

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Einsatz von GeoAI zur Optimierung Urbaner Digitaler Zwillinge:
Semantische Anreicherung und automatisierte Klassifikation
von Gebäuden in urbanen 3D-Modellen

verfasst von | submitted by
Elias Kriegl BSc

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of
Master of Science (MSc)

Wien | Vienna, 2026

Studienkennzahl lt. Studienblatt | Degree
programme code as it appears on the
student record sheet:

UA 066 856

Studienrichtung lt. Studienblatt | Degree
programme as it appears on the student
record sheet:

Masterstudium Kartographie und Geoinformation

Betreut von | Supervisor:

Ass.-Prof. Mag. Dr. Andreas Riedl

Kurzfassung

Diese Arbeit widmet sich der Synthese von Urbanen Digitalen Zwillingen (UDZ) und GeoAI (künstliche Intelligenz im Bereich der Geoinformation). Angesichts der fortschreitenden Urbanisierung und der damit einhergehenden Herausforderungen für Städte und urbane Räume gewinnen UDZ zunehmend an Bedeutung, da sie das Potenzial bieten, die enormen Datenmengen einer Stadt zu managen, zu analysieren und anschaulich zu visualisieren. Nach einer präzisen Begriffsdefinition und einer Analyse des aktuellen Forschungsstands werden zunächst die Unterschiede zwischen UDZ und verwandten Systemen herausgearbeitet. Anschließend werden UDZ in vier Phasen unterteilt: (Datenakquise, Datenverarbeitung, Simulation & Analyse sowie Visualisierung) und entlang dieser Phasen genau untersucht. In diesen Phasen werden jeweils Chancen und Herausforderungen identifiziert. In der Synthese werden die Stärken der GeoAI mit diesen Herausforderungen und Chancen in Verbindung gebracht.

Im Mittelpunkt der Arbeit steht ein praxisorientierter Ansatz, um UDZ mithilfe von Machine Learning (ML) semantisch anzureichern. Am Beispiel der oberösterreichischen Landeshauptstadt Linz werden Gebäude automatisiert einer von fünf Gebäudekategorien zugeordnet. Grundlage hierfür bildet der in Europa etablierte INSPIRE-Standard. Um das optimale ML-Modell zu bestimmen, wurden sowohl verschiedene Attributkombinationen als auch unterschiedliche Modellvarianten und Parametereinstellungen systematisch getestet. Mit der finalen Kombination aus mehreren ML-Modellen und der optimalen Attributauswahl wird eine Genauigkeit von fast 90 % erreicht.

Die Ergebnisse zeigen, dass eine KI-gestützte, semantische Anreicherung von Gebäudedaten mit vertretbarem Aufwand und auf Basis allgemein verfügbarer Geodaten realisierbar ist. Die entwickelte Methode ist flexibel, lässt sich problemlos wiederholen und erweitern und kann unter bestimmten Voraussetzungen auch auf andere Städte übertragen werden. Die Überführung der Ergebnisse ins CityGML-Format ermöglicht zudem die Integration in gängige 3D-Viewer und urbane Datenplattformen.

Abstract

This thesis is dedicated to the synthesis of Urban Digital Twins (UDT) and GeoAI (artificial intelligence in the field of geoinformation). In light of ongoing urbanization and the associated challenges for cities and urban areas, UDTs are gaining increasing importance, as they offer the potential to manage, analyze, and vividly visualize the enormous amounts of data generated by a city. After a precise definition of terms and an analysis of the current state of research, the differences between UDTs and related systems are first elaborated. Subsequently, UDTs are divided into four phases (data acquisition, data processing, simulation & analysis, and visualization) and these phases are examined in detail. For each phase, specific opportunities and challenges are identified. In the synthesis, the strengths of GeoAI are linked to these challenges and opportunities.

The core of the thesis is a practice-oriented approach to semantically enrich UDTs using machine learning (ML). Using the example of the Upper Austrian capital Linz, buildings are automatically assigned to one of five building categories. The basis for this is the INSPIRE standard, which is widely used in Europe. To determine the optimal ML model, various attribute combinations as well as different model variants and parameter settings were systematically tested. With the final combination of several ML models and the optimal selection of attributes, an accuracy of almost 90% is achieved.

The results show that AI-supported semantic enrichment of building data is feasible with reasonable effort and based on generally available geodata. The developed method is flexible, can be easily repeated and extended, and, under certain conditions, can also be transferred to other cities. Converting the results into the CityGML format also enables integration into common 3D viewers and urban data platforms.

Inhalt

Kurzfassung	1
Abstract	2
1 Einleitung	7
1.1 Motivation und Zielsetzung	7
1.2 Stand der Forschung	7
1.3 Problemstellung und Relevanz der Arbeit.....	9
1.4 Wissenschaftliche Fragestellungen	9
1.5 Methodischer Ansatz und Vorgehensweise	10
1.6 Aufbau der Arbeit	10
2 Digitaler Zwilling (DZ)	12
2.1 Definition und Konzept	12
2.1.1 Historie	14
2.1.2 Digitale Repräsentationsstufen.....	15
2.2 Abgrenzung zu verwandten Systemen.....	16
2.2.1 Geografisches-Informations-System (GIS).....	16
2.2.2 Building Information Model (BIM)	16
2.2.3 BIM + GIS	17
2.2.4 City Information Model (CIM)	18
2.2.5 3D-Modell	19
2.2.6 3D-Modell + BIM	19
2.2.7 Zusammenfassung	19
2.3 Digitaler Zwilling im urbanen Kontext.....	21
2.3.1 Smart Cities	21
2.3.2 Urbaner Digitaler Zwilling	22
2.3.3 Hierarchie von Urbanen Digitalen Zwillingen	22
3 Urbaner Digitaler Zwilling (UDZ)	23
3.1 Datenquellen.....	25
3.1.1 Satellitendaten	26
3.1.2 Sensoren und IoT	26
3.1.3 Crowdsourcing, mobile Geräte, Social Media	28
3.1.4 Punktwolken	28

3.1.5 Probleme und Herausforderungen	29
3.2 Datenverarbeitung und –management.....	30
3.2.1 Big Data	30
3.2.2 Datenharmonisierung	31
3.2.3 Datenschutz & Sicherheit	31
3.2.4 Berechnungsmodelle	33
3.2.5 Datenspeicherung	34
3.2.6 Echtzeitfähigkeit und Synchronisation.....	35
3.2.7 Probleme und Herausforderungen	35
3.3 Simulation & Analyse.....	36
3.3.1 Prognosemodelle.....	37
3.3.2 Szenarien & Warnsysteme	37
3.3.3 Decision-Support.....	38
3.3.4 Probleme und Herausforderungen	38
3.4 Visualisierung	39
3.4.1 3D-Stadtmodelle	39
3.4.2 4D-Modelle (zeitbasierte Simulation)	45
3.4.3 Erreichbarkeit.....	45
3.4.4 Austausch- und Streamingformate	45
3.4.5 Cloudbasierte Plattformen	47
3.4.6 Virtuelle Realität	48
3.4.7 Probleme und Herausforderungen	48
3.5 Baustein-Konzept.....	49
3.5.1 Transport & Mobilität.....	50
3.5.2 Klimawandel	51
3.5.3 Verschmutzung	53
3.5.4 Energiemanagement.....	53
4 GeoAI	58
4.1 Definition	58
4.2 Historie	60
4.3 Methoden	62
4.3.1 Machine Learning (ML)	62

4.3.2 Deep Learning	63
4.3.3 Räumliche Grundmodelle (Prithvi)	66
4.3.4 XAI-Modelle.....	67
4.4 Einsatzgebiete	67
4.4.1 Nationale Kartenwerke/Datahubs	68
4.4.2 urbane Planung	68
4.4.3 Semantische Anreicherung.....	69
4.5 Zukunft von GeoAI	70
5 Synthese.....	71
5.1 Datenquellen.....	71
5.2 Datenverarbeitung und –management.....	73
5.3 Simulation & Analyse.....	75
5.4 Visualisierung	76
5.5 Fazit	78
6 UDZ in verschiedenen Städten	79
6.1 Digitaler GeoZwilling Wien	79
6.2 Digital Twin Zürich	79
6.3 Virtual Singapore.....	80
6.4 Kalasatama Helsinki.....	81
6.5 Herrenberg	81
6.6 Probleme und Herausforderungen	82
7 Der Geozwilling der Stadt Linz	83
7.1 Status quo	83
7.2 Bausteine	84
7.3 “Intelligente” Gebäude.....	85
8 KI-gestützte Klassifizierung von Gebäuden	86
8.1 Verwandte Theorien	86
8.2 Methodik	87
8.3 Datengrundlage	88
8.3.1 Gebäude	88
8.3.2 AGWR	89
8.3.3 True Orthofoto	90

8.4 Vorverarbeitung der Daten	90
8.4.1 Art der Kategorisierung	90
8.4.2 Händische Klassifizierung	91
8.4.3 Datensätze	92
8.4.4 Anreicherung mit zusätzlichen Attributen	93
8.5 Modelltraining und Klassifikation	93
8.5.1 Modell-Evaluierung	93
8.5.2 Attribute analysieren	99
8.5.3 Modelle kombinieren	105
8.5.4 Attribute eliminieren	105
8.6 Bewertung der Ergebnisse	110
8.6.1 Confusion-Matrix	110
8.6.2 Ergebnis für ganz Linz	111
8.7 Umwandlung in CityGML	113
8.8 Fortführung & Umsetzung in anderen Städten	116
8.9 Fazit	117
9 Zusammenfassung	119
Abkürzungsverzeichnis	121
Tabellenverzeichnis	123
Abbildungsverzeichnis	124
Hilfsmittelverzeichnis	126
Literaturverzeichnis	127
Anhang	145

1 Einleitung

1.1 Motivation und Zielsetzung

Ziel der Arbeit ist das Bilden einer Synthese von Urbanen Digitalen Zwillingen (UDZ) und künstlicher Intelligenz im Bereich der Geoinformation (GeoAI). Dadurch soll herausgefunden werden, wie GeoAI helfen kann, aktuelle Probleme im Zusammenhang mit UDZ zu lösen. Nach einer allgemeinen Begriffsdefinition sollen aktuelle Methoden und Probleme bei der Erstellung und Fortführung eines UDZ herausgearbeitet werden. Um einen besseren Eindruck davon zu bekommen, werden bestehende urbane Projekte zum Thema UDZ untersucht.

Anschließend wird der Fokus auf GeoAI gelegt. Auch hier wird zuerst der aktuelle Stand der Forschung festgestellt. Danach werden einzelne GeoAI-Methoden analysiert, die möglicherweise beim Thema UDZ helfen können. Es folgt eine Synthese der beiden aktuellen Themen. Dabei wird genauer auf den möglichen Einsatz von GeoAI-Methoden zur Erstellung oder Verbesserung von UDZ eingegangen.

Der praktische Teil zielt darauf ab, einen UDZ mit Hilfe von Machine Learning (ML) semantisch anzureichern. Dies geschieht am Beispiel des UDZ der oberösterreichischen Stadt Linz. Es werden Gebäude aus dem Linzer Modell mit Attributen aus anderen Datensätzen verknüpft und mittels ML automatisch klassifiziert. Ziel ist es herauszufinden welche ML-Modelle und Einstellungen und welche Attribute sich gut und welche weniger gut für die automatische Klassifizierung eignen. Die Klassifizierung erfolgt nach den INSPIRE-Gebäudeklassen, die vor allem in Europa sehr häufig eingesetzt werden.

1.2 Stand der Forschung

Um den Stand der Forschung bestmöglich zusammenzufassen, sollten die beiden Begriffe UDZ und GeoAI zunächst isoliert voneinander betrachtet werden. Wichtig anzumerken ist der signifikante Unterschied zwischen einem gängigen 3D-Modell und einem UDZ. Während ein normales 3D-Modell die Stadt zu einem bestimmten Zeitpunkt visuell darstellt, wird ein UDZ laufend mit Daten gefüttert und ermöglicht Analysen und Simulationen. Diese Simulationen können in der Vergangenheit, Gegenwart oder Zukunft erfolgen. Weiters weisen 3D-Stadtmodelle im Gegensatz zu UDZ in vielen Fällen kaum attributive Daten für die Objekte darin auf. In einigen Fällen lassen sich einzelne Objekte gar nicht identifizieren, da es sich um ein Mesh-Modell handelt. (Shahat et al. 2021) Die spezifischen Unterschiede zwischen 3D-Modellen, UDZ und weiteren verwandten Systemen werden in Kapitel 2.2 detailliert erläutert.

Ein fertiger UDZ ist ein Spiegelbild der Realität. Eine Änderung in der realen Welt wirkt sich automatisch auf den UDZ aus. Umgekehrt können mit dem UDZ Veränderungen in der Realität bewirkt werden. Allerdings sind aktuelle Projekte noch sehr weit von diesem Stadium entfernt. Aktuelle UDZ befinden sich in ihrer Entwicklung näher an einem

einfachen 3D-Modell als an dem eben angeführten Spiegelbild der Realität. Kritzinger et al. (2018) haben die Entwicklung von Digitalen Zwillingen (DZ) in drei Stufen unterteilt: digitales Modell, digitaler Schatten und digitaler Zwilling. Ein digitales Modell ist eine digitale Repräsentation eines geplanten oder bereits existierenden Objekts. Zwischen dem Modell und dem Objekt besteht keinerlei automatischer Datenaustausch. Alle Änderungen werden somit manuell durchgeführt. Ein digitaler Schatten weist einen einseitigen, automatischen Datenaustausch auf. So kann beispielsweise eine Änderung am realen Objekt automatisch eine Anpassung des digitalen Objekts herbeiführen. Die dritte Stufe (Digitaler Zwilling) beschreibt einen beidseitigen automatischen Datenaustausch. (Kritzinger et al. 2018)

Ein großer Vorteil von UDZ ist das Herstellen von Verbindungen zwischen vergangenen, aktuellen und zukünftigen Daten. Je mehr Daten in einem UDZ enthalten sind desto genauer und umfangreicher können Simulationen durchgeführt werden. Diese Simulationen und Analysen können in fast allen Bereichen einer Stadt hilfreich sein. Allerdings gibt es noch sehr viele Hürden und Probleme bei der Umsetzung eines umfangreichen UDZ. Zu den größten Problemen zählen die Datenharmonisierung, die Datenmenge und die Anreicherung von Modellen mit semantischen Inhalten. (Shahat et al. 2021) Diese und weitere Herausforderungen werden in den Kapiteln 3.1.5, 3.2.7, 3.3.4 und 3.4.7 näher behandelt.

Bereits vor über 30 Jahren wurden Überlegungen angestellt, wie künstliche Intelligenz für geographische Modelle und Fragestellungen eingesetzt werden kann. (Scheider und Richter 2023) Mit dem rasanten technologischen Fortschritt im Bereich der künstlichen Intelligenz (KI), gewinnt auch GeoAI an Bedeutung. GeoAI ergänzt bestehende KI-Modelle um geographische Beziehungen. Dadurch lassen sich zum Beispiel Objekte auf Luftbildern automatisch erkennen. Diese automatisierte Erkennung wird immer besser, da die KI aus "Fehlern" lernt. Eine weitere Stärke von GeoAI ist die Integration von heterogenen Daten. Dadurch können größere Datenmengen rascher gemeinsam analysiert werden. Außerdem ermöglicht GeoAI Simulationen auf Basis von vergangenen und aktuellen Daten durchzuführen. Die Methoden im Bereich der GeoAI entwickeln sich dynamisch weiter, und regelmäßig entstehen neue Ansätze und Werkzeuge. Trotz dieser vielversprechenden Möglichkeiten dürfen die Risiken und Herausforderungen, die mit dem Einsatz von GeoAI einhergehen, nicht außer Acht gelassen werden. Ein bewusster und reflektierter Umgang mit den Technologien ist daher unerlässlich. (Li et al. 2024b)

Natürlich können auch UDZ stark von den aktuellen Entwicklungen im Bereich der GeoAI profitieren. Wissenschaftliche Publikationen gibt es aber noch wenige in diesem Zusammenhang. Eine der wenigen ist die Arbeit von Hu et al. (Hu et al. 2023). Es gibt einige wissenschaftliche Arbeiten, die die Relevanz von künstlicher Intelligenz für Smart Cities beschreiben. (Mortaheb und Jankowski 2023) (Alastal und Shaqfa 2022) Die Synthese mit dem Thema UDZ bleibt aber meistens aus. Deshalb ist eines der Ziele

dieser Masterarbeit das Erstellen einer solchen Synthese zwischen diesen beiden aktuellen Themen der Geoinformation.

