self.x=x
        self.y=y
    def __str__(self):
        return '(' + str(self.x) + ' / ' + str(self.y) + ')'

class Fahrt(object):
    def __init__(self, startX, startY, zielX, zielY, fahrzeit):
        self.start = Point(startX, startY)
        self.ziel = Point(zielX, zielY)
        self.gemessen = float(fahrzeit)
        self.doRouting()

    if self.route["code"] == 'NoRoute':
        print ('Keine Route gefunden: ' + self.route["message"])
    else:

```

```

self.segmente = pd.DataFrame({'von': self.getNodes()[0:-1],
                             'nach': self.getNodes()[1:],
                             'distanz': self.getDistances(),
                             'kategorie': None
                             })
for index, row in self.segmente.iterrows():
    von_node = min(int(row['von']), int(row['nach']))
    nach_node = max(int(row['von']), int(row['nach']))

    cursor.execute('SELECT kategorie FROM edges WHERE node_von=? and node_nach=? order by
isarea asc,cast(osm_maxspeed as integer) desc limit 1', (von_node, nach_node))
    res = cursor.fetchone()
    if not res:
        kat = 'xxx'
    else:
        kat = res[0]
    self.segmente.at[index, 'kategorie'] = kat # ermittelte Kategorie

self.anteile = self.segmente.groupby(by='kategorie', as_index=False)['distanz'].sum()
self.anteile = self.anteile.set_index('kategorie').T # Zeilen in Spalten
self.anteile["fz_einsatz"] = self.gemessen
self.anteile["fz_osm"] = self.getDuration() / 60.0

if float(self.gemessen) == 0.0:
    v_gemessen = 0.0
else:
    v_gemessen = round((self.getLength() / float(self.gemessen)) * 0.06 , 1) # km/h
self.anteile["v_einsatz"] = v_gemessen

del (self.segmente)

def doRouting(self):
    server = 'http://127.0.0.1:5000/route/v1/driving/'
    options = '?overview=false&alternatives=false&steps=false&annotations=true&generate_hints=false'
    self.request = server + str(self.start.x) + ',' + str(self.start.y) + ';' + str(self.ziel.x) +
', ' + str(self.ziel.y) + options

    try:
        f = urllib.request.urlopen(self.request, timeout=120)
        result = f.read()
        self.route = json.loads(result)
    except urllib.error.HTTPError as e:
        print ('HTTPError ' + str(e.code))
        print ('Request:')
        print (self.request)
        if e.code == 400:
            result = e.read()
            self.route = json.loads(result)
    except Exception:
        import traceback
        print ('generic exception: ' + traceback.format_exc())

def getNodes(self):
    # befahrene Knoten
    return self.route["routes"][0]["legs"][0]["annotation"]["nodes"]

def getDistances(self):
    # Abstände zwischen den Knoten
    return self.route["routes"][0]["legs"][0]["annotation"]["distance"]

def getLength(self):
    # Gesamt-Distanz
    return float(self.route["routes"][0]["distance"])

def getDuration(self):
    # Gesamt-Fahrzeit
    return float(self.route["routes"][0]["duration"])

os.chdir(r'C:\directory_path')
datei = r'filename.csv' # Einsatzdaten
database = r"C:\file_path\database_name.sqlite"
def create_connection(db_file):
    dbconnect = None
    try:
        dbconnect = lite.connect(db_file)
        return dbconnect
    except Error as e:
        print(e)

```