1.3 Problemstellung und Relevanz der Arbeit

55 % der Weltbevölkerung leben in Städten und Wissenschaftler gehen davon aus, dass 2050 bereits 70% im urbanen Raum leben. Außerdem wird sich die städtische Fläche um 1,2 Millionen Quadratkilometer vergrößern. Heutzutage verbrauchen Städte zwei Drittel des gesamten Energiekonsums und sind für 70 % des weltweiten Ausstoßes von Treibhausgasen verantwortlich. Deshalb ist es unabdingbar, Städte effizienter zu machen. Sogenannte Smart Cities verknüpfen Objekte und Prozesse innerhalb der Stadt miteinander und sorgen dafür, dass Ressourcen minimiert und besser aufgeteilt werden. (Xia et al. 2022) Der Begriff UDZ taucht häufig im Zusammenhang mit Smart Cities auf. Gerade im Kontext der Nachhaltigkeit ist es wichtig, dass Städte effektiv verwaltet werden. Dies geschieht durch eine Vielzahl an Informationen, die aktuell abgefragt werden können und zur Steuerung und Planung auf Entscheidungsebene beitragen. UDZ sind dafür die perfekte Lösung. Sie können mit Sensoren und verschiedenen Datenquellen verbunden werden und visualisieren die Daten für Entscheidungsträger ansprechend. Außerdem kann eine Vielzahl an Analysen und Simulationen durchgeführt werden. So können beispielsweise die Auswirkungen einer städtischen Baumaßnahme vorab simuliert werden. (Weil et al. 2023)

GeoAI hat das Potenzial aktuelle Probleme im Zusammenhang mit UDZ in vielen Bereichen zu lösen. Eine Synthese der beiden wichtigen Themen wird Bereiche und Methoden hervorheben, die sich für zukünftige Forschungen und Analysen anbieten. Dadurch kann es gelingen, Städte noch “smarter” und nachhaltiger zu machen. Der Schwerpunkt dieser Synthese wird darauf liegen, die aktuellen Schwächen von UDZ mit Stärken von GeoAI zu verbinden.

1.4 Wissenschaftliche Fragestellungen

Hauptfrage:

Wie kann GeoAI zur semantischen Anreicherung von 3D-Stadtmodellen und zur automatischen Gebäudeklassifizierung beitragen, um Urbane Digitale Zwillinge zu verbessern?

Unterfragen:

1. Welche aktuellen Herausforderungen gibt es bei der Erstellung und Verwaltung von Urbanen Digitalen Zwillingen, und wie kann GeoAI helfen diese zu bewältigen?
2. Welche Datenquellen eignen sich für die semantische Anreicherung von 3D-Stadtmodellen?
3. Welche KI-Methoden und Modelle sind geeignet, um 3D-Stadtmodelle semantisch anzureichern?

1.5 Methodischer Ansatz und Vorgehensweise

Um diese Fragen zu beantworten, wird ein methodischer Ansatz gewählt, der eine umfassende Literaturrecherche mit einem praktischen Anwendungsbeispiel kombiniert. Zunächst wird der Begriff DZ erklärt und von verwandten Systemen abgegrenzt. Anschließend werden die Themen UDZ und GeoAI genau analysiert. Ziel ist es, den aktuellen Stand der Forschung zu erfassen und gemeinsame Probleme auszuarbeiten. Hierfür werden UDZ in vier Phasen eingeteilt (Datenquellen, Datenverarbeitung, Simulation & Analyse und Visualisierung). Im nächsten Schritt erfolgt die Synthese der beiden Themen. Es werden systematisch die Herausforderungen und Schnittstellen in den einzelnen Phasen ausgearbeitet. So wird herausgefunden, wie GeoAI potenziell bestehende Probleme von UDZ lösen kann. In diesem Schritt werden viele Forschungslücken sichtbar, die eine vertiefende Analyse nahelegen. Eine dieser Forschungslücken wird im praktischen Teil behandelt.

Der praktische Teil der Arbeit konzentriert sich auf die semantische Anreicherung von Gebäudedaten. Hierfür wird ein ML-Workflow entwickelt, der Gebäude nach INSPIRE-Gebäudeklassen klassifiziert. Davor werden die Gebäude mit Attributen aus verschiedenen Quellen (Adressregister, Orthofotos, ...) angereichert. Um das optimale Modell zu finden, werden verschiedene ML-Modelle miteinander verglichen und kombiniert und Parameter auf ihre Bedeutung analysiert. Die Performances der Modelle werden mittels Genauigkeitswerten und F1-Scores verglichen. Nach Abschluss des Trainings werden sämtliche Gebäude in der Stadt Linz mit dem ML-Modell klassifiziert. Das Ergebnis wird stichprobenartig überprüft, um typische Fehler und Muster in den Daten zu erkennen. Abschließend werden die Ergebnisdaten ins CityGML-Format überführt, um visualisiert werden zu können. Weiters wird diskutiert, wie die Umsetzung des Workflows in anderen Städten erfolgen kann.

Die beiden zentralen Schwerpunkte des methodischen Workflows liegen einerseits in der systematischen Verknüpfung und Analyse der Themen UDZ und GeoAI und andererseits in der praktischen Umsetzung einer semantischen Anreicherung und automatisierten Klassifizierung von Gebäudedaten.

1.6 Aufbau der Arbeit

Die Gliederung der Arbeit folgt der methodischen Vorgehensweise und führt schrittweise von der Literaturanalyse bis zum praktischen Teil der Arbeit. Kapitel 1 erläutert die Motivation und Zielsetzung dieser Arbeit. Kapitel 2 widmet sich dem Konzept des DZ und grenzt diesen Begriff gegenüber verwandten Konzepten ab. Dazu zählen beispielsweise 3D-Modelle oder Building-Information-Modelle (BIM). Im dritten Kapitel stehen UDZ im Mittelpunkt. Die Erstellung eines UDZ wird in vier Phasen (3.1 Datenquellen, 3.2 Datenverarbeitung und -management, 3.3 Simulation & Analyse, 3.4 Visualisierung) gegliedert. Jede dieser Phasen wird genau analysiert und es werden Herausforderungen für jede Phase ausgearbeitet. Kapitel 4 erklärt Begriffe und Methoden der GeoAI. In

Kapitel 5 erfolgt die Synthese der Kapitel 3 und 4. Hier werden Überschneidungen und Forschungslücken ausgearbeitet. Kapitel 6 bietet einen Überblick über aktuelle Projekte zum Thema UDZ. Es werden fünf verschiedene UDZ behandelt. Kapitel 7 erklärt den Aufbau des UDZ der Stadt Linz, der im praktischen Teil erneut behandelt wird. Der praktische Teil wird in Kapitel 8 beschrieben und bildet den Schwerpunkt dieser Arbeit. In diesem wird die Datenaufbereitung, der Workflow, die Ergebnisse und die Übertragbarkeit auf andere Städte detailliert beschrieben. Im abschließenden Kapitel 9 erfolgt eine Zusammenfassung der wichtigsten Erkenntnisse dieser Arbeit. Außerdem wird ein Ausblick gegeben und offene Fragestellungen werden angeführt. Mit diesem Aufbau bietet die Arbeit neben einem umfassenden theoretischen Überblick auch einen praxisnahen Einblick in die Anwendung von GeoAI im Zusammenhang mit UDZ.

2 Digitaler Zwilling (DZ)

2.1 Definition und Konzept

Drei primäre Charakteristika von DZ streichen Wang et al. (Wang et al. 2023b) in ihrer Arbeit hervor. Erstens die Verbindung von virtueller und realer Welt, zweitens, die Aktualisierung des virtuellen Modells bei Änderungen in der Realität und drittens, die Fähigkeit der Vorhersage und Simulation. Allerdings gibt es sehr viele weitere Definitionen. Barricelli et al. (2019) stellen in ihrem Paper mehrere Definitionen gegenüber (Abbildung 2 & 3). Weiters leiten sie aus der Vielzahl an Definitionen eine eigene Beschreibung ab. Diese wird im folgenden Absatz zusammengefasst.

DZ können als physikalische oder virtuelle Maschinen oder Modelle definiert werden, die das Leben einer physischen Entität (Objekt, Prozess, Mensch, Stadt etc.) simulieren oder spiegeln. Jeder DZ ist eindeutig mit seinem physischen Zwilling verknüpft. Ein DZ geht über ein normales Modell oder eine normale Simulation hinaus. Er ist der virtuelle Gegenspieler seines physischen Zwillings und überwacht, kontrolliert und optimiert dessen Verhalten. Der DZ kann Probleme voraussagen und den Einfluss verschiedener Parameter simulieren. Dies geschieht dank eines stetigen Informationsflusses zwischen den beiden Zwillingen. (Barricelli et al. 2019) Auch Alvi et al. (2025) haben verschiedene Definitionen in einer Tabelle zusammengefasst (Tabelle 1).

Author	Area	Definition	Ref.
Framling et al.	Product Mgmt.	"Agent-based architecture with each product having a virtual counterpart." (2003)	Hribernik, Cabri, Mandreoli, and Mentzas (2021)
NASA	Aviation	"Improves lifecycle prediction and management of structures." (2011)	Liu, Fang, Dong, and Xu (2021)
NASA	Aviation	"Integration of records, data, and simulations for safety and dependability." (2012)	Wang et al. (2022)
Hochhalter	Aerospace	"Life management paradigm using vehicle-specific history for high-fidelity modeling." (2014)	Hochhalter et al. (2014)
National instruments	HVAC	"Real-time digital replica of a physical device." (2017)	Bacchiega (2017)
Grieves	PLM	"Virtual data structure fully describing a product from micro to macro level." (Basic Idea 2003, Pub. 2014 and 2017)	Grieves and Vickers (2017)
El Saddik	Multimedia	"Digital replica of physical entities, bridging physical and virtual worlds." (2018)	El Saddik (2018)
Bruynseels et al.	Health care	"Engineering paradigm where physical artifacts are paired with digital models." (2018)	Bruynseels, Santoni de Sio, and Van den Hoven (2018)
Leng et al.	MCPS	"Real-time cyber copy of a physical manufacturing system." (2019)	Leng et al. (2019)
Luo et al.	CNCMT	"Multi-domain, high fidelity digital model integrating various subjects." (2019)	Luo, Hu, Zhang, and Wei (2019)
Nikolakis et al.	Factory area	"Digital representation of real-world objects/processes, enhanced by sensor data." (2019)	Nikolakis, Alexopoulos, Xanthakis, and Chrysosouris (2019)
Erkoyuncu et al.	Robotics	"Digital representation of a physical asset for property, condition, and behavior analysis." (2020)	Erkoyuncu et al. (2020)
Zheng et al.	CPS	"Provides deep integration with control systems for direct physical system control." (2020)	Zheng and Sivabalan (2020)
Latif and Starly	Industrial	"Digital representation of physical entities/processes to optimize performance." (2020)	Latif and Starly (2020)
Minerva et al.	IoT	"Clones physical objects into software counterparts, reflecting essential properties." (2020)	Minerva, Lee, and Crespi (2020)
Autodesk	IoT services	"Living model with dynamics/functionality, for analytics and predictive maintenance." (2021)	Nath, Van Schalkwyk, and Isaacs (2021)
Mylonas et al.	Smart cities	"Virtual representation simulating behavior of physical things throughout lifecycle." (2021)	Mylonas et al. (2021)

Abbildung 1: Digital Twin Definitionen (Alvi et al. 2025)

Key points	Definitions	References	Ties, links	Connections of data and information that ties the virtual and the real product together.	[19]
Integrated system	Integrated multi-physics, multiscale, and probabilistic simulation composed of physical product, virtual product, data, services and connections between them.	[16], [57], [58]		New mechanisms to manage IoT devices and IoT systems-of-systems.	[11]
	An ultra-realistic integrated multi-physics, multiscale, probabilistic simulation of a system.	[22]		Technology that links the real and the digital worlds.	[65]
	A big collection of digital artifacts that has a structure, all elements are connected and there exists meta-information as well as semantics.	[53]	Description, construct, information	Comprehensive physical and functional description of a component, product or system.	[45]
	Comprehensive physical and functional description of a component, product or system together with all available operational data.	[55]		A digital information construct about a physical system.	[24]
	A systematic approach consisting of sensing, storage, synchronization, synthesis and service.	[60]		The notion where the data of each stage of a product lifecycle is transformed into information.	[62]
Clone, counterpart	Computerized clones of physical assets.	[15]	Simulation, test, prediction	A safe environment in which you can test the impact of potential change on the performance of a system.	[51]
	The virtual and computerized counterpart of a physical system.	[46]		Reengineering computational model of structural life prediction and management.	[20]
	Functional system formed by the cooperation of physical production lines with a digital copy.	[33]		A simulation based on expert knowledge and real data collected from the existing system.	[54]
				Virtual models for physical objects to simulate their behaviors.	[56]

Abbildung 2: Digital Twin Definitionen (Barricelli et al. 2019)

Virtual, mirror, replica	A virtual representation of the system.	[27]
	Digital mirror of the physical world.	[61]
	Digital model that dynamically reflects the status of an artifact.	[63]
	Detailed virtual model of ourselves. ^a	[64]
	Virtual representation of a real product.	[47]
	A digital copy of a physical system.	[66]
	Virtual model of a physical asset.	[67]
	A replication of real physical production system.	[59]
	Cyber copy of a physical system.	[48]
	A dynamic digital representation of a physical system.	[52]
	A virtual model of physical object.	[34]

The definitions reported in this table are the result of the authors' re-elaboration and are not to be intended as precise quotations from the original sources.

^a This definition refers to Human Digital Twins.

Abbildung 3: Digital Twin Definitionen (Barricelli et al. 2019)

2.1.1 Historie

2003 präsentierte Dr. Michael Grieves das Konzept des DZ im Rahmen einer Diskussion zum Thema Produktlebenszyklus-Management (Grieves 2015). Das Ziel seines Modells war eine Verknüpfung der realen und digitalen Welt. Im DZ soll ein kontinuierlicher Austausch von Daten und Informationen zwischen Realität und digitalem Modell erfolgen. Der Begriff gewann an Bedeutung, als die NASA ihn 2011 aufgriff und auf Produkte in der Raumfahrt anwendete. Wie bei der ursprünglichen Präsentation von Dr. Michael Grieves lag auch bei der Raumfahrtbehörde der Fokus auf der Überwachung des Produktlebenszyklus. Doch mit der Zeit wurde der Begriff weitläufiger verwendet. (Alvi et al. 2025) Bereits 2012 beschrieb die NASA den DZ als eine nahtlose Integration von Wartungsaufzeichnungen, Flotten- und historischen Daten sowie hochpräzisen Simulationen mit einem integrierten Fahrzeug-Gesundheitsmanagement-System. (Lee et al. 2024) Auch Dr. Michael Grieves ergänzte seine Definition und sprach von einem vollständigen Set virtueller Datenstrukturen, die ein physisch hergestelltes Produkt von der atomaren Mikroebene bis zur geometrischen Makroebene vollständig beschreiben. (Wang et al. 2022b)

Mittlerweile gibt es viele verschiedene Definitionen und Anwendungsgebiete von DZ (Abbildungen 1-3). Alvi et al. (2025) haben ein Diagramm erstellt, das die Häufigkeit von wissenschaftlichen Artikeln und Google-Suchen zu den Themen Digital Twin und Smart Cities darstellt (Abbildung 4). Diese Abbildung zeigt, dass das Interesse am Thema „Digital Twin“ in den letzten Jahren stark gestiegen ist. Während es 2019 noch kaum Artikel zu diesem Thema gab, wurden 2024 mehr als 150 Artikel veröffentlicht.

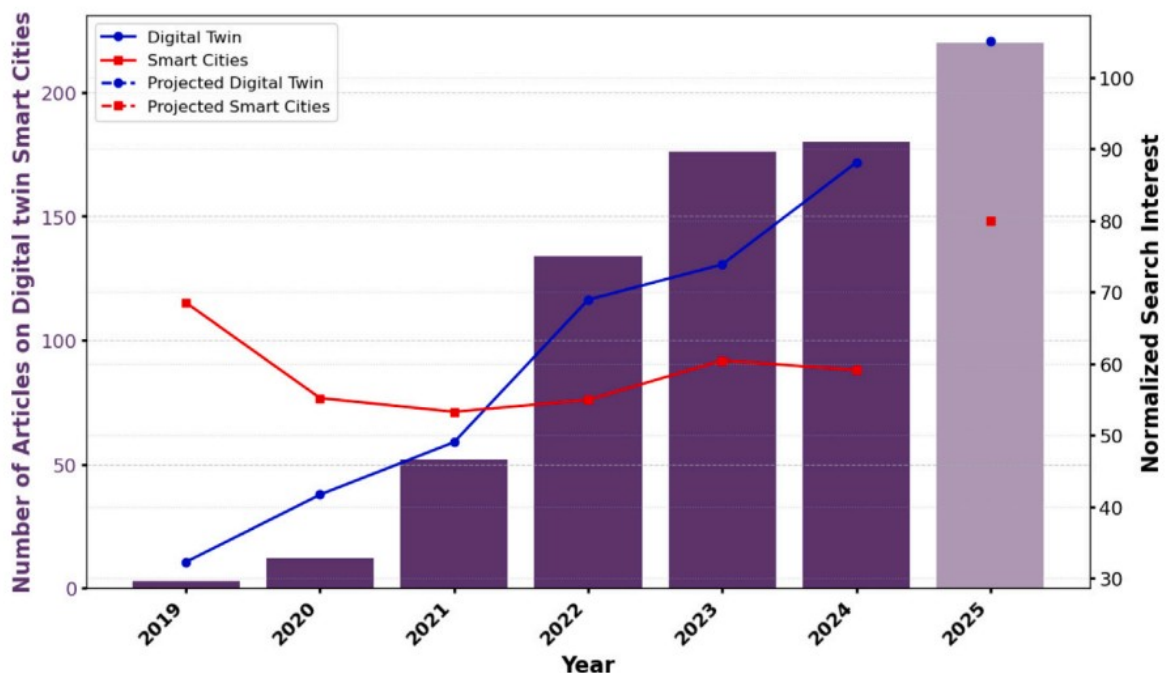


Abbildung 4: Wissenschaftliche Artikel und Google Search Trends (2019-2025) (Alvi et al. 2025)

2.1.2 Digitale Repräsentationsstufen

Digitale Zwillinge tauchten zuerst in der Industrie auf. In der Produktion beschäftigten sich Wissenschaftler mit diesem Begriff. Kritzinger et al. (2018) unterteilten DZ nach ihrem Integrationslevel in drei Kategorien. Die erste Stufe sind **Digitale Modelle** (Abbildung 5 links). Ein Digitales Modell ist eine Repräsentation eines bereits existierenden oder geplanten Objekts. Zwischen dem physischen Objekt und dem digitalen Objekt gibt es keinerlei automatischen Datenfluss. Solche Modelle können zwar in der Lage sein, Simulationen und Analysen durchzuführen, doch der Datenaustausch wird stets manuell durchgeführt. Es folgen **Digitale Schatten** (Abbildung 5 rechts), die im Gegensatz zu Digitalen Modellen einen einseitigen automatischen Datenfluss aufweisen. Meist handelt es sich dabei um Daten, die automatisch vom physikalischen Objekt ins digitale Objekt fließen. Die finale Stufe ist der **DZ** (Abbildung 6). Für Kritzinger et al. (2018) muss ein beidseitiger automatischer Datenfluss bestehen, um von einem DZ sprechen zu können. (Kritzinger et al. 2018)

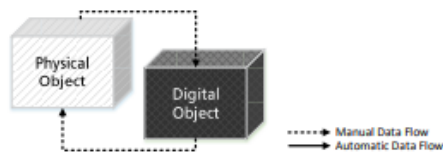


Fig. 1. Data Flow in a Digital Model

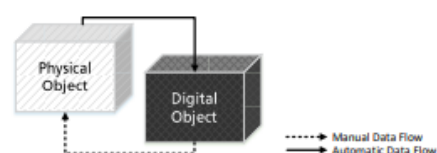


Fig. 2. Data Flow in a Digital Shadow

Abbildung 5: Datenfluss im Digitalen Modell und im Digitalen Schatten (Kritzinger et al. 2018)

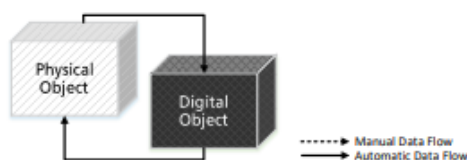


Fig. 3. Data Flow in a Digital Twin

Abbildung 6: Datenfluss im Digitalen Zwilling (Kritzinger et al. 2018)

Es gibt also DZ, die näher an einem tatsächlichen Zwilling sind als andere DZ. DZ können von reinen Gedankenexperimenten bis zu Modellen reichen, die versuchen so viele Aspekte der Realität wie möglich zu spiegeln, reichen. Denkt man jedoch an das Endstadium eines DZ, also an einen Zwilling der Realität, stellt sich die Frage, ob so ein DZ überhaupt realisierbar ist. Man kann sogar so weit gehen, zu sagen, dass sich ein voll entwickelter Zwilling nicht mehr vom Originalsystem unterscheidet und somit selbst dieses System ist. Dieser Gedanke mag zunächst abstrakt wirken. Doch bei genauerer Betrachtung fällt auf, dass ein DZ, wenn er weit entwickelt ist, beginnt, immer mehr und mehr Verknüpfungen zur Realität einzugehen. Der DZ und das reale Modell vermischen sich und eine genaue Trennung ist nicht mehr möglich. Das heißt jedoch nicht, dass das

Konzept eines DZ unnötig ist. Es macht nur deutlich, dass manche DZ stärker mit der Realität verwoben sind als andere und eine klare Definition, ab wann man von einem “Zwilling” sprechen kann, nur sehr schwer zu finden ist. (Batty 2018)

2.2 Abgrenzung zu verwandten Systemen

2.2.1 Geografisches-Informationen-System (GIS)

GIS sammeln, speichern, verwalten, analysieren und präsentieren räumliche Daten. Die große Stärke eines GIS ist die exzellente Fähigkeit, räumliche Daten zu verarbeiten. Deshalb werden GIS in vielen verschiedenen Anwendungsbereichen eingesetzt. Dazu zählen beispielsweise die Regionalplanung, das Risikomanagement oder die Fortführung von Katastern. Außerdem können GIS Simulationen durchführen und somit bei Planungen und Problemen behilflich sein. (Wang et al. 2019a)

Im Vergleich zu Building-Information-Modellen (BIM) haben GIS einen großen Vorteil, wenn es um das Arbeiten mit großmaßstäbigen Daten geht. GIS analysieren und visualisieren räumliche und attributive Daten mittels räumlicher Analysetools und Methoden. Sie verfügen über eine robuste Datengrundlage und die notwendige Rechenleistung, um Daten aus mehreren Quellen effektiv zu verarbeiten. GIS eignen sich daher gut für großmaßstäbige, räumliche Analysen. (Xia et al. 2022)

Es gibt einige Unterschiede zwischen einem GIS und einem DZ. Ein GIS fokussiert sich auf räumliche Daten und stellt diese dar. Dabei handelt es sich fast immer um Raster- oder Vektordaten. Ein DZ hingegen kombiniert räumliche Daten und andere Informationen und präsentiert diese in einer Umgebung. Dies kann jedoch auch ohne geografische Informationen erfolgen. So wird beispielsweise für den DZ einer Maschine die genaue Lage des Objekts nicht benötigt. Dieses Beispiel zeigt auch den oft sehr großen Maßstab bei DZ. Ein GIS macht erst ab einem bestimmten Maßstab Sinn. Die Ziele eines GIS bestehen hauptsächlich in der räumlichen Analyse und der statischen Darstellung. DZ modellieren die Prozesse dynamisch und bestenfalls in Echtzeit. Außerdem werden Prognosen getätigt. Ein DZ ist also in der Lage, in die Zukunft zu blicken. GIS-Systeme stellen oft nur vergangene oder aktuelle Daten dar. (Xia et al. 2022)

Ein GIS kann vielmehr als Basis, beziehungsweise Grundlage für DZ gesehen werden. Dies trifft vor allem auf UDZ (siehe 2.3.2) zu. UDZ bauen oft auf einem GIS auf und erweitern das GIS um einige Funktionen. So werden beispielsweise präzise Gebäudedaten aus BIM-Modellen integriert und Daten von Sensoren in Echtzeit ergänzt. Mit diesen Daten können dann Simulationen durchgeführt werden, die in einem klassischen GIS nicht möglich sind. (DIN Media GmbH 2024)

2.2.2 Building Information Model (BIM)

Bei einem BIM kann man von einem weiterentwickelten 3D-Modell sprechen. BIM steht für „Building Information Model“, wobei vor allem die Information den Unterschied zum klassischen Gebäudemodell ausmacht. BIM helfen beim Design, der Errichtung und

Wartung von Projekten und können zu erheblichen Zeit- und Kostenersparnissen beitragen. (Chiến Phạm 2014) BIM können auch sehr gut in Kombination mit virtueller Realität (VR) eingesetzt werden. Anwendungen auf Stadtebene sind aber sehr selten, da es viele Probleme bei größeren BIM-Projekten gibt. BIM können zwar genauso wie DZ mit Sensoren verknüpft werden und so Live-Daten abfragen, sind jedoch nicht primär dafür gemacht. Der große Unterschied zwischen einem BIM und einem DZ liegt darin, was mit den Daten gemacht werden kann. In einem DZ können Analysen, Simulationen und Prognosen durchgeführt werden. Ein normales BIM ist selbst mit Verknüpfung zu Sensoren dazu nicht in der Lage. Außerdem sind BIM zu detailliert, um für größere Gebiete zu funktionieren. Die Kosten, die für eine Hardware, die mit so großen BIM umgehen kann, nötig wären, machen es unmöglich, ein BIM für eine ganze Stadt zu erstellen. (Shahat et al. 2021) (Xia et al. 2022) (Khajavi et al. 2019) BIM-Modelle sind also genau wie GIS-Systeme häufig ein Bestandteil von Udz (Abbildung 7). (DIN Media GmbH 2024; Dou et al. 2020; Azhar 2011; Zaker und Coloma 2018)

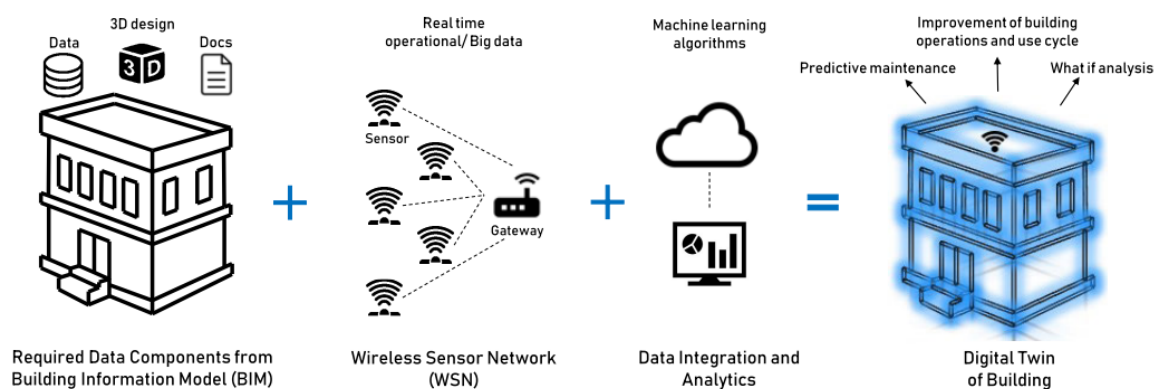


Abbildung 7: Komponenten eines Gebäudezwillings (Khajavi et al. 2019)

2.2.3 BIM + GIS

GIS und BIM können einander sehr gut komplementieren. Die Idee, diese beiden Modelle miteinander zu kombinieren, stammt aus dem Brandschutz. Ursprünglich wurde die Berechnung des optimalen Weges zum Gebäude für die Einsatzkräfte mit einem GIS durchgeführt. Denn ein wesentlicher Vorteil von GIS ist die Navigationsfunktion. Ein noch besseres Ergebnis kann jedoch erzielt werden, wenn auch Informationen aus dem Inneren des Gebäudes in die Berechnung integriert werden. Mit diesen zusätzlichen Informationen wissen die Einsatzkräfte, wo der beste Eingangspunkt ist und wie im Inneren des Gebäudes vorzugehen ist. Um die beiden Modelle miteinander zu kombinieren, wurden funktionale Integrationsmethoden angewandt. Dazu zählen die Standortauswahl, das Brandbekämpfungsmanagement, die Datenvisualisierung und die Entwicklung einer Benutzeroberfläche. Wichtig bei dieser Integration ist, dass nicht zu viele Daten aus dem BIM übernommen werden, da sonst die Dateigröße zu groß wird. Im Fall der Brandbekämpfung wurde die geometrische Hülle des Gebäudes und einige

semantische Informationen aus dem BIM ins GIS übernommen. Auch in vielen anderen Anwendungsgebieten findet man diese Kombination aus GIS und BIM. (Xia et al. 2022); Wang et al. 2019a) haben drei verschiedene Kombinationsmöglichkeiten von GIS und BIM und deren Vor- und Nachteile herausgearbeitet. Die Varianten unterscheiden sich je nachdem, welches System (BIM oder GIS) dominiert (Abbildung 8).

Advantages and disadvantages of three modes of BIM-GIS integration.

Mode name	Advantage	Disadvantage
Mode 1: BIM leads and GIS supports	Have a better ability to express the building project itself and its internal information	Lack of ability to integrate with information of other buildings or the surrounding environment
Mode 2: GIS leads and BIM supports	Excellent spatial data processing capabilities	Lack of detailed attribute information for building entities
Mode 3: BIM and GIS equally involved	Have a good combination of information processing application capabilities in the building itself and the surrounding space	The cumbersome data increases the burden of computer processing

Abbildung 8: Vor- und Nachteile der verschiedenen BIM-GIS Integrationen (Wang et al. 2019a)

Zusammenfassend kann man sagen, dass BIM qualitativ hochwertige und detaillierte semantische und geometrische Informationen über einzelne Gebäude und Projekte umfasst. Informationen außerhalb des Projekts sind jedoch nicht Teil des Modells. GIS können räumliche, großmaßstäbige Analysen durchführen. Allerdings fehlen einem GIS Details über einzelne Gebäude. Diese beiden Modelle zu kombinieren, kann auf verschiedene Arten erfolgen. Eine Möglichkeit ist die Daten zu fusionieren. Dabei werden die Daten des BIM vom IFC-Format (Industry Foundation Classes) ins CityGML-Format umgewandelt. Da die IFC-Daten meistens im Level of Detail (LoD) 5, und die CityGML im LoD0-LoD4 vorliegen, müssen die IFC-Daten herunterskaliert werden. Anschließend müssen die Informationen aus dem IFC-Format in eine passende Klasse des CityGML-Formats überführt werden. Bei diesen Schritten kommt es zu Datenverlusten. Um dieses Problem zu minimieren wurde ein Erweiterungsmechanismus entwickelt (Variante 2), der die IFC- und CityGML-Standards so erweitert, dass eine bessere Integration gelingt. Ein Problem dieser Variante sind inkonsistente semantische Informationen. Die dritte Variante reichert die beiden Formate zunächst semantisch an. Es wird ein Ontologiesystem geschaffen, das Beschreibungen der Elemente beider Formate enthält. Die Ontologie ist eine Anwendung der Computertechnologie, die auf der Logik beider Systeme aufbaut. Dadurch sollen die semantischen Informationen beim Überführen der IFC-Daten ins CityGML-Format nicht verloren gehen. Ein weiterer Vorteil der dritten Variante ist, dass sie eine Interoperabilität zwischen den beiden Formaten schafft, wohingegen bei den ersten beiden Wegen die Transformation von IFC nach CityGML irreversibel ist. (Xia et al. 2022)

2.2.4 City Information Model (CIM)

Oft spricht man von einem City Information Model (CIM), wenn man BIM, GIS und Internet of Things (IoT) kombiniert. Das Ziel eines CIM ist die digitale Repräsentation eines urbanen Gebiets. (Zhang 2024) Diese Darstellung kann zwei- oder dreidimensional erfolgen und ist die Grundlage für viele DZ. Der Unterschied zwischen einem CIM und einem DZ liegt vor allem in der Simulations- und Analysefähigkeit. Ein CIM integriert

verschiedene Daten und präsentiert diese. Ein DZ ermöglicht eine Echtzeit-Synchronisation und erweitert das Modell um Vorhersagen, Simulationen und automatische Entscheidungen. Ein weiterer Unterschied besteht in der Interaktion mit dem Modell. Ein DZ ist wesentlich dynamischer und kann vom Benutzer immer wieder angepasst werden. (Xu et al. 2021) (DIN Media GmbH 2024)

2.2.5 3D-Modell

Eine klare Abgrenzung zwischen DZ und klassischen 3D-Modellen ist notwendig, um das Potenzial von DZ zu verstehen. Ein 3D-Modell ist eine Visualisierung eines physischen Objekts. Dabei kann es sich um ein Gebäude, eine Maschine oder eine Stadt handeln. Jedoch fehlen einem 3D-Modell einige Eigenschaften im Vergleich zum DZ. (Shahat et al. 2021) Dazu zählen beispielsweise die Modellierung von semantischen Informationen, die Abfrage von Echtzeitdaten oder die Simulation von Szenarien. Im Grunde genommen ist das 3D-Modell einer Stadt meist nichts anderes als eine dreidimensionale Karte. (Batty 2024) 3D-Modelle sind jedoch sehr bedeutsam für die meisten UDZ, da sie oft die Grundlage des UDZ bilden.

2.2.6 3D-Modell + BIM

Da es 3D-Modellen an attributiven Eigenschaften fehlt, gab es bereits viele Projekte, die untersuchten, wie man 3D-Modelle mit BIM verknüpfen kann. Genauer gesagt soll ähnlich wie bei der Kombination aus GIS und BIM, eine Auswahl an Eigenschaften aus dem BIM extrahiert und ins 3D-Modell importiert werden. Im Gegensatz zur Kombination aus BIM und GIS liegen bei dieser Variante die Stärken in der Visualisierung und der Betonung der wichtigsten Eigenschaften. Die Analysefähigkeit ist jedoch bei weitem nicht so hoch wie bei der Kombination aus GIS und BIM. (Xia et al. 2022)

2.2.7 Zusammenfassung

Die Analyse der verschiedenen Begriffe, die oft im Zusammenhang mit DZ verwendet werden zeigt, dass GIS, BIM, 3D-Modelle und CIM sowie sämtliche Kombinationen dieser Modelle und Systeme eng mit DZ vernüpft sind. Da es an einer einheitlichen Definition für einen DZ fehlt, ist auch die Abgrenzung zu anderen Begriffen oft nicht einfach. Drei Eigenschaften können jedoch hervorgehoben werden, die ein DZ erfüllen sollte und die in ihrer Kombination den DZ von anderen verwandten Systemen abgrenzen. Die **Echtzeitfähigkeit**, die **Simulations-/Prognosefähigkeit** und die **Interaktion mit dem DZ**. Die Echtzeitfähigkeit ist relativ zu betrachten, da eine vollständige Live-Übertragung und Darstellung der Daten nicht möglich sind. Doch ein DZ versucht die Daten so aktuell wie möglich zu halten und ist klar von statischen Systemen zu unterscheiden. Die oben beschriebenen Systeme und Modelle bilden oft die Grundlage oder fungieren als Bausteine für einen DZ. Sie sind zwar teilweise in der Lage Echtzeitdaten zu übertragen oder Simulationen durchzuführen. Jedoch sind das nicht ihre primären Funktionen. Deshalb sind oft nur sehr spezifische oder einfache Simulationen möglich und die Übertragung von Live-Daten beschränkt sich auf einen

bestimmten Anwendungsbereich. Ein DZ versucht das reale Objekt so gut es geht zu spiegeln. Das allein unterscheidet ihn noch nicht von einem CIM. Allerdings kann mit diesem Spiegelbild der Realität umfangreich kommuniziert werden. Man kann Simulationen durchführen und in manchen Bereichen direkt Einfluss auf die Realität ausüben. (Grieves und Vickers 2017) (Xia et al. 2022) (Batty 2024) (Xu et al. 2021) (Shahat et al. 2021) Die Bedeutung einer umfangreichen Kommunikation zwischen Realität und DZ heben auch Jafari et al. (2023) in ihrer Arbeit hervor. Lehner (2020) ist der Meinung, ein CIM sei die optimale Grundlage für einen UDZ (Lehner und Dorffner 2020). Wie Lehner beschreiben die meisten Experten die erwähnten verwandten Systeme als Baustein oder Fundament für einen DZ beziehungsweise UDZ.