```

    return dbconnect

dbconnect = create_connection(database)
cursor = dbconnect.cursor()

fahrten = pd.read_csv(datei, sep=',', decimal = '.')
routen = []
anzahl = len(fahrten.index)
n = 0
fehlerhafte = 0 # nicht erfolgreich geroutete Strecken
for index, row in fahrten.iterrows():
    n += 1
    trip = Fahrt(str(row['VON_X']), str(row['VON_Y']), str(row['NACH_X']), str(row['NACH_Y']),
str(row['FAHRZEIT']))
    if trip.route["code"] == 'Ok':
        routen.append(trip)
    else:
        fehlerhafte += 1

gesamt = pd.concat((fahrt.anteile for fahrt in routen), sort=True)
gesamt.fillna(value=0, inplace=True)
gesamt.to_csv(os.path.splitext(datei)[0] + r'_GEROUTET.csv', sep=';', decimal=',')

dbconnect.close()

```

Anhang IV: leastsquares.py

```

# -*- coding: utf-8 -*-
import os, sys, time
import pandas as pd
import numpy as np
import scipy.optimize as opt

os.chdir(r'C:\directory_path')
trips = pd.read_csv(r'filename_GEROUTET.csv', sep=',', decimal='.')

ausgeschlossene_spalten = ['FZ_EINSATZ', 'FZ_OSM', 'V_EINSATZ', 'PK_UID', 'DISTANZ']
kategorien = [x for x in (trips.columns.values.tolist())[1:] if x not in ausgeschlossene_spalten]

fahrten = trips[kategorien]
fahrzeiten = trips['FZ_EINSATZ']

fahrten_np = fahrten.values
fahrzeiten_np = fahrzeiten.values.flatten()

def toKmh(x):
    if x == 0.0:
        return 0.0
    else:
        return (60.0 / x) / 1000.0

#-----Lineare Regression-Berechnung-----#
from sklearn import linear_model
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from matplotlib import pyplot as plt
import seaborn as sns

X = fahrten_np
y = fahrzeiten_np

model = LinearRegression(positive=True)
model.fit(X, y)

print("y-Achsen-Abschnitt:", model.intercept_)
print("Steigung:", model.coef_)
print("R²:", model.score(X, y))

geschwindigkeiten = [toKmh(x) for x in model.coef_]

file = open('v_sp_' + str(int(time.time())) + '.txt', 'w')
for ele in zip(kategorien, geschwindigkeiten):
    print(ele[0] + ' = ' + str(round(ele[1], 2)) + ',')
    file.write(ele[0] + '\t' + str(round(ele[1], 2)).replace('.', ',') + '\r\n')

```

```
file.close()
```

Anhang V: RTW_con.lua

```
-- RTW profile
api_version = 4
Set = require('lib/set')

function setup()
  return {
    properties = {
      max_speed_for_map_matching = 130/3.6, -- m/s
      weight_name = 'duration',
      process_call_tagless_node = false,
      u_turn_penalty = 20,
      continue_straight_at_waypoint = true,
      use_turn_restrictions = true,
      traffic_light_penalty = 2, --seconds
    },
    default_mode = mode.driving,
    default_speed = 30, -- eigene Schätzung
    vehicle_height = 2.5,
    vehicle_width = 1.9,

    speeds = {
      autobahn_x = 133.81,
      autobahn_120 = 129.16,
      autobahn_100 = 105.47,
      autobahn_80 = 88.66,
      autobahn_link = 59.98,
      schnell_x = 135.72,
      schnell_100 = 110.49,
      schnell_60 = 59.56,
      bundes_x = 106.91,
      bundes_80 = 79.76,
      bundes_50 = 55.96,
      land_x = 87.72,
      land_50 = 56.56,
      land_30 = 36.69,
      neben_x = 65.54,
      neben_50 = 54.49,
      anlieger_50 = 44.76,
      spiel = 20.42,
      fussgaenger = 14.68,
      kreisverkehr = 60.02,
      link = 50.29,
      zufahrt = 30.0, -- manuell festgelegt basierend auf OSM
      feldweg = 7.0, -- manuell festgelegte Schrittgeschwindigkeit
      sonstige = 30.0 -- manuell festgelegt basierend auf OSM
    },

    fclass_whitelist = Set {
      'motorway',
      'motorway_link',
      'trunk',
      'trunk_link',
      'primary',
      'primary_link',
      'secondary',
      'secondary_link',
      'tertiary',
      'tertiary_link',
      'unclassified',
      'residential',
      'living_street',
      'pedestrian',
      'construction', -- Construction später mit der zugrundeliegenden Straßenkategorie überschrieben
      'service',
      'track',
    },
  }
end
```

```
function in_list (val, tab)
  for index, value in ipairs(tab) do
    if value == val then
      return true
    end
  end
  return false
end

-- Funktion um die ursprünglichen OSM-Straßenkategorien den neuen Kategorien zuzuweisen
function getKategorie (fclass, junction, maxspeed)
  -- Kreisverkehr
  if junction == 'roundabout' then
    return 'kreisverkehr'
  end

  -- Autobahn
  if fclass == 'motorway' then
    if maxspeed > 0 and maxspeed <= 80 then
      return 'autobahn_80'
    elseif maxspeed > 80 and maxspeed <= 100 then
      return 'autobahn_100'
    elseif maxspeed > 100 and maxspeed <= 120 then
      return 'autobahn_120'
    elseif maxspeed > 120 then
      return 'autobahn_x'
    else
      return 'autobahn_x'
    end
  end

  -- Autobahnauffahrt
  elseif fclass == 'motorway_link' then
    return 'autobahn_link'
  end

  -- Schnellstraße
  elseif fclass == 'trunk' then
    if maxspeed > 0 and maxspeed <= 60 then
      return 'schnell_60'
    elseif maxspeed > 60 and maxspeed <= 100 then
      return 'schnell_100'
    elseif maxspeed > 100 then
      return 'schnell_x'
    else
      return 'schnell_x'
    end
  end

  -- Fußgängerzone
  elseif fclass == 'pedestrian' then
    return 'fussgaenger'
  end

  -- Bundesstraße
  elseif fclass == 'primary' then
    if maxspeed > 0 and maxspeed <= 50 then
      return 'bundes_50'
    elseif maxspeed > 50 and maxspeed <= 80 then
      return 'bundes_80'
    elseif maxspeed > 80 then
      return 'bundes_x'
    else
      return 'bundes_x'
    end
  end

  -- Landes- und Kreisstraßen
  elseif fclass == 'secondary' or fclass == 'tertiary' then
    if maxspeed > 0 and maxspeed <= 30 then
      return 'landes_30'
    elseif maxspeed > 30 and maxspeed <= 50 then
      return 'landes_50'
    elseif maxspeed > 50 then
      return 'landes_x'
    else
      return 'landes_x'
    end
  end

  -- Auffahrten
```

```

elseif fclass == 'primary_link' or fclass == 'secondary_link' or fclass == 'tertiary_link' or
fclass == 'trunk_link' then
    return 'link'

-- Zufahrtswege
elseif fclass == 'service' then
    return 'zufahrt'

-- Feldwege
elseif fclass == 'track' then
    return 'feldweg'

-- Nebenstraßen
elseif fclass == 'unclassified' then
    if maxspeed > 0 and maxspeed <= 50 then
        return 'neben_50'
    elseif maxspeed > 50 then
        return 'neben_x'
    else
        return 'neben_x'
    end
end

-- Anliegerstraßen
elseif fclass == 'residential' then
    return 'anlieger_50'

-- Spielstraße / verkehrsberuhigter Bereich
elseif fclass == 'living_street' then
    return 'spiel'

-- Baustelle
elseif fclass == 'construction' then
    return 'baustelle'