Tabelle 1 fasst die Stärken und Schwächen der verwandten Systeme und Modelle zusammen.

Tabelle 1: Stärken und Schwächen der verwandten Systeme

System/Modell	Stärken	Schwächen
GIS	– großmaßstäbige Analysen – solide Datengrundlage – räumliche Analysen	– meistens nur aktuelle oder vergangene Daten – kleinmaßstäbige Darstellung
BIM	– hoher Detaillgrad – Darstellung von Innenräumen	– großmaßstäbige Anwendungen – keine Analysefähigkeit
BIM + GIS	– Kombination der Stärken beider Systeme – groß- und kleinmaßstäbige Informationen	– Systeme müssen zusammengeführt werden – Informationen gehen bei Zusammenführung verloren
CIM	– umfangreiche Repräsentation eines urbanen Gebiets – Kombination von BIM, GIS und IoT	– beschränkte Analysefähigkeit – meist keine Echtzeit-Synchronisation
3D-Modell	– einfache und effektive Visualisierung – Dreidimensionalität – Durchführung einfacher dreidimensionaler Analysen	– fehlende semantische Informationen – beschränkte Analysefähigkeit – keine Echtzeitsynchronisation
3D-Modell + BIM	– Fokus auf Visualisierung und Betonung der wichtigsten Eigenschaften – Erweiterung des 3D-Modells um wichtigste semantische Informationen	– fehlende Analysefähigkeit – Probleme bei der Integration von BIM-Daten

2.3 Digitaler Zwilling im urbanen Kontext

2.3.1 Smart Cities

Im heutigen digitalen Zeitalter versuchen immer mehr Städte, sich in Richtung „Smart City“ zu entwickeln. Es gibt zwar keine eindeutige Definition dieses Begriffs, aber es gibt einige Dinge, die in fast allen Definitionen vorkommen. Eine Smart City zeichnet sich dadurch aus, dass eine Vielzahl an Objekten und Sensoren miteinander verknüpft ist und dadurch Ressourcen geschont und besser verteilt werden können. Auch das Miteinbeziehen der Bevölkerung ist Teil des Smart City-Konzepts. Insgesamt besteht eine Smart City aus einer Unmenge an Daten und Objekten. Diese Daten und Verbindungen brauchen eine Plattform, um visualisiert und analysiert werden zu können. Denn nur so können Entscheidungen getroffen werden, um die intelligente Stadt noch nachhaltiger und effizienter zu machen. Hier können Digitale Zwillinge helfen. Die Fähigkeiten, aktuelle und Live-Daten aufzubereiten und darzustellen, ist eine der großen Vorteile der DZ. Deshalb stehen die beiden Begriffe in engem Zusammenhang. Durch die stetig wachsende Relevanz von Smart Cities investieren immer mehr Städte und Regionen in die Entwicklung von DZ. (Alvi et al. 2025; Ruohomaki et al. 2018) Abbildung 9 zeigt die Anzahl an Studien zum Thema DZ im Zusammenhang mit Smart Cities.

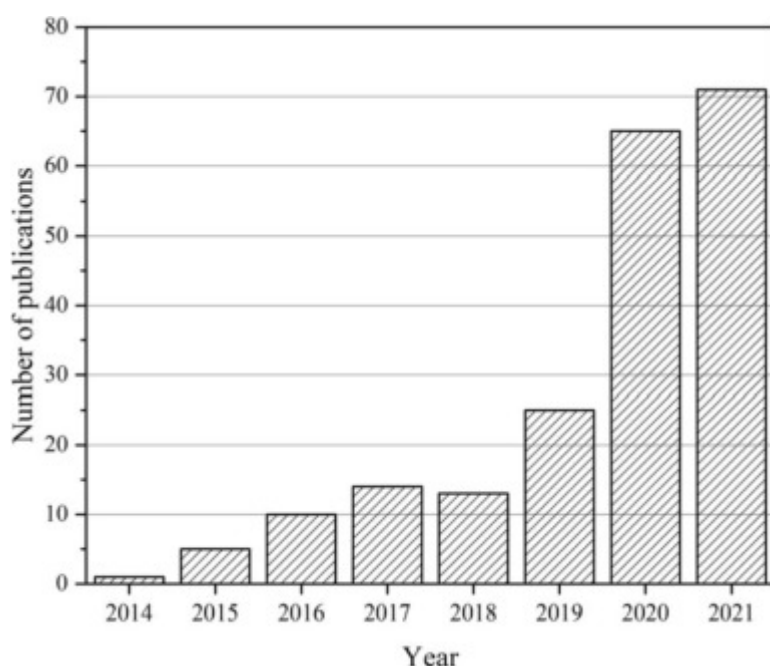


Abbildung 9: Publikationen zum Thema DZ im Zusammenhang mit "Smart Cities" (Wang et al. 2023b)

Smart Cities sind ein großer internationaler Markt. Laut einer Analyse von Grand View Research Inc. (2019) liegt der Marktwert für Smart Cities im Jahr 2025 bei mehr als 2.376 Milliarden US-Dollar. Vor allem kleine und mittelgroße Städte und Gemeinden verfügen nicht über genug Ressourcen, um selbst eine Smart City zu entwickeln und fortzuführen. Diese Voraussetzungen werden von großen Technologieunternehmen ausgenutzt. Sie bieten an, den Aufbau und das Management von Smart Cities zu übernehmen. Dieses Modell zwingt die Städte und Gemeinden jedoch oft in eine Abhängigkeit. Sie sind

abhängig vom Unternehmen und müssen ihre "eigenen" Daten zurückkaufen. Des Weiteren führt diese Variante oft dazu, dass die Stadt selbst weniger bestimmen und umsetzen kann und die Beteiligung der Bürger zurückgeht. Um dieses Szenario zu verhindern, gibt es bereits Projekte, die auf Basis von Open-Source-Programmen einen UDZ aufbauen (z. B. Herrenberg). Dadurch sind die Gemeinden und Städte unabhängig von großen Technologiekonzernen. (Dembski et al. 2020)

2.3.2 Urbaner Digitaler Zwilling

Digitale Zwillinge, die im Kontext von Smart Cities oder urbanen Umgebungen allgemein verwendet werden, werden in dieser Arbeit als Urbane Digitale Zwillinge (UDZ) bezeichnet. Was diese spezielle Form von DZ auszeichnet, wird im Kapitel "3 Urbaner Digitaler Zwilling" behandelt.

2.3.3 Hierarchie von Urbanen Digitalen Zwillingen

Eine Stadt besteht aus verschiedenen Objekten, die miteinander interagieren. Dasselbe gilt für einen UDZ. Man kann sagen, dass ein UDZ aus mehreren DZ besteht. Diese DZ können wiederum aus verschiedenen DZ bestehen. Abbildung 10 zeigt diesen hierarchischen Aufbau verschiedener DZ. In der obersten Stufe steht ein nationaler DZ. Ein solch nationaler DZ besteht aus mehreren Zwillingen auf Stadtebene, sprich UDZ. Diese UDZ bestehen aus mehreren Infrastruktur-Zwillingen. Dazu zählen Gebäude, Straßen, Brücken etc. Jeder dieser Infrastruktur-Zwillinge kann wiederum aus mehreren System-Zwillingen bestehen. Ein Beispiel wäre die Ausstattung eines Gebäudes (Heizung, Klimaanlage etc.). (Lu et al. 2020) Damit diese DZ miteinander verknüpft werden können, sind einheitliche Austausch- und Streamingformate notwendig. Diese werden im Kapitel 3.4.4 näher beschrieben.

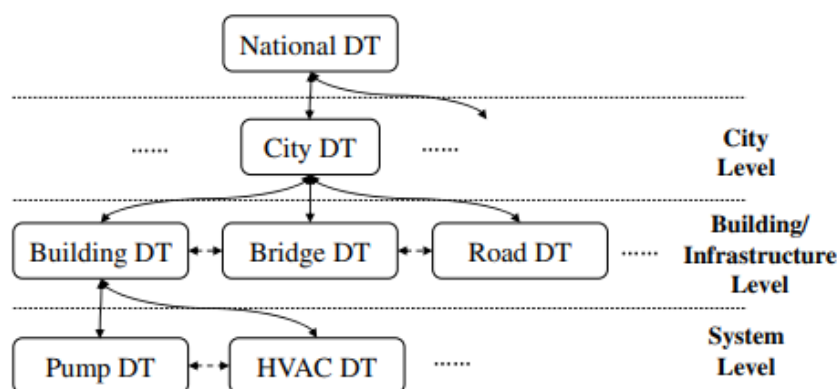


Abbildung 10: Hierarchischer Aufbau und Verbindungen verschiedener DZ (Lu et al. 2020)

3 Urbaner Digitaler Zwilling (UDZ)

Alvi et al. (2025) sprechen von “Smart city digital twins”. Dieser Begriff steht dem des UDZ sehr nahe. Sie beschreiben diese Form von Digitalen Zwillingen als fortgeschrittene technologische Modelle, die eine urbane Umgebung genau und interaktiv simulieren. Sie unterteilen “Smart city digital twins” oder UDZ in 4 Kategorien.

Die erste Form sind **passive UDZ**. Sie konzentrieren sich auf die möglichst genaue Replikation der städtischen Infrastruktur und dienen dazu, den aktuellen Stand der Stadt festzuhalten. Passive UDZ bieten eine gute Grundlage für einfache städtische Analysen und Planungen. **Vorhersagende UDZ** ermöglichen die Verbindung zwischen realer und digitaler Welt. In solchen Modellen können Live-Daten dank Sensoren in den UDZ integriert werden. Durch diese Daten können Muster erkannt und Prognosen erstellt werden. Informationen über zukünftige Entwicklungen sind in dieser Form des UDZ möglich. Die dritte Form sind **programmierbare UDZ**. Simulationen sind hier noch umfangreicher und feingliedriger möglich. Die finale, vierte Form sind **autonome UDZ**. Diese UDZ treffen selbst Entscheidungen und haben für sämtliche Fragestellungen im urbanen Kontext mehrere Lösungsvorschläge. Sie optimieren sich selbst und die reale Welt automatisch. (Alvi et al. 2025) Aktuell kann man die meisten UDZ einer der ersten beiden Kategorien zuordnen. Von der vierten Form sind alle aktuellen UDZ noch sehr weit entfernt. Neben den technologischen Hürden kann die vierte Stufe auch ethisch hinterfragt werden, da das Modell automatisch Entscheidungen trifft. Abbildung 11 bietet einen Überblick über die 4 Formen bzw. Stufen der UDZ. Wie bereits in Kapitel 2 erwähnt, gibt es keine eindeutige Definition für DZ. Dasselbe gilt auch für DZ im urbanen Raum (in dieser Arbeit UDZ genannt). Oft ist unklar ab wann man von einem UDZ sprechen kann. Somit sind auch weitere Unterteilung, wie sie Alvi et al. (2025) vornehmen, oft subjektiv und der Übergang zwischen den verschiedenen Formen ist fließend. Aufbauend auf den verschiedenen Definitionen für DZ und UDZ legt diese Arbeit folgende Eigenschaften fest, die einen UDZ definieren und von anderen Modellen abgrenzen.

1. Echtzeitfähigkeit und automatischer Datenfluss
2. Simulations- und Prognosefähigkeit
3. Kommunikation zwischen Realität, digitalem Modell und Anwender
4. Anwendung für einen gesamten urbanen Raum oder einen Teil davon (Stadtteile, Viertel, Bezirke)

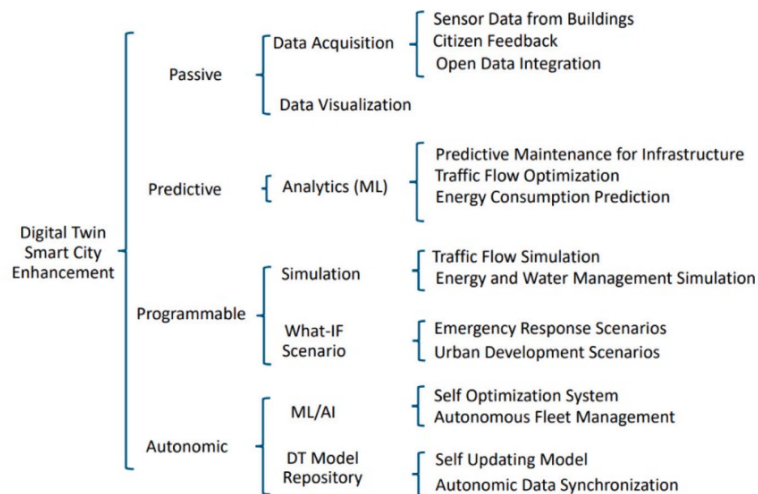


Abbildung 11: Stufen/Formen von UDZ (Alvi et al. 2025)

Petrova et al. (2019) haben die Funktionsweise eines UDZ näher beschrieben. Sie stellen die Prozesse innerhalb eines UDZ als Kreislauf dar (Abbildung 12). Dieser Kreislauf besteht aus sechs verschiedenen Phasen. Die CREATE-Phase besteht aus der Datenkollektion: Es werden Daten von verschiedenen Sensoren oder durch bestimmte Methoden abgefragt. Die INTERACT-Phase liefert die Informationen aus der realen Welt an das virtuelle Modell. Anschließend werden die Daten bereinigt und gespeichert (AGGREGATE). Daraufhin werden durch Analysen und Simulationen (ANALYSE) wertvolle Informationen über die Stadt gewonnen (INSIGHT). Der letzte Schritt, DECISION, ist das Ergebnis der Informationen und führt häufig dazu, dass wieder neue Daten generiert werden (CREATE). Somit ist der Kreislauf geschlossen. (Petrova-Antonova und Ilieva 2019)

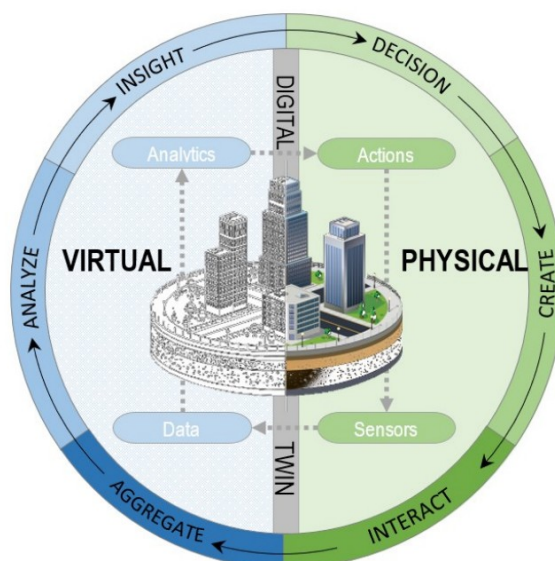


Fig. 3. Digital twin

Abbildung 12: Die 6 Phasen eines UDZ (Petrova-Antonova und Ilieva 2019)

Für diese Arbeit wurde die Funktionsweise von UDZ auf vier Phasen reduziert. Kapitel 3.1 (Datenquellen) beschreibt, wie Daten für einen UDZ gewonnen werden können und welche Herausforderungen in dieser Phase auftreten. Anschließend werden die Daten manipuliert, damit eine effektive Speicherung und Weiterverarbeitung möglich ist. Diesen Teil beschreibt das Kapitel 3.2 (Datenverarbeitung und –management). Kapitel 3.3 (Simulation & Analyse) entspricht der ANALYSE-Phase von Petrova et al. (2019). Diese Phase ist das Herzstück eines UDZ und schafft wertvolle neue Einblicke. Im letzten Schritt werden die Ergebnisse und Informationen dargestellt (3.4 Visualisierung).