-- nicht explizit definierte Kategorien
else
    return 'sonstige'
end
end

-- Funktion um die OSM-Höchstgeschwindigkeiten zu erhalten, als Basis für die Zuordnung der OSM-
Straßenkategorien zu den neuen Straßenkategorien
function getSpeed(maxspeed, zone_t, zone_m, source)
    if maxspeed ~= nil then
        return maxspeed
    elseif zone_t ~= nil then
        if zone_t == 'DE:urban' or zone == 'AT:urban' or zone == 'urban' then
            return 50
        elseif zone_t == 'DE:rural' then
            return 100
        elseif zone_t == 'DE:motorway' then
            return 0
        elseif zone_t == 'DE:30' or zone == '30' or zone == 'DE:speed:30' then
            return 30
        end
    elseif zone_m ~= nil then
        if zone_m == 'DE:urban' or zone_m == 'DE:50' or zone_m == 'AT:50' then
            return 50
        elseif zone_m == 'DE:rural' then
            return 100
        elseif zone_m == 'DE:motorway' then
            return 0
        elseif zone_m == 'DE:30' or zone_m == '30' or zone_m == 'de:30' or zone_m == 'AT:30' then
            return 30
        elseif zone_m == 'DE:40' then
            return 40
        elseif zone_m == 'DE:20' then
            return 20
        elseif zone_m == 'DE:10' then
            return 10
        end
    elseif source ~= nil then
        if in_list(source, {'DE:urban', 'de:urban', 'AT:urban', 'DE:urban', 'CZ:urban', 'urban', '50',
'DE:urban;sign', 'de_urban', 'sign;DE:urban', 'source:maxspeed=DE:urban'}) then
            return 50
        end
    end
end

```

```

        elseif in_list(source, {'DE:rural', 'de:rural', 'AT:rural', 'DE:rural;sign', 'rural',
'sign;DE:rural'}) then
            return 100
        elseif source == 'CZ:rural' then
            return 90
        elseif source == 'DE:motorway' or source == 'sign;DE:motorway' then
            return 0
        elseif in_list(source, {'AT:motorway', '130'}) then
            return 130
        elseif source == '80' then
            return 80
        elseif source == 'AT:zone40' then
            return 40
        elseif in_list(source, {'zone', 'zone30', 'DE:zone', 'DE:zone30', 'de:zone30', 'DE:274[30]',
'AT:zone30', 'AT:zone:30', 'DE:zone:30', 'DE:274-30', 'DE:zone:30;DE:bicycle_road', 'DE:30',
'sign:zone30', 'DE:30zone', 'sign DE:274-30', 'DE:bicycle_road', 'DE:bicycle_zone',
'DE:Fahrradstraße'}) then
            return 30
        elseif in_list(source, {'DE:zone:20', 'DE:zone20', 'CZ:living_street'}) then
            return 20
        elseif in_list(source, {'DE:zone:10', 'DE:zone10'}) then
            return 10
        end
    end
    return 0
end

function process_node (profile, node, result)
    local traffic_signal = node:get_value_by_key("highway")
    if traffic_signal and traffic_signal == "traffic_signals" then
        result.traffic_lights = true
    end
end

function process_way(profile, way, result)
    local highway = way:get_value_by_key("highway")
    local route = way:get_value_by_key("route")
    if (not highway or highway == '') and (not route or route == '') then
        return
    elseif not profile.fclass_whitelist[highway] then
        return
    end
    local name = way:get_value_by_key("name")
    local oneway = way:get_value_by_key("oneway")
    local oneway_emergency = way:get_value_by_key("oneway:emergency")
    local duration = way:get_value_by_key("duration")
    local maxspeed = tonumber(way:get_value_by_key ("maxspeed"))
    local maxspeed_forward = tonumber(way:get_value_by_key( "maxspeed:forward"))
    local maxspeed_backward = tonumber(way:get_value_by_key( "maxspeed:backward"))
    local speed_zone_t = way:get_value_by_key("zone:traffic")
    local speed_source = way:get_value_by_key("source:maxspeed")
    local speed_zone_m = way:get_value_by_key("zone:maxspeed")
    local junction = way:get_value_by_key("junction")
    local construction = way:get_value_by_key("construction")

    -- bei Baustellen wird der Wert des Attributs construction als fclass gesetzt
    -- außer bei kleinen (minor) Baustellen
    if construction ~= nil and construction ~= '' and construction ~= 'minor' then
        fclass = construction
    end

    if name then
        result.name = name
    end

    osm_speed = getSpeed(maxspeed, speed_zone_t, speed_zone_m, speed_source)
    kat = getKategorie(highway, junction, osm_speed)

    result.forward_mode = mode.driving
    result.backward_mode = mode.driving
    result.forward_speed = profile.speeds[kat] or profile.default_speed
    result.backward_speed = profile.speeds[kat] or profile.default_speed

    -- wenn oneway:emergency definiert ist, wird dieser Wert für das Attribut oneway übernommen
    if oneway_emergency ~= nil then
        oneway = oneway_emergency
    end
end

```

```

end
if oneway == "no" or oneway == "0" or oneway == "false" then
  -- nothing to do
elseif oneway == "-1" then
  result.forward_mode = mode.inaccessible
elseif oneway == "yes" or oneway == "1" or oneway == "true" or junction == "roundabout" then
  result.backward_mode = mode.inaccessible
end

if junction == 'roundabout' then
  result.roundabout = true
end
end

function process_turn (profile, turn)
  if turn.is_u_turn then
    turn.duration = turn.duration + profile.properties.u_turn_penalty
    turn.weight = turn.weight + profile.properties.u_turn_penalty
  end
  if turn.has_traffic_light then
    turn.duration = turn.duration + profile.properties.traffic_light_penalty
  end
end

return {
  setup = setup,
  process_way = process_way,
  process_node = process_node,
  process_turn = process_turn
}

```

Anhang VI: calc_fz.py

```

# -*- coding: utf-8 -*-
"""
Mit neuem RTW-Geschwindigkeitsprofil routen, als Basis zur Berechnung des Korrekturfaktors.
Skript ermittelt die vorläufige Fahrzeit über das Routing und erstellt die Eingabetabelle zur
Berechnung des Korrekturfaktors.
"""
import pandas as pd
import numpy as np
import pprint, urllib.request, urllib.error, json, os

os.chdir(r'C:\directory_path')
df = pd.read_csv(r'filename.csv', sep=',', decimal='.') # Einsatzdaten

df["FZ_RTW"] = np.nan # leere Spalte einfüegen
df["DIST_RTW"] = np.nan # leere Spalte einfüegen
df["V_RTW"] = np.nan # leere Spalte einfüegen
df["V_EINSATZ"] = np.nan # leere Spalte einfüegen

server = 'http://127.0.0.1:5000/route/v1/driving/'
options = '?overview=false&alternatives=false&steps=false&annotations=false&generate_hints=false'

for index, row in df.iterrows():

    request = server + str(row['VON_X']) + ',' + str(row['VON_Y']) + ';' + str(row['NACH_X']) + ',' +
    str(row['NACH_Y']) + options

    try:
        f = urllib.request.urlopen(request)
        result = f.read()
        route = json.loads(result)
        fz = float(route["routes"][0]["legs"][0]["duration"])/60 # Minuten
        dist = float(route["routes"][0]["legs"][0]["distance"])/1000 # Kilometer

        if fz == 0:
            v = 0
        else:
            v = round((dist / fz) * 60, 1) # km/h

        fahrzeit = float(row["FZ_EINSATZ"]) # Fahrzeit laut Einsatzdaten
        if fahrzeit == 0:
            v_gemessen = 0

```

```

else:
    v_gemessen = round((dist / fahrzeit) * 60, 1) # km/h

df.at[index, 'FZ_RTW'] = fz # Fahrzeit laut Routing
df.at[index, 'DIST_RTW'] = dist # Distanz im Routing
df.at[index, 'V_RTW'] = v # durchschnittliche Geschwindigkeit im Routing
df.at[index, 'V_EINSATZ'] = v_gemessen # Geschwindigkeit aus Einsatzfahrzeit und OSM-Distanz

except urllib.error.HTTPError as e:
    print ('HTTPError ' + str(e.code))
    print ('Request:')
    print (request)
    if e.code == 400:
        result = e.read()
        route = json.loads(result)

except Exception:
    import traceback
    print ('generic exception: ' + traceback.format_exc())

except:
    print ('Berechnungsfehler bei Zeile ' + str(index))

df.to_csv(r'OUTPUT_filename.csv', sep=';', decimal=',')

```

Anhang VII: reg_koeff.py