Shahat et al. (2021) fassen in ihrer Arbeit die Potenziale von UDZ zusammen. Bei der Betrachtung der einzelnen Kategorien fällt auf, dass sich diese teilweise mit den ausgearbeiteten Phasen überschneidet (Visualisierung und Datenmanagement). Es werden aber auch mögliche Anwendungsgebiete von UDZ in das Potenzial-Diagramm integriert. Das Diagramm zeigt nicht nur Potenziale, sondern auch Bausteine, Anwendungsgebiete und Herausforderungen (Abbildung 13).

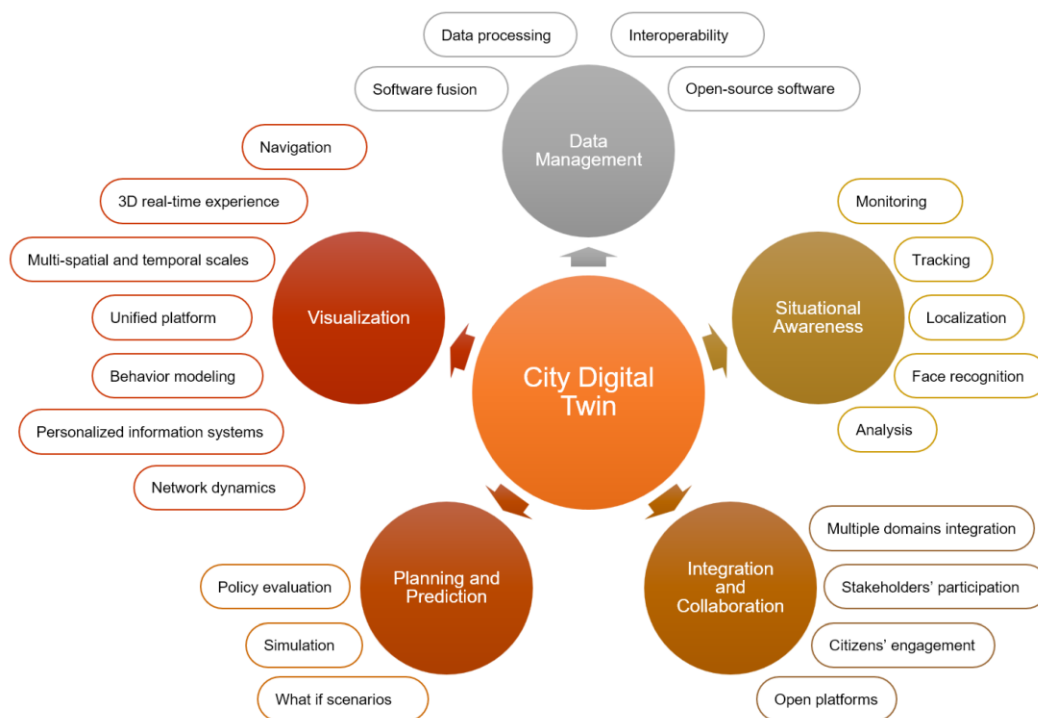


Abbildung 13: Bausteine, Potenziale, Herausforderungen und Anwendungsgebiete von UDZ (Shahat et al. 2021)

3.1 Datenquellen

Die Datenakquise ist das Fundament jedes UDZ. Aufgrund der Heterogenität und großen Datenmenge im urbanen Kontext ist eine strukturierte und effiziente Datenakquise eine große Herausforderung. Es gibt verschiedene Möglichkeiten, Daten zu gewinnen. (Lu et al. 2020) In diesem Kapitel werden einige davon näher beschrieben.

3.1.1 Satellitendaten

In manchen Bereichen sind Satellitendaten sehr wichtig. Vor allem bei ökologischen Analysen und Simulationen werden oft Daten von Satelliten herangezogen. So sind Satelliten beispielsweise in der Lage, hochauflösende Informationen über die Luftqualität einer Stadt zu liefern. Auch bei der Vorhersage von Extremwetterereignissen sind Satellitendaten meist essenziell. Ein großer Vorteil von Satellitendaten ist die meist kostenlose Verfügbarkeit. Dadurch können Städte Kosten bei bestimmten Analysen einsparen. Allerdings weisen Satellitendaten auch einige Limitationen auf. Dazu zählen die geringe zeitliche Abdeckung, Störungen aufgrund von Verhältnissen in der Atmosphäre (Wolken, Feuchtigkeit etc.) und Reflexionen an der Erdoberfläche. (Kaginalkar et al. 2021) Es besteht die Möglichkeit, dass in Zukunft Live-Daten von Satelliten zu einer täglichen Aktualisierung der Landnutzung beitragen können. Eine solche Integration von Satellitendaten würde jedoch dazu führen, dass eine sehr große Datenmenge täglich verarbeitet werden muss. Im Kontext von UDZ können Satellitendaten vor allem als Datenquelle für Analysen verwendet werden. Die physische Basis, wie etwa Gebäude und andere Objekte, können nur sehr schwer und ungenau aus Satellitendaten gebildet werden. (Batty 2024)

3.1.2 Sensoren und IoT

Das wachsende Internet of Things (IoT) bietet wertvolle Informationen für moderne Smart Cities (Shahrour und Xie 2021). Die Bedeutung des IoT ist im Einklang mit der von Smart Cities gewachsen. Abbildung 14 zeigt die Zahl der Publikationen im “Web of Science” zu den Schlagwörtern “Smart City”, “IoT” und “Crowdsourcing”.

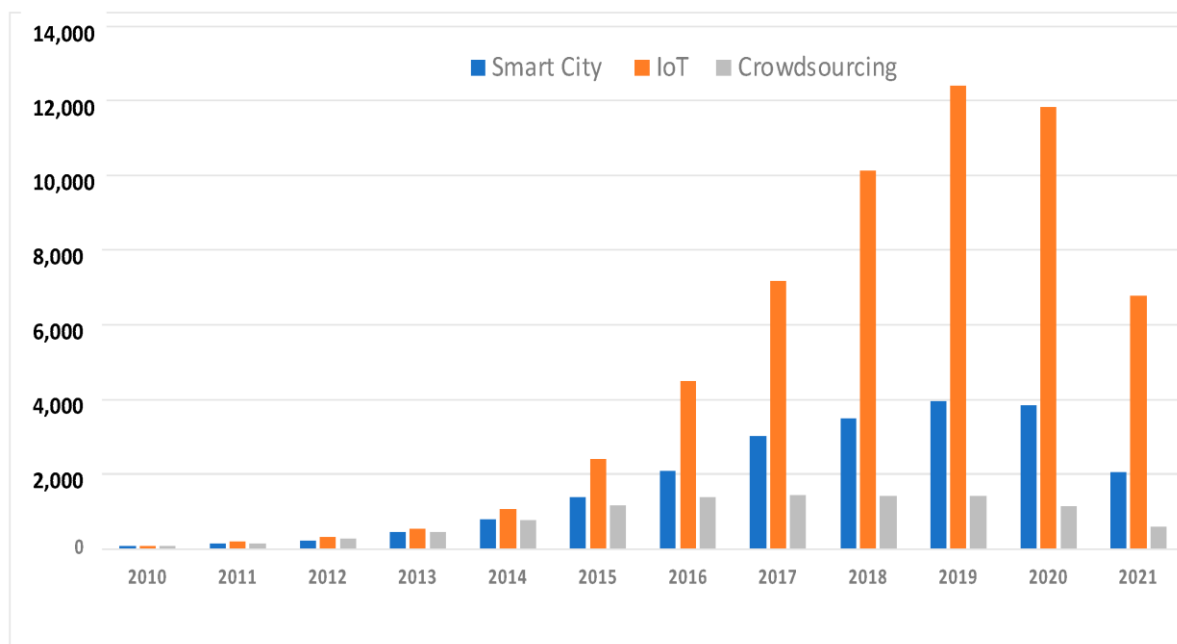


Abbildung 14: Publikationen zu den Themen "Smart City", "IoT" und "Crowdsourcing" (Shahrour und Xie 2021)

Madakam et al. (Madakam et al. 2015) beschreiben das IoT als offenes Netzwerk intelligenter Objekte, die die Fähigkeit besitzen, Daten automatisch aufzunehmen und zu teilen. Das Konzept der Smart City ist aufgrund des stetig wachsenden IoT aufgekommen. Objekte und Sensoren werden miteinander verbunden und liefern wertvolle Informationen für Analysen und Simulationen auf städtischer Ebene. (Khajavi et al. 2019) In diesem Ökosystem profitieren UDZ stark von der direkten Kommunikation von Objekten und Sensoren. Eine direkte und beidseitige Kommunikation zwischen Sensoren auf der einen und einem UDZ auf der anderen Seite hat das Potenzial die kurzfristige urbane Planung zu verbessern. Diese Verbesserungen können sich langfristig bemerkbar machen, da eine stetige kurzfristige Verbesserung auch zu einer langfristigen Änderung führt. (Weil et al. 2023)

Sensoren zählen zu den verlässlichsten und qualitativ hochwertigsten Datenquellen eines UDZ. Die Anzahl an verschiedenen Sensoren ist mittlerweile sehr umfangreich. Von physikalischen (z. B. Temperatursensoren) über optische (z. B. Kamerasensoren) bis zu GNSS-Sensoren können diverse Sensoren verknüpft werden und wertvolle Daten liefern. Doch für einen umfangreichen UDZ sind viele Sensoren notwendig, die zu hohen Ausgaben führen. Deshalb versuchen viele Städte und Projekte, Alternativen zu finden. Alvi et al. (2025) beschreiben verschiedene Methoden, um Verkehrsdaten alternativ zu erlangen. So versuchten Awan et al. (Awan et al. 2021), Verkehrsdaten aus Lärmdaten zu gewinnen. Andere Möglichkeiten sind das Ableiten von Verkehrsdaten aus Feinstaubwerten oder Social-Media-Daten. Diese Methoden sind niemals so genau wie vor Ort eingesetzte Verkehrssensoren. Jedoch sind sie meist wesentlich günstiger, da keine neuen Daten erhoben werden. Hier gilt es abzuwägen, wie genau die Daten für den UDZ sein müssen und ob sich diese Genauigkeit mit den zur Verfügung stehenden Mitteln erzielen lässt. (Alvi et al. 2025)

Unterirdische Systeme

Die Dichte an unterirdischer Infrastruktur steigt in Städten stetig an und ist für das Funktionieren einer Stadt notwendig. Das Sammeln von digitalen Daten im Zusammenhang mit dieser kritischen Infrastruktur kann helfen, diese effektiver zu warten und Probleme frühzeitig zu erkennen. Während die Infrastruktur an der Erdoberfläche leicht zu erheben und zu validieren ist, weiß man über viele unterirdische Systeme nur sehr wenig. Vor allem über die Charakteristik und den Zustand älterer Leitungen sind kaum Informationen vorhanden. Oft ist nicht einmal der genaue Standort der unterirdischen Infrastruktur bekannt. Noch weniger weiß man oft über die Umgebung der unterirdischen Infrastruktur. Dazu zählt die Leitfähigkeit des Bodens, die Feuchtigkeit, geologische Verwerfungen oder Bodensenkungen. Weiterhin hat man keine Informationen darüber, wie sich die verschiedenen unterirdischen Systeme gegenseitig beeinflussen. Deshalb ist die Integration der unterirdischen Infrastruktur in einen UDZ ein wichtiger Schritt, um diese notwendigen Systeme gut erhalten zu können. Dabei gibt es jedoch einige Herausforderungen. Dazu zählen beispielsweise die genaue

Verortung und Anbindung an den oberirdischen Zwilling. Gerade die Datenerfassung gestaltet sich schwieriger. Meist werden Daten nur bei Umbauarbeiten oder Neuinstallationen aufgezeichnet. Methoden, um unterirdische Infrastruktur aufzuzeichnen, ohne direkt am Objekt zu sein, entwickeln sich rasch, sind aber dennoch mit sehr hohen Kosten verbunden. Wie auch bei anderen UDZ-Daten spielt die Datensicherheit eine große Rolle. Vor allem bei kritischer Infrastruktur muss der Schutz der Daten gewährleistet sein. Auch die Installation von Sensorik unter der Erde birgt besondere Herausforderungen. Denn Grundvoraussetzungen für einen funktionierenden Sensor (Stromanschluss, Internetverbindung, Sicherheit, Wartung etc.) sind in dieser Umgebung schwieriger umzusetzen. Dennoch gibt es bereits vielversprechende unterirdische UDZ, die verdeutlichen, welche Vorteile dadurch entstehen. Zu den Vorteilen zählt beispielsweise eine bessere Planung der Wartungs- und Reparaturarbeiten. Auch bei der Planung und Konstruktion von neuen unterirdischen Leitungen und Systemen kann ein UDZ helfen. Und wie bei einem oberirdischen UDZ können auch unterirdisch Gefahren und Katastrophen simuliert werden. (Lieberman et al. 2017)

3.1.3 Crowdsourcing, mobile Geräte, Social Media

Als Alternative zu Sensoren können mobile Endgeräte (vor allem Smartphones) wertvolle Daten für einen UDZ liefern. Dazu zählen neben der Position des Geräts auch Text-, Bild- oder Videodateien, die veröffentlicht werden. Vor allem in Kombination mit Sozialen Medien sind mobile Geräte in der Lage, eine große Menge an relevanten Daten zu liefern. Sind Social-Media-Beiträge mit einem sogenannten Geotag versehen, sind sie besonders wertvoll. Ein Merkmal, das die Datengewinnung von Social-Media-Kanälen gegenüber den meisten anderen Datengrundlagen abhebt, ist das Messen von Emotionen und Meinungen der Bevölkerung. So können Maßnahmen und Umstände erkannt werden, die zu Verständnis oder Unverständnis in der Bevölkerung führen. Die große Herausforderung beim Arbeiten mit Social-Media-Daten ist die Verarbeitung und das Filtern der großen Datenmengen. Weiters ist es nicht bei allen Kanälen möglich, Daten abzufragen. Ist es möglich, werden die Daten vorab anonymisiert und der Zugang zu ihnen ist in den allermeisten Fällen kostenpflichtig. (Kaginalkar et al. 2021)

3.1.4 Punktwolken

Als universelle 3D-Repräsentation können 3D-Punktwolken fast jedes physische Objekt darstellen. Die Darstellung kann in sämtlichen Maßstäben und Genauigkeiten erfolgen. Die Punktwolken können beispielsweise durch Laserscanning (terrestrisch oder luftgestützt) oder photogrammetrische Methoden erstellt werden. Bei vielen räumlichen Anwendungen spielen Punktwolken eine bedeutende Rolle. Oft werden sie als Grundlage für die Erstellung von 3D-Stadtmodellen verwendet. Eine wichtige Eigenschaft von 3D-Punktwolken ist das Fehlen jeglicher Struktur, Hierarchie oder semantischer Information. Deshalb ist es wichtig, die Punkte mit Informationen anzureichern. Das kann zum Beispiel durch Machine- oder Deep-Learning geschehen.

(Döllner 2020; Wang et al. 2022a) Abbildung 15 skizziert die Segmentierung von 3D-Punktwolken mittels Machine- und Deep-Learning.

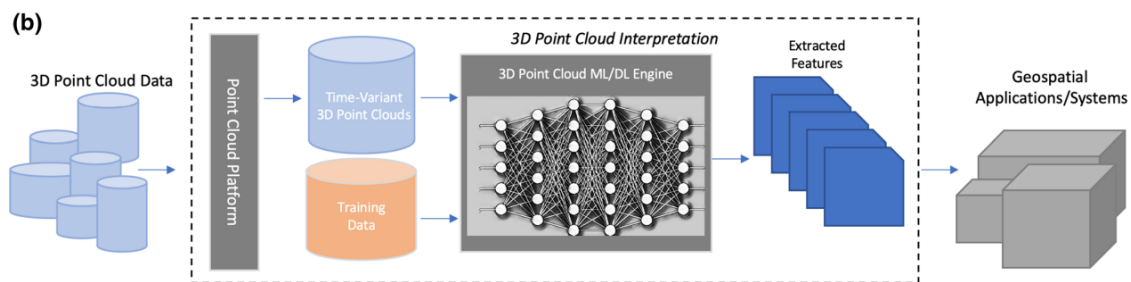


Fig.3 3D point cloud processing: classical workflow **a** based on 3D reconstruction, 3D modelling, and object derivation; **b** ML/DL workflow based on 3D point cloud interpretation

Abbildung 15: Verarbeitung von Punktwolken mittels Machine-/Deep-Learning (Döllner 2020)

3.1.5 Probleme und Herausforderungen

Eine der größten Herausforderungen bei der Erstellung von UDZ ist die Verfügbarkeit von hochwertigen Daten (Alvi et al. 2025). Schlechte Datenqualität zeigt sich häufig durch Ungenauigkeiten, Inkonsistenzen und fehlende Daten. Weiters fehlt bei einigen Datenquellen (IoT-Sensoren, Social-Media-Daten) die Validierung. (Rozony et al. 2024) Erschwerend kommt hinzu, dass Daten im urbanen Raum vielen verschiedenen Stakeholdern gehören. Dazu zählt die Stadt selbst, aber auch wirtschaftliche Institutionen, Privatpersonen oder akademische Einrichtungen. (Alvi et al. 2025) Diese Daten gehören nicht nur unterschiedlichen Stakeholdern, sondern werden auch in unterschiedlichen Dateiformaten gespeichert und weisen verschiedene Strukturen auf (Wang et al. 2023). Die bloße Existenz der Daten reicht also noch lange nicht aus, um die im UDZ nutzen zu können. (Alvi et al. 2025).