```

# -*- coding: utf-8 -*-
"""
berechnet die Regressionskoeffizienten für die Gleichung
zur Abschätzung der Routing-Geschwindigkeiten
"""

import os, sys
import pandas as pd
import numpy as np
import scipy.optimize as opt
from matplotlib import pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression

os.chdir(r'C:\directory_path')
trips = pd.read_csv(r'filename.csv', sep=',', decimal='.')

einsatz = trips['FZ_EINSATZ']
osm = trips['FZ_RTW']
df = trips[['FZ_EINSATZ', 'FZ_RTW']]

X = einsatz.values.flatten()
X = X.reshape(-1,1)

y = osm.values.flatten()

#-----Lineare Regression-Berechnung-----#
from sklearn.linear_model import LinearRegression

model = LinearRegression()
model.fit(X, y)

print("y-Achsen-Abschnitt:", model.intercept_)
print("Steigung:", model.coef_)
print("R²:", model.score(X, y))

sns.jointplot(x="FZ_EINSATZ", y="FZ_RTW", data=trips, kind="reg", line_kws={"color": "C1"},
scatter_kws={'s': 2})

plt.show()

```

Anhang VIII: DM_GEM_KH.py

```

# -*- coding: utf-8 -*-
"""
berechnet die Distanzmatrix zwischen Krankenhäuser und Gemeinden;
Krankenhäuser werden nicht differenziert nach Eignung für Tracer-Diagnose
"""

```

```
import urllib.request,urllib.error,os,json
os.chdir(r'C:\directory_path')
df = pd.read_csv(r'filename.csv', sep=',', decimal = '.') # kartesisches Produkt
df["FZ_RTW_RAW"] = np.nan # leere Spalte einfüegen
df["FZ_RTW"] = np.nan # leere Spalte einfüegen
df["DIST_RTW"] = np.nan # leere Spalte einfüegen

server = 'http://127.0.0.1:5000/route/v1/driving/'
options = '?overview=false&alternatives=false&steps=false&annotations=false&generate_hints=false'

for index, row in df.iterrows():
    request = server + str(row['VON_X']) + ',' + str(row['VON_Y']) + ';' + str(row['NACH_X']) + ',' +
    str(row['NACH_Y']) + options
    try:
        f = urllib.request.urlopen(request)
        result = f.read()
        route = json.loads(result)
        fz = float(route["routes"][0]["legs"][0]["duration"])/60 # Minuten
        dist = float(route["routes"][0]["legs"][0]["distance"])/1000 # Kilometer

        df.at[index, 'FZ_RTW_RAW']= fz # Fahrzeit laut Routing
        df.at[index, 'FZ_RTW']= fz+2.1877 # geschätzte Fahrzeit entsprechend Regressionsgleichung
        df.at[index, 'DIST_RTW']= dist # Distanz im Routing
    except urllib.error.HTTPError as e:
        print ('HTTPError ' + str(e.code))
        print ('Request:')
        print (request)
        if e.code == 400:
            result = e.read()
            route = json.loads(result)
    except Exception:
        import traceback
        print ('generic exception: ' + traceback.format_exc())
    except:
        print ('### Berechnungsfehler bei Zeile ' + str(index))