In einem späteren Kapitel (3.2.2) wird über einheitliche Standards und Dateiformate bei der Datenverarbeitung gesprochen. Es gibt auch Projekte (Major et al. 2021), die versuchen, bereits bei der Erhebung in bestimmten vordefinierten Formaten zu speichern. Das Problem hierbei ist, dass für diese Art der Speicherung viel Arbeitsaufwand notwendig ist. Die Alternative ist eine spätere Manipulation der Daten, um sie interoperabel zu machen. Dafür gibt es verschiedene Methoden, die im Kapitel 3.2.2 behandelt werden. (Wang et al. 2023b)

Ein weiteres Problem bei der Datenerhebung betrifft die Sensoren. Einige Studien erwähnen, dass viele Sensoren nur eine Datenanfrage gleichzeitig bearbeiten können. Wenn also mehrere Systeme gleichzeitig Datenanfragen an den Sensor schicken, kann das zu Konflikten führen. Hier gibt es bereits Forschungen (Wang et al. 2019a), die Algorithmen verwenden, um die Probleme bei mehreren gleichzeitigen Serviceanfragen zu vermeiden.

Neben den technischen Herausforderungen ist die Gewährleistung der Privatsphäre unabdingbar. Obwohl es Tools zur späteren Anonymisierung gibt, muss dieser Aspekt bereits bei der Datenerhebung berücksichtigt werden, um Verletzungen der Privatsphäre zu vermeiden. (Wang et al. 2023)

Zwischen der Datenerhebung und der Datenverarbeitung müssen die Daten transferiert werden. Um den stetigen Austausch von Daten zu ermöglichen, braucht man gute Verbindungen und ein zuverlässiges Netzwerk. Doch gerade für Daten, die sehr rasch und aktuell abgefragt werden müssen, reicht die Bandbreite heutiger Netzwerke oft nicht aus. Selbst die jüngste Funktechnologie "Long Range Wide Area Network (LoRaWAN)" kann in manchen Anwendungen sehr große Datensätze nicht verarbeiten. Neben einem raschen und zuverlässigen Datentransfer muss auch in diesem Schritt die Sicherheit der Daten gewährleistet sein. (Weil et al. 2023)

Die großen Herausforderungen bei der Datenerhebung insbesondere bei der Erhebung durch Sensoren, die für UDZ von besonderer Bedeutung sind, fassen Simic et al. (2020) in ihrer Arbeit zusammen. Neben der Interoperabilität verschiedener IoT-Systeme, sehen sie die Echtzeitübertragung, das vorzeitige Filtern, das Sicherstellen der Privatsphäre sowie das Gewährleisten von ausreichendem Speicherplatz, starker Rechenleistung und guter Netzwerkqualität als große Herausforderungen. (Simic et al. 2020)

3.2 Datenverarbeitung und –management

Nach der Datenakquise ist die Datenverarbeitung ein essenzieller Schritt. Hochqualitative Daten sind für einen funktionierenden UDZ notwendig. Weisen die Daten Lücken oder Fehler auf, wirken sich diese direkt auf Analysen, Simulationen und Prognosen aus. Zu solchen Lücken oder Fehlern können zum Beispiel Hardwarefehler, eine fehlerhafte Datenfusion oder externe Faktoren wie etwa Stromausfälle führen. (Alvi et al. 2025) Dieses Kapitel beschreibt die Schritte, die notwendig sind, um die Eingabedaten so zu manipulieren, dass sie für die spätere Verwendung im UDZ verwendet werden können und welche Probleme bei diesen Schritten häufig auftreten.

3.2.1 Big Data

Selbst ohne lückenhafte oder fehlerhafte Daten ist die Verarbeitung der Eingangsdaten für UDZ aufgrund der schieren Datenmenge eine große Herausforderung. Deshalb ist die Effizienz der Datenverarbeitung besonders wichtig geworden. Da sich die Menge an Daten in Zukunft weiter vergrößern wird und die Rechenzeit von Computern linear zur Datenmenge verläuft, braucht es immer bessere Hard- und Software, um die enormen Datenmengen verarbeiten zu können. Vor allem die große Anzahl an Sensoren und die steigende Komplexität von 3D-Modellen führen zu einer sehr großen Datenmenge. (Batty 2024) Es sind bereits einige Technologien angewendet worden, um die Effizienz zu steigern. Eine davon ist die 5G-Technologie, die die Verarbeitung von Daten in UDZ beschleunigen kann. Allerdings ist der Einsatz von 5G kostenintensiv und benötigt wesentlich mehr Basisstationen als 4G-Technologien. Außerdem weisen 5G-

Technologien eine geringere Dateninteroperabilität auf. Diese Gründe führen dazu, dass Stand heute nur wenige UDZ 5G nutzen. Weitere Technologien, die helfen können, mit sehr großen Datenmengen zu arbeiten, sind das Cloud- und Edge-Computing (mehr dazu in 3.2.4). (Wang et al. 2023) Allerdings sind die Ressourcen beim Edge-Computing noch stark limitiert und es gibt keine einheitliche Architektur. Das Cloud-Computing ist für die meisten Projekte und Städte zu teuer. Im Zusammenhang mit Big Data können zwei große Herausforderungen hervorgehoben werden. Zum einen das **Speichern der Daten** und zum anderen die **Rechenzeit**. (Weil et al. 2023)

3.2.2 Datenharmonisierung

Um Daten für einen UDZ aufzubereiten, reicht klassisches Datenmodellieren meist nicht aus, da es den Daten an Semantik fehlt. Unter Semantik versteht man das Verständnis, was die Daten genau bedeuten und in welcher Verbindung sie zueinanderstehen. Um die Daten semantisch anzureichern, werden meist Ontologien verwendet. Die Ontologie eines Datenmodells dient als gemeinsame Sprache und soll die Qualität und Konsistenz der Daten gewährleisten. (Singh et al. 2021) Die Herausforderung bei der Ontologie-Bildung ist, dass diese umso komplizierter wird, je heterogener die Eingangsdaten sind. Zur Erstellung von Ontologien ist das Wissen mehrerer Experten in einem iterativen Prozess notwendig. Da sich die Daten für UDZ schnell ändern können und aus diversen Quellen kommen, sind besonders viele Experten und regelmäßige Updates notwendig. Eine weitere Herausforderung ist die Übertragung der Ontologie in relationale Datenbanken. Man kann also zusammenfassen, dass Ontologien zwar sehr mächtig sind, aber dennoch an ihre Grenzen stoßen, wenn es um die Harmonisierung von Daten für UDZ geht. (Karabulut et al. 2024) Der nächste Schritt in der Interoperabilität wäre ein einheitlicher Aufbau von UDZ. Das Ergebnis wäre eine Grundform eines UDZ, die verschiedene Städte implementieren und erweitern können. An solch einer Basisform haben Badawi et al. (Badawi et al. 2021) gearbeitet. (Wang et al. 2023)

Weiters ist es wichtig, bei der Datenverarbeitung Redundanzen zu vermeiden. Hier können so genannte GUIDs helfen. GUID steht für Globally Unique Identifier, und ist eine Zahl mit 128 Bit, die gewährleisten soll, dass jede ID nur einmal vorkommt. Haben die Daten keinen GUID-Standard ist es schwierig zu erkennen, ob ein Objekt in unterschiedlichen Datenquellen vorkommt. Des Weiteren können dieselben Informationen in unterschiedlichen Dateitypen und Einheiten gespeichert sein. So kann beispielsweise die Temperatur in einer Tabelle in Grad Celsius und in einer anderen Tabelle in Grad Fahrenheit angegeben sein. Hier sind Metadaten essenziell, denn sie liefern wertvolle Informationen über die Daten einer Datenquelle. (Lu et al. 2020)

3.2.3 Datenschutz & Sicherheit

Eine der größten Herausforderungen bei UDZ ist die Datensicherheit. Zum einen ist es wichtig, dass der UDZ sicher vor Hackerangriffen ist. Zum anderen müssen die Daten der verschiedenen Stakeholder so weit wie möglich anonymisiert werden. Um die Daten zu

anonymisieren, gibt es verschiedene Möglichkeiten. Dazu zählt beispielsweise die k-Anonymität. Hier werden personenbezogene Informationen so weit generalisiert, dass Gruppen von Datensätzen entstehen, die ident sind und somit nicht mehr auf eine einzelne Person zurückgeführt werden können. Eine Alternative zur Anonymisierung ist Differential Privacy. Bei diesem Verfahren wird gezielt Rauschen erzeugt, um die Daten zu schützen. Jedoch haben einige dieser Varianten Schwierigkeiten mit mehrdimensionalen Daten. (Alvi et al. 2025)

Neben der Anonymisierung ist auch die sichere Speicherung ein wichtiger Aspekt beim Datenschutz. Blockchain ist eine der häufigsten Methoden, Daten und Informationen sicher zu speichern. Dies ist keine Methode der Anonymisierung, sondern der Sicherung. Informationen werden in Blöcken in einer dezentralen Datenbank gespeichert und kryptografisch miteinander verbunden. Durch dieses Aufteilen der Informationen ist eine Manipulation der Daten im Nachhinein nahezu unmöglich. Die Stärke von Blockchain ist es, die Kontinuität der Daten zu gewährleisten. Der Schutz der Privatsphäre ist hier nicht automatisch gewährleistet. (Rahman et al. 2019; Wang et al. 2023b)) Abbildung 16 zeigt die sichere Speicherung von IoT-Daten mittels Blockchain und "Distributed Hash Tables" (DHT). Bei einer DHT sind die Daten nicht auf einem einzelnen, sondern auf mehreren Rechnern gespeichert. Man spricht von Knoten. Jeder Knoten ist für einen bestimmten Teil des Datensatzes bzw. der Tabelle zuständig. Der eindeutige Fingerabdruck der Dateien (Hash) wird in der Blockchain gespeichert. So bleibt die Blockchain schlank und übersichtlich und die DHT ermöglicht den schnellen Datenzugriff. (Li et al. 2019)

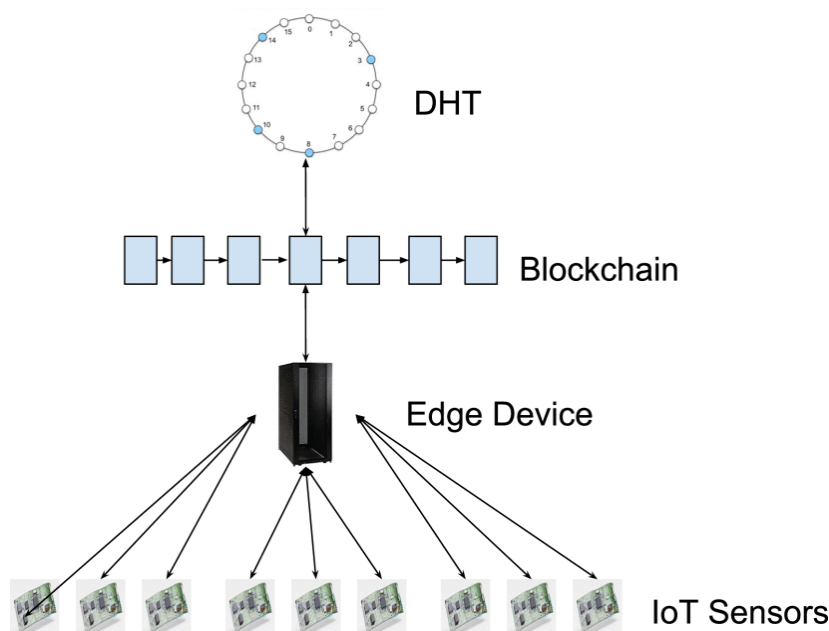


Abbildung 16: Speicherung von IoT-Daten mittels Blockchain und DHT (Li et al. 2019)

Manche Daten eines UDZ sind besonders schützenswert. Dazu zählt beispielsweise kritische Infrastruktur (Elektrizitätssystem, Wasser- und Verkehrsmanagement). Die Herausforderung besteht darin, ein Gleichgewicht zwischen Datenschutz und Datennutzung zu finden. Vor allem bei bürgerzentrierten Daten und Anwendungen ist dieser Aspekt wichtig. Zu solchen Anwendungen zählen beispielsweise “Emotional Mapping” oder die Analyse von Bewegungsmustern anhand von Handydaten. Die Komplexität von UDZ und die verschiedenen Datenkombinationen innerhalb eines UDZ machen es fast unmöglich die Anonymität einzelner Personen zu gewährleisten. Bestehende Regularien, wie beispielsweise das GDPR (European General Data Protection Regulation) sind oft schwer umzusetzen. Ein weiteres Risiko ist der Verlust der Datensouveränität. Vor allem in weniger demokratischen Regimen besteht das Risiko, dass die Daten missbraucht werden. Dies ist auch ein häufiger Kritikpunkt, wenn das Management an ein Unternehmen ausgelagert wird. Deshalb sind die Transparenz und das Miteinbeziehen der Bevölkerung essenziell. Die Komplexität von UDZ macht es jedoch oft schwer, für Transparenz zu sorgen. (Weil et al. 2023)

Zusätzlich sind viele UDZ nicht barrierefrei und schließen bestimmte Bevölkerungsgruppen aus. Oft werden die sozialen, ethischen und politischen Aspekte von UDZ nicht oder nicht ausreichend beachtet. Das kann dazu führen, dass sich manche Gruppen ausgeschlossen oder benachteiligt fühlen. In diesem Zusammenhang werden bevölkerungszentrierte UDZ immer wichtiger. Bei dieser Art von UDZ steht der Mensch im Zentrum. Das Ziel ist, die Lebensqualität aller Bewohner zu verbessern. Um einen solchen UDZ betreiben zu können, ist eine effektive Verwaltung notwendig. Es muss gelingen, diverse Stakeholder laufend in die Prozesse des UDZ einzubinden. Dabei darf der UDZ jedoch nicht träge und langsam werden. (Weil et al. 2023)

Da ein UDZ den Anspruch hat, Daten in nahezu Echtzeit zu übertragen, müssen die Methoden zum Datenschutz nicht nur sicher, sondern auch schnell funktionieren. Häufig wird Machine Learning eingesetzt, da so die gewünschte Geschwindigkeit erreicht werden kann. (Weil et al. 2023; Zhang und Tao 2021)

3.2.4 Berechnungsmodelle

Die Menge und Vielfältigkeit der Daten eines UDZ stellen viele Städte vor große Herausforderungen. Eine wichtige Entscheidung ist die Wahl der passenden IT-Infrastruktur. Bei dieser Entscheidung spielen vor allem die Leistungsfähigkeit, die Kosten, die Skalierbarkeit und die Sicherheit eine bedeutende Rolle.

Weil et al. (2023) arbeiten vier verschiedene Berechnungsarten heraus. Beim On-Premise Computing (1) wird die gesamte IT-Infrastruktur von der jeweiligen Organisation selbst betreut. Die Vorteile dieser Variante liegen in der Kontrolle über den Aufbau und die Sicherheit der Infrastruktur. Allerdings können beim On-Premise Computing sehr hohe Kosten entstehen. Auch die Skalierbarkeit ist ein Nachteil dieses Ansatzes. Cloud Computing (2) hat seine Stärken in der Skalierbarkeit und der Kosteneffizienz. Bei dieser

Form der Berechnung werden die Ressourcen (Speicherplatz, Rechenkapazität, Software) über das Internet in Anspruch genommen. Wie bereits erwähnt ist die Kosteneffizienz ein großer Vorteil, denn man zahlt nur für die tatsächlich beanspruchte Leistung. Die Skalierbarkeit ist jedoch nur bis zu einer gewissen Größe ein Vorteil, denn bei sehr großen Projekten ist diese Variante kostenintensiv. Ein weiterer Nachteil ist die Sicherheit der Daten, da man nicht dasselbe Ausmaß an Kontrolle wie beim On-Premise Computing hat. Edge Computing (3) versucht die Verarbeitung der Daten näher an den Daten selbst durchzuführen. Dabei wird die, zu transferierende Datenmenge minimiert, um Kosten zu reduzieren. Außerdem ermöglicht Edge Computing eine raschere Datenabfrage und ist somit für Echtzeit-Übertragungen gut geeignet. Neben der Datensicherheit ist die Betreuung der verschiedenen Edge-Geräte eine große Herausforderung bei dieser Methode. Der vierte Weg die Berechnungen im UDZ durchzuführen ist das Cluster Computing (4). Es werden mehrere Computer miteinander verbunden und arbeiten als ein einzelnes großes System. Dadurch steigt die Rechen- und Speicherkapazität. Weitere Vorteile sind eine verbesserte Skalierbarkeit und eine größere Fehlertoleranz. Allerdings sind die Erstellung und das Management dieser Variante sehr kostenintensiv. Zusätzlich ist die Koordinierung der verschiedenen Computer eine Herausforderung. Vor allem bei mangelnder Datenkonsistenz treten beim Cluster Computing häufig Probleme auf. (Weil et al. 2023)

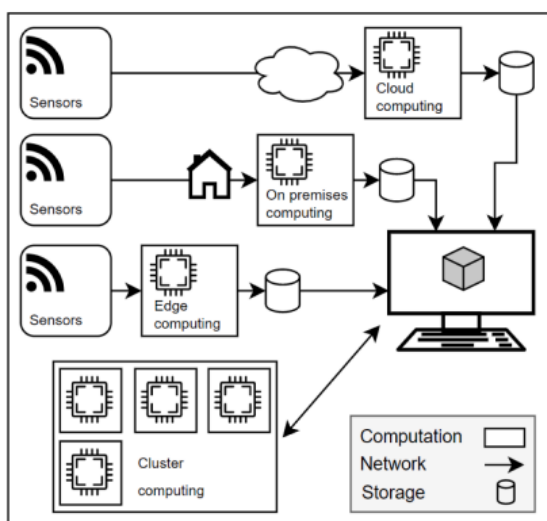


Abbildung 17: Cloud-, On-Premise & Edge-Computing als Teil eines UDZ (Weil et al. 2023)

Bei der Beschreibung dieser vier Varianten wird deutlich, dass sich die optimale Lösung von UDZ zu UDZ unterscheidet. Die optimale Lösung hängt davon ab, welche Anforderungen und Ziele an den UDZ gestellt werden und welche finanziellen Ressourcen vorhanden sind.

3.2.5 Datenspeicherung

Da die Datenmenge, die für einen UDZ benötigt wird, in der Regel sehr groß ist, spielt die Speicherung eine große Rolle. Wie bei den Berechnungsmethoden gibt es auch hier

verschiedene Arten, die Daten eines UDZ zu speichern. Es können drei Möglichkeiten herausgestrichen werden: **relationale Datenbanken**, **Datenbanken ohne SQL** und sogenannte **Data Lakes**. Relationale Datenbanken, wie etwa PostgreSQL oder MySQL sind weit verbreitet, um strukturierte Daten zu speichern und damit komplexe Abfragen auszuführen. Ein Vorteil von relationalen Datenbanken ist das Bewahren der Konsistenz und Integrität der Daten. Mit semistrukturierten Daten haben relationale Datenbanken jedoch Schwierigkeiten. Diese treten häufig im Kontext von UDZ auf, da viele Eingangsdaten nicht voll strukturiert sind. Dazu zählen beispielsweise Sensor- und Social-Media-Daten. (Batty 2018) NoSQL-Datenbanken wie etwa MongoDB oder Cassandra können mit diesen Daten besser umgehen. Allerdings können komplexe Abfragen in diesen Datenbanken teilweise nicht ausgeführt werden. (Strauch) Sind die Daten noch unstrukturierter, können Data Lakes verwendet werden. Diese speichern die Daten in ihrer ursprünglichen, rohen Form. Hier kann man leicht den Überblick verlieren und es entsteht ein "Data Swamp". Außerdem dauern Abfragen länger, da die Daten weniger strukturiert gespeichert werden. (Hai et al. 2023)

3.2.6 Echtzeitfähigkeit und Synchronisation

Eine große Herausforderung bei UDZ ist die Datensynchronisation. Zwischen den Kosten für die Synchronisation und der Datenqualität besteht ein direkter Zusammenhang. Je aktueller man die Daten halten möchte, desto höher sind die Kosten für Informationstechnologie und Rechenleistung. Viele Organisationen verwenden aktuell Batch-Synchronisationen. Das heißt, dass die Daten zu einem bestimmten Zeitpunkt aktualisiert werden. Dieser Zeitpunkt liegt meist außerhalb der Geschäftszeiten. Hat ein UDZ jedoch den Anspruch, echtzeitfähig zu sein, ist das keine Lösung. Eine komplette Echtzeitübertragung ist Stand heute unrealistisch, da die aufgezeichneten Daten noch manipuliert werden müssen, um miteinander zu funktionieren. (Lu et al. 2020)

3.2.7 Probleme und Herausforderungen

Wie bereits mehrfach erwähnt gibt es bei der Erstellung eines UDZ einige Herausforderungen im Bereich des Datenmanagements. Simic et al. (2020) unterstreichen sechs Herausforderungen im Zusammenhang mit Big Data und UDZ:

1. unterschiedliche Dateiformate
2. unterschiedliche Dateneigentümer
3. verschiedene Datenqualität
4. hohe Kosten (Wer bezahlt dafür?)
5. stetig steigende Datenmenge
6. Sicherheit und Datenschutz

Es gibt bereits verschiedene Technologien, die mit unterschiedlichen Dateiformaten umgehen können und diese in ein einheitliches Format überführen. Diese funktionieren meist nach dem ETL-Prinzip (Extract-Transform-Load). Jedoch gibt es hier keine Lösung, die für alle Daten beziehungsweise UDZ funktioniert. So muss für jede neue

Datenschicht und jeden neuen UDZ eine eigene Lösung entwickelt werden. Somit bleiben die fehlerfreie, automatische Ausbesserung und Umwandlung der Eingangsdaten weiterhin eine der größten Herausforderungen. Um diesen Schritt mit großen Datenmengen gewährleisten zu können, ist das Herstellen von semantischen Verbindungen wichtig. (Lu et al. 2020)

Die Interoperabilität wird von mehreren wissenschaftlichen Publikationen als eine der größten Herausforderungen bei UDZ gesehen. Weil et al. (2023) haben 200 Quellen zum Thema “UDZ” untersucht. Probleme bei der Interoperabilität wurden in 19 % dieser Artikel erwähnt. Es gibt viele Faktoren, die die Interoperabilität erschweren. Dazu zählen beispielsweise sogenannte Datensilos. Dabei handelt es sich um isolierte Datenspeicher, die den Datenaustausch zwischen verschiedenen Systemen verhindern (Eicker et al. 2020). Auch Methoden werden oft sehr isoliert eingesetzt. Das führt dazu, dass die Methode nur für einen bestimmten UDZ oder Datensatz verwendet werden kann. Des Weiteren sind manche Sensoren eng mit einer bestimmten Software verknüpft und lassen sich nur sehr schwer mit anderen Applikationen verbinden. Obwohl es mittlerweile Formate gibt, die die Echtzeitübertragung und den Datenaustausch vereinfachen, unterstreichen viele Wissenschaftler den Bedarf an standardisierten Metadaten und Austauschformaten. (Weil et al. 2023)

Auch Probleme im Zusammenhang mit der Semantik der Daten wurden von knapp einem Viertel der von Weil et al. (2023) untersuchten Literatur, erwähnt. Die Heterogenität der Daten im Format und in der Qualität und die Komplexität urbaner Systeme erschweren die Umsetzung einheitlicher semantischer Regeln oder standardisierten Ontologien. Deshalb verwenden aktuell nur sehr wenige UDZ eine Ontologie. (Weil et al. 2023) Werden ähnliche Datenelemente unterschiedlich gespeichert, führt das zu erheblichen Problemen. So kann beispielsweise die ID eines Objekts in einem System als numerischer Wert und in einem anderen System als alphanumerischer Code gespeichert werden. Diese Unterschiede erschweren das Verknüpfen verschiedener Datenquellen. (Rozony et al. 2024)

Die Liste an Herausforderungen im Zusammenhang mit dem Datenmanagement von UDZ ist lang und könnte hier noch weiter ausgeführt werden. Bei den, in diesem Kapitel erwähnten Problemen handelt es sich um die häufigsten Komplikationen. Auf weitere Herausforderungen, wie etwa im Bereich der Datensicherheit oder der Rechenkapazität, wird in diesem Kapitel nicht näher eingegangen.

3.3 Simulation & Analyse

Vergangene und aktuelle Objekte und Prozesse können mit einem UDZ sehr gut dargestellt werden und helfen, die Stadt besser zu verstehen. Die große Stärke von UDZ liegt jedoch in der Fähigkeit, wertvolle Informationen über zukünftige Prozesse darzustellen. Dies geschieht oft, indem mehrere Szenarien simuliert und miteinander verglichen werden. Viele Projekte nutzen sogenannte „Was-wäre-wenn?“-Szenarien, um

Auswirkungen bestimmter Bedingungen zu testen. Die Einsatzgebiete solcher Simulationen und Szenarien sind sehr vielfältig. (Shahat et al. 2021) In diesem Kapitel werden verschiedene Modelle und Anwendungsfälle beschrieben. Außerdem wird auf die Probleme und Herausforderungen in diesem Bereich eingegangen.

3.3.1 Prognosemodelle

Unter diese Kategorie fallen viele verschiedene Modelle von UDZ. Ein sehr häufiges Prognosemodell ist die Erkennung von Verkehrsmustern und Ableitung von Prognosen über Fahrtzeiten und mögliche Staubildungen. Kusic et al. (Kusic et al. 2023) entwickelten ein Modell, das mit Live-Daten von Verkehrssensoren umgehen kann. Dafür verwenden sie die SUMO-Software (Simulation of Urban Mobility). Als Eingangsdaten dienen die Sensordaten eines Autobahnabschnitts bei Genf. Das Ergebnis ist ein DZ für diesen Bereich der Autobahn. Die Ergebnisse zeigen, dass die SUMO-Software auch für Live-Daten sehr gut funktioniert. Um die Abläufe im DZ noch näher an die reale Welt zu bringen, können weitere Sensoren und Daten in das Modell eingebunden werden. Dazu zählen Wetterdaten oder Informationen über Geschwindigkeitsbeschränkungen. Werden diese Anpassungen vorgenommen, können genaue Prognosen erstellt werden. Der große Nachteil solcher Projekte ist der begrenzte Maßstab. Um großflächigere Prognosen durchführen zu können, müssen wesentlich mehr Sensoren eingebunden und Daten verarbeitet werden. Auch im Bereich Umwelt gibt es diverse Prognosemodelle. So kann beispielsweise die Auswirkung eines geplanten Bauprojekts auf die Umgebung simuliert werden. Dazu zählen etwa Wind- und Temperatursimulationen. Als Eingangsdaten dienen aktuelle und historische Wetter- und Klimainformationen. Dazu ist eine hohe Rechenleistung notwendig. Die deutsche Stadt Herrenberg verwendete die Open-Source-Software OpenFOAM (The OpenFOAM Foundation 2026) um Windsimulationen durchzuführen. OpenFOAM ermöglicht es, die Windströme im 3D-Modell zu simulieren. Bei neuen Bauprojekten oder Veränderungen im Erscheinungsbild der Stadt, können diese Projekte in das 3D-Modell eingefügt werden. Dadurch kann man die Auswirkung der Veränderung auf die Windströme in der Stadt simulieren und prognostizieren. (Dembski et al. 2020) Weitere Anwendungsgebiete sind die Simulation des Fußgängerverkehrs (Gholami et al. 2022), die Prognose von Grundstückspreisen (Ying et al. 2023), die Simulation von Hochwasserereignissen (Ge und Qin 2025) oder Vorhersagen von Änderungen im Elektrizitätsnetz (Jafari et al. 2023).

3.3.2 Szenarien & Warnsysteme

Wie der vorherige Abschnitt zeigt, können in einem UDZ Prognosen über zukünftige Ereignisse erstellt werden. Diese Prognosen können entweder mit dem aktuellen UDZ (Status quo) oder einem abgeänderten Modell erfolgen. Ein abgeändertes Modell entsteht, indem man Objekte entfernt, ergänzt, abändert oder andere Parameter editiert. Mit diesen Eingriffen können verschiedene Szenarien erstellt und miteinander verglichen werden. Eine weitere Möglichkeit ist, das Ergebnis der Prognosen für Warnsysteme zu

verwenden. Der Blick in die Zukunft ermöglicht es, Gefahren frühzeitig zu erkennen und die betroffene Bevölkerung zu warnen. (Jafari et al. 2023)

3.3.3 Decision-Support

Wie bereits in 3.1.2 kurz behandelt, ist ein UDZ in der Lage, die kurzfristige urbane Planung wesentlich zu erleichtern. Neben einer guten Kommunikation und Synchronisation zwischen Datenquellen und Modell sind hierbei auch sogenannte "Decision-Support-Systeme" von großer Bedeutung. In diesem Bereich kann künstliche Intelligenz sehr hilfreich sein. KI-Systeme können den Menschen als Entscheidungsträger ersetzen oder als Assistent dienen. Aktuell ist die Rolle als Assistent realistischer, da KI-Anwendungen in diesem Bereich noch nicht sehr lange im Einsatz sind und deshalb fehlerhafte Schlussfolgerungen gezogen werden können. Eine Entscheidung könnte wie folgt ablaufen. Der KI-Algorithmus erhält Daten eines Objekts und leitet daraus eine Handlungsempfehlung ab. Dies wird von der KI genau begründet. Außerdem werden mögliche Alternativen erwähnt und bewertet. Die endgültige Entscheidung trifft dann ein Experte. Durch die stetige Kommunikation zwischen Entscheidungsträger, UDZ und realem Objekt entsteht ein Kreislauf von Verbesserungen. Eine der größten Herausforderungen ist, wie in vielen anderen Bereichen, die enorme Datenmenge, die für diesen stetigen Verbesserungsprozess notwendig ist. (Weil et al. 2023)

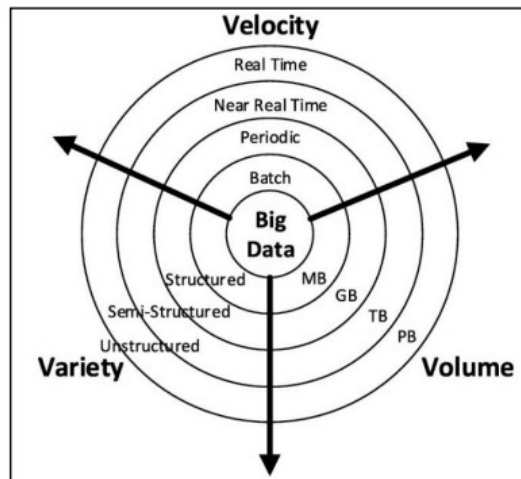
3.3.4 Probleme und Herausforderungen

Eine Stadt besteht aus vielen Prozessen. Manche davon lassen sich wesentlich leichter vorhersagen und analysieren als andere. Die meisten in dieser Arbeit erwähnten Studien beschäftigen sich mit der Simulation von physikalischen Prozessen. Physikalische Prognosen sind deutlich leichter zu treffen als beispielsweise soziale oder ökonomische Vorhersagen. Vor allem der Blick in die Zukunft, was soziale Prozesse betrifft, ist sehr schwierig. Laut Batty (2024) blicken viele soziale Analysen nicht weiter als drei Tage in die Zukunft. Doch gerade diese Analysen können von sehr großer Bedeutung sein. (Batty 2024)

Eine weitere Herausforderung ist die Analyse von Interaktionen dritter Ordnung. Dabei handelt es sich um Interaktionen zwischen Interaktionen. Dazu zählen beispielsweise multimodale Verkehrsmuster, also Bewegungen mit verschiedenen Transportmitteln. Die Vielzahl an Möglichkeiten und Verbindungen zwischen Punkten führt zu einer Überproportionalität. Das heißt, dass die Rechenzeit bei einer Verdoppelung der Datenmenge um mehr als das Doppelte steigt. (Batty 2024)

Die wohl größte Herausforderung ist die Skalierbarkeit. Der steigende Umfang (**Volume**), die wachsende Geschwindigkeit (**Velocity**) und die immer größer werdende Vielfalt (**Variety**) der Daten erschweren auch bei der Simulation und Prognoseerstellung in UDZ die Arbeitsabläufe. Man spricht von den "3Vs" (Abbildung 18). Da die Daten exponentiell wachsen, sind Programme und Algorithmen notwendig, die skalierbar prozessieren

können. Vor allem bei Echtzeit-Analysen wird die Skalierbarkeit zu einem großen Problem. Die Open-Source-Frameworks Hadoop und Spark versuchen durch verteilte Datenverarbeitung, dieses Problem zu lösen. Allerdings sind weitere Verbesserungen in diesem Bereich notwendig, da die “3Vs” der Eingangsdaten stetig ansteigen. (Rozony et al. 2024; Varlamis et al. 2023)



Source: Sabri et al. (2020)

Abbildung 18: 3Vs von Big Data (Rozony et al. 2024)

3.4 Visualisierung

Die Darstellung von UDZ spielt eine wichtige Rolle, um komplexe, urbane Systeme verständlich, zugänglich und interaktiv zu präsentieren. Während die Datenintegration und Datenanalyse das Herzstück des UDZ bilden, ist die Visualisierung die Verknüpfung zu den verschiedenen Stakeholdern und Entscheidungsträgern. Durch die Erweiterung von 2D- und 3D-Modellen um verschiedene Applikationen entsteht ein UDZ, der für diverse urbane Aufgabenbereiche eingesetzt werden kann. Shahat et al. (2021) betonen die Wichtigkeit einer effektiven Visualisierung, um die Struktur und die Prozesse innerhalb einer Stadt verständlich zu machen und die zukünftige Entwicklung des urbanen Raums zu planen. Der folgende Abschnitt beschreibt verschiedene Ansätze, um einen UDZ zu visualisieren.

3.4.1 3D-Stadtmodelle

3D-Modelle bilden sehr oft die Basis für einen UDZ. Ein 3D-Modell an sich ist jedoch noch kein UDZ. Es ist lediglich eine Darstellung der Stadt in einem bestimmten Detailgrad und zu einem bestimmten Zeitpunkt.