df.to_csv(r'OUTPUT_filename.csv', sep=',', decimal='.')
```

Anhang IX: Karte Landesentwicklungsprogramm

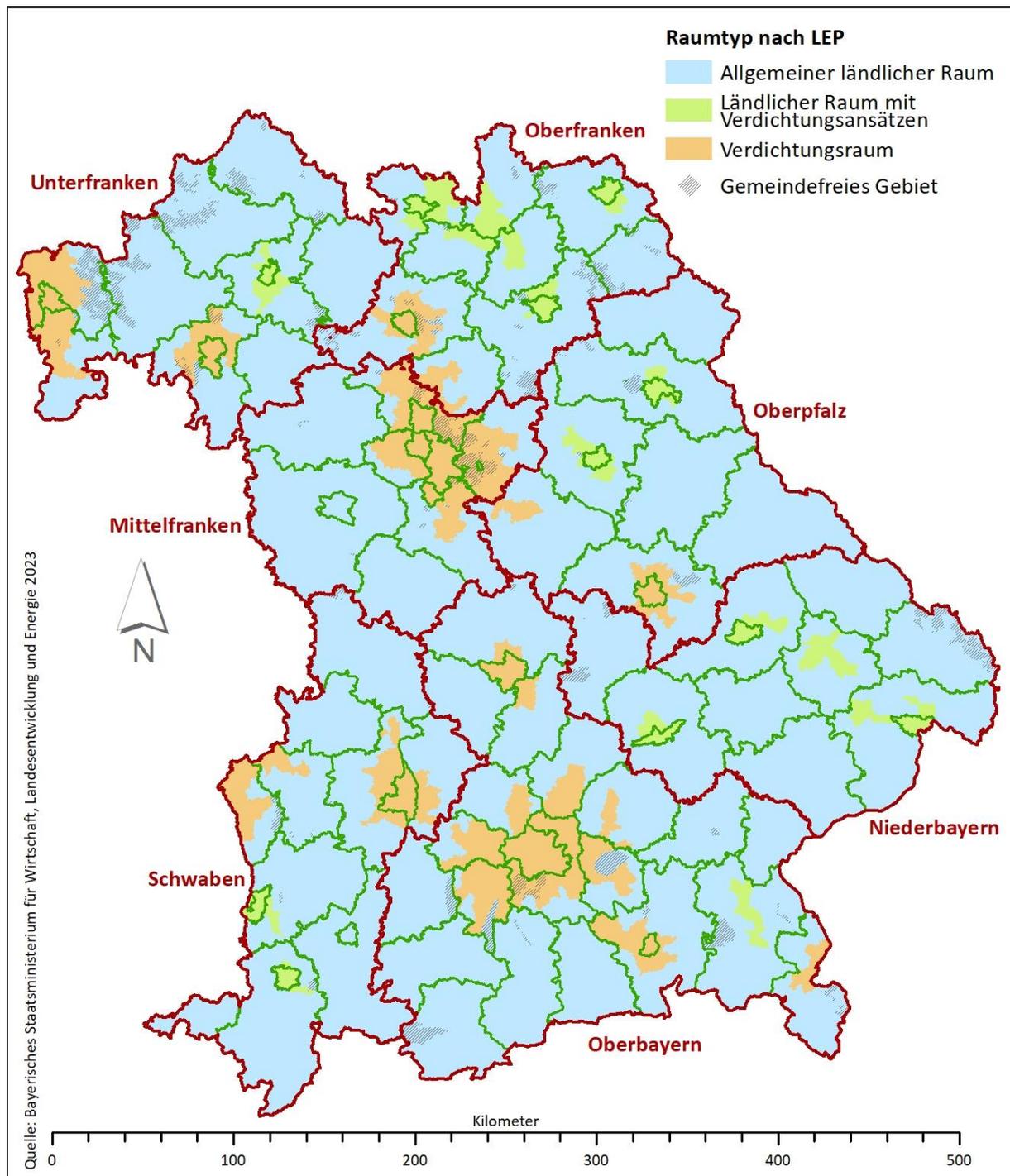


Abb. 59 Raumtypen nach bayrischem Landesentwicklungsprogramm