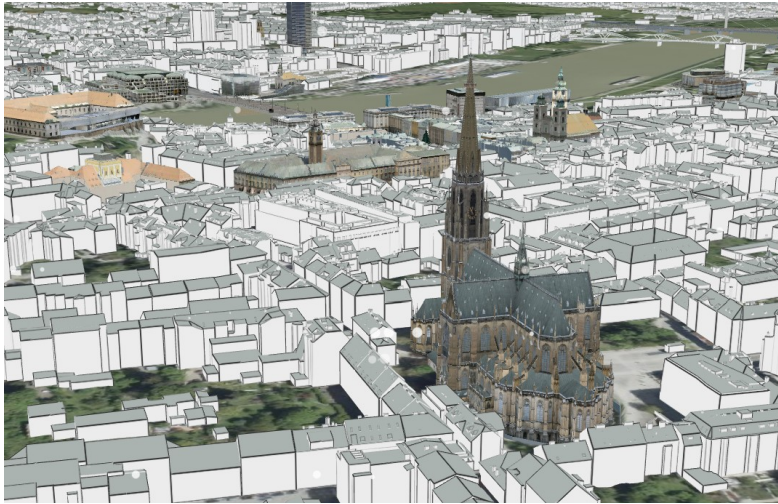


Abbildung 19: 3D-Gebäudemodell der Stadt Linz (www.3dlinz.at)



Abbildung 20: 3D-Mesh-Modell der Stadt Linz (www.3dlinz.at)

Viele 3D-Modelle werden nur für reine Visualisierungszwecke erstellt. Oft handelt es sich dabei um Mesh-Modelle. Diese Modelle können die Stadt sehr genau visualisieren. Allerdings sind sie geometrisch unstrukturiert und können nur sehr schwer mit semantischen Daten verknüpft werden. Es gibt dazu Versuche und erste Erfolge, jedoch meist nur für eine bestimmte Auswahl an semantischen Informationen. Will man komplexe räumliche Analysen und Simulationen durchführen, braucht man semantische Modelle. Diese lassen sich einfacher mit geometrisch strukturierten 3D-Modellen generieren. Werden semantische Daten mit einem Stadtmodell verknüpft, spricht man oft von einem City Information Model (CIM). (Lehner und Dorffner 2020; Senyurdusev et al. 2020)

3.4.1.1 Level-of-Detail (LoD)

3D-Modelle können starke Unterschiede in der Genauigkeit aufweisen. Die Genauigkeit wird meist in den sogenannten LoD-Klassen angegeben. Je höher die LoD-Zahl, desto genauer wird die Realität dargestellt. Je nach Anwendung werden oft sehr unterschiedliche LoDs verwendet. Dies liegt daran, dass manche Simulationen bei zu

genauen 3D-Modellen aufgrund der hohen Rechenzeit nicht durchführbar sind. Ein Beispiel dafür ist die Simulation von Geräuschverschmutzung. Hier kann maximal ein Modell mit LoD1 verwendet werden. Im Optimalfall kann ein UZ zwischen verschiedenen Genauigkeitsstufen hin- und herspringen. (Lehner und Dorffner 2020)

Im Bereich der 3D-Stadtmodelle ist das LoD-Konzept von CityGML, das am weitesten verbreitete LoD-Konzept. Es gibt jedoch Kritik an der Abstufung von CityGML. Biljecki et al. (Biljecki et al. 2016) haben das Konzept weiter verfeinert, da sie die bisherige Abstufung für zu ungenau hielten. Das Konzept von Biljecki et al. unterteilt jede LoD-Stufe in vier Unterstufen. Im Konzept von CityGML gibt es vier Stufen, die die äußere Genauigkeit eines Gebäudes definieren. Die Unterteilung von Biljecki et al. umfasst 16 verschiedene Genauigkeiten. Lehner et al. (2020) haben diese Abstufung um eine weitere Unterteilung je LoD-Stufe ergänzt. Die zusätzliche Spalte umfasst die Modellierung von Überhängen, in der Grundform des Gebäudes. Überhänge von Dachflächen sind bereits Bestandteil der Gliederung von Biljecki et al. (2016) (Abbildung 21).

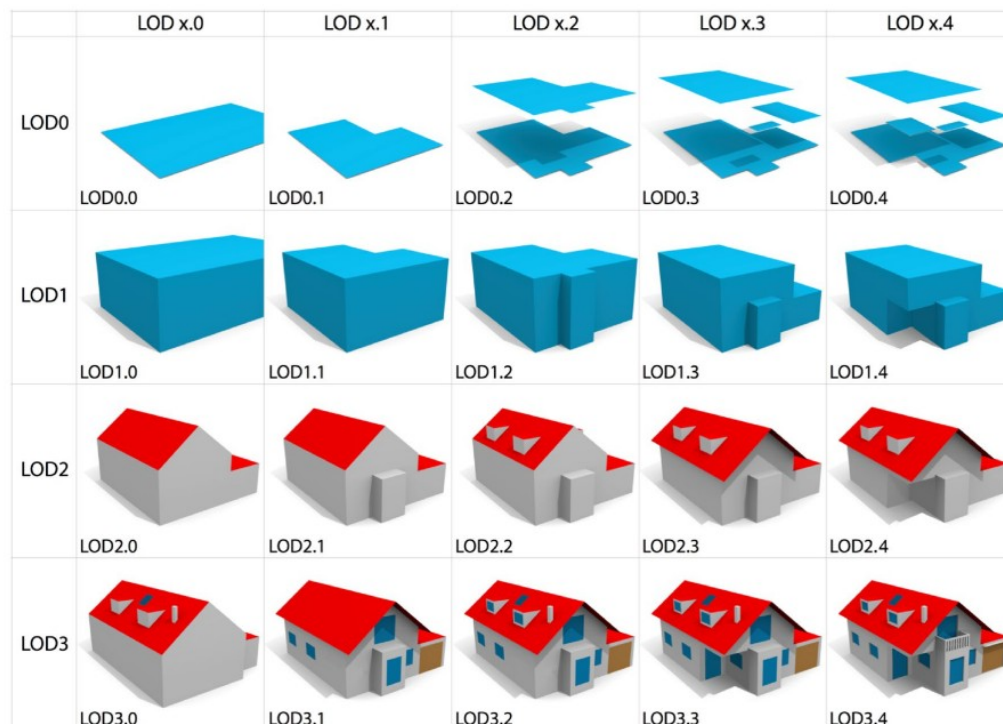


Abbildung 21: LODs für ein Wohngebäude (Biljecki et al. 2016)

Außerdem legten Biljecki et al. (2016) Höhenreferenzen fest, die für die Modellierung von Gebäuden im LoD1 verwendet werden. Diese Höhenreferenzen (H0-H6) erklären, welche Höhe für die höchste Fläche eines Gebäudes verwendet wird. Wichtig anzumerken ist, dass das LoD1-Modell ein sogenanntes Klötzchen-Modell ist. Das heißt, dass es nur ebene Flächen gibt. Somit wird jedes Dach als Flachdach modelliert. Abbildung 22 zeigt die verschiedenen Referenzen. Bei der Betrachtung dieser Tabelle wird klar, dass die Auswahl der Höhenreferenz große Auswirkungen auf die Darstellung

der Gebäude sowie auf spätere Analysen und Simulationen haben kann. Auch bei der Darstellung der Gebäude-Footprints gibt es verschiedene Varianten.

Table 1 List of vertical geometric references (representations for the height of the top surface of the LOD1 block model)

	Height at	INSPIRE <i>verticalGeometry-Reference3DTop</i>	References Biljecki et al. (2016b)
<i>H0</i>	Roof edges	generalRoofEdge	=
<i>H1</i>	Roof eaves	generalEave	=
<i>H2</i>	One-third of the roof height	generalRoof	oneThirdRoof
<i>H3</i>	Half of the roof height	generalRoof	halfRoof
<i>H4</i>	Two-third of the roof height	generalRoof	twoThirdRoof
<i>H5</i>	Top of the roof (i.e. ridge)	topOfConstruction	=
<i>H6</i>	Highest point of the building	highestPoint	=

Biljecki et al. (2016b) uses the equal sign to state that the reference is re-used from INSPIRE

Abbildung 22: Höhenreferenzen für ein LOD1 Gebäudemodell (Biljecki et al. 2016)

Dieser Aspekt betrifft nicht nur LoD1, sondern LoD0 bis LoD3. Denn überhängende Dächer werden nur in manchen Abstufungen (nach Biljecki 2016) von LoD3 und LoD4 dargestellt. Der Footprint kann auf Grundlage des tatsächlichen Gebäudes, sprich der bebauten Fläche, modelliert werden. Alternativ kann man alle Flächen, die bei stereographischer Draufsicht von Dachflächen bedeckt sind, als Footprint zusammenfassen. Diese Variante führt bei überhängenden Dachflächen zu einem größeren Abdruck. Die dritte Variante basiert auf der zweiten Variante. Sie verwendet einen Versatz, um die Fläche von Variante 2 zu verkleinern. Oft ergibt sich die Darstellung des Gebäude-Footprints automatisch aus den vorhandenen Daten. Sind die Außenwände der Gebäude nicht vermessen worden oder nicht auf Luftbildern erkennbar, muss man eine der beiden letzteren Varianten wählen. (Lehner und Dorffner 2020)

Die Auswahl einer LoD-Stufe für einen Digitalen Zwilling hängt in erster Linie von den vorhandenen Daten ab. Aus diesen Daten ergibt sich die maximale Genauigkeit des Modells. Ist diese maximale LoD-Stufe bekannt, kann man für unterschiedliche Anwendungen die Genauigkeit herabstufen. Sehr oft weicht die maximale Genauigkeit innerhalb einer Stadt stark voneinander ab. In Wien sind beispielsweise bestimmte Gebäude in LoD 3.4 vorhanden. Die gesamte Innere Stadt ist in einem Detailgrad von LoD 2.4 verfügbar. Flächendeckend für ganz Wien ist jedoch nur ein Modell in LoD 2.1 verfügbar. Lehner et al. 2020 sind der Meinung, dass ein LoD2.4-Modell die optimale Wahl für einen UDZ ist. Denn dieses Modell kann sehr gut in niedrigere Stufen vereinfacht und mit Ergänzungen in höhere LoD-Stufen überführt werden. Da ein Stadtmodell aus unterschiedlichen Datengrundlagen besteht, sollte neben dem LoD auch ein Level of Accuracy (LoA) für die Gebäude angegeben werden. (Lehner und Dorffner 2020; Caldarelli et al. 2023)

3.4.1.2 Erstellung

Wie bereits in Kapitel 3.4.1 beschrieben, können 3D-Punktwolken eine Grundlage für die Erstellung von 3D-Stadtmodellen sein. Um aus einer Punktwolke ein dreidimensionales Stadtmodell zu erstellen, sind mehrere Schritte notwendig. Dazu zählen: Punktklassifizierung, Punktsegmentierung, Formerkennung und Objektklassifizierung. Diese werden heutzutage oft mittels Machine Learning umgesetzt. Bei der Punktklassifizierung wird jeder Punkt einer der vorab festgelegten Kategorien zugewiesen. Um diesen Schritt mithilfe von ML durchführen zu können, muss ein Trainingsdatensatz angelegt werden. In diesem werden einzelne Punkte händisch kategorisiert. Die Punktsegmentierung zerteilt die gesamte Punktwolke in kleinere Punktwolken, um einzelne Objekte zu isolieren. Normalerweise müssen in diesem Schritt Ecken und Kanten gefunden werden. Dank Machine- und Deep Learning (DL) können jedoch auch semantische Informationen für die Segmentierung verwendet werden. Dafür werden beispielsweise Convolutional Neural Networks (CNN) aufgebaut. Im nächsten Schritt werden die einzelnen Segmente in geometrische Formen umgewandelt. Nachdem die Objekte isoliert und in geometrische Formen umgewandelt wurden, erfolgt die Klassifizierung. (Lehner und Dorffner 2020)

Nys et al. (2020) haben einen unüberwachten Segmentierungsalgorithmus verwendet, um ebene Flächen zu isolieren. Im Gegensatz zu vielen anderen Methoden arbeiten sie mit Daten im CityJSON-Format. Abbildung 23 zeigt das Ergebnis.



Abbildung 23: Ergebnis einer unüberwachten Gebäudesegmentierung (Nys et al. 2020)

Dieser Algorithmus, namens RANSAC (RANDOM SAMPLE CONSENSUS) wurde zwar ursprünglich für die Segmentierung von Punktwolken im Inneren von Gebäuden entwickelt, liefert aber auch bei großmaßstäbigeren Anwendungen brauchbare Resultate. Nys et al. (2020) testeten zwei verschiedene Varianten, um mit RANSAC Punktwolken zu segmentieren. Bei der ersten Variante werden dem Algorithmus einige Parameter mitgeteilt. Dazu zählt beispielsweise die Mindestanzahl an Punkten, die für eine ebene Fläche notwendig ist. Die zweite Variante erstellt Cluster anhand gemeinsamer intrinsischer oder assimilierter Attribute. Intrinsische Attribute sind gemessene Informationen. Dazu zählt neben den Punktkoordinaten auch die Intensität oder der Scan-Winkel. Assimilierte Attribute werden berechnet. Beispiele hierfür sind die Punktdichte, die Höhe über dem Boden oder, falls vorhanden, ein RGB-Wert. Beispielsweise. Anschließend versucht RANSAC, die geclusterten Punkte zu segmentieren. Die Qualität

der beiden Methoden ist sehr ähnlich. Nur in der Berechnungszeit schneidet die zweite Methode deutlich besser ab. Denn RANSAC braucht 58 % länger, um die nicht geclusterten Punkte zu segmentieren. Das Clustern vorab, benötigt nicht viel Zeit. Die Fehler treten bei beiden Varianten in denselben Gebieten auf. Daraus schließen Nys et al., dass die Dichte oder die Verteilung der Punkte in manchen Bereichen die Ursachen für die Qualitätseinbuße sind. Aus den einzelnen Segmenten kann die jeweilige räumliche Ausdehnung als rechteckige Bounding-Box berechnet werden. Aus der Topologie der Flächen können dann Dachflächen abgeleitet werden. Außerdem werden die Footprints der Gebäude berechnet. Aus dem Footprint und den Dachflächen wird anschließend das Gebäude erstellt. Wichtig anzumerken ist, dass für diese Art der Gebäudeerstellung die Segmentierung eine wesentliche Rolle spielt. Wird eine Dachfläche nicht als Segment erkannt, wird daraus keine Fläche berechnet und das Gebäude weist eine Lücke auf. Die Methode von Nys et al. funktioniert aber nur für Gebäude. Der nächste Schritt wäre, auch Brücken, Statuen oder Straßen auf diese Art und Weise zu erstellen. (Nys et al. 2020)

Alternativ können 3D-Modelle auch aus 3D-Mesh-Modellen erstellt werden. Seit einigen Jahren wird in diesem Bereich auch künstliche Intelligenz eingesetzt. Feng et al. (2019) entwickelten MeshNet, um aus 3D-Meshes 3D-Gebäude zu generieren. Li et al. (2021) entwickelten DNF-Net, um Fehler in Mesh-Modellen zu erkennen und auszubessern. (Liu et al. 2023) Diese Fehler entstehen entweder beim Scannen oder beim Modellieren der Daten. Die Grundidee von DNF-Net ist es, Abnormalitäten in der Richtung von Oberflächen des Meshes zu erkennen. Dazu werden Muster in der Richtung in kleinem und größerem Maßstab untersucht. Im ersten Schritt wird das Mesh-Modell in viele kleine, sich überlappende Meshes unterteilt. In jedem kleinen Mesh wird die Richtung der Fläche erkannt und gespeichert. Anschließend werden die Meshes in verschiedenen Maßstäben untersucht und fehlerhafte Features erkannt und entfernt. Abschließend wird das Mesh-Modell rekonstruiert, um wieder natürlich auszusehen. Wichtig ist, dass scharfe Kanten bestehen bleiben. Im Gegensatz zu klassischen Methoden benötigt dieser Ansatz keine händisch festgelegten Regeln oder Selektionen. (Li et al. 2021)

Es gibt auch Versuche, die beiden Datengrundlagen gemeinsam für die Erstellung von geometrisch korrekten Gebäudemodellen zu verwenden. Liu et al. (2023) verwenden eine Deep-Learning-Methode, um 3D-Punktwolken und 3D-Mesh-Modelle zu fusionieren und daraus Gebäude zu erstellen. Zuerst erstellen sie ein Evaluierungs-Netzwerk, das die Qualität der Eingabedaten bewertet. Dieses Netzwerk wird anschließend verwendet, um die Bestandteile und Formen der Gebäude zu extrahieren. Für das Training der Methode wurde ein 3D-Datensatz aus mobilem Laserscanning und 3D-Mesh-Daten erstellt, der Paare aus übereinstimmenden Meshes und Punktwolken für das Gebäudemodell enthält. Der Vorteil dieses Ansatzes liegt in der Kombination der Stärken von Mesh-Modellen und Punktwolken. Die Stärke von Mesh-Modellen liegt im reichen Informationsgehalt. Im Gegensatz zu Punktwolken enthalten sie Flächen, Texturen und

Nachbarschaftsbeziehungen. Punktwolken hingegen sind die rohe Form der Daten und sind geometrisch robuster als Mesh-Modelle. (Liu et al. 2023)

3.4.2 4D-Modelle (zeitbasierte Simulation)

Im Vergleich zu 3D-Modellen haben 4D-Modelle eine Zeitsequenz als zusätzliches Attribut. Durch dieses Attribut können räumliche Objekte, bei denen sich die Eigenschaften und Informationen über die Zeit ändern, gespeichert, analysiert und dargestellt werden. (Wang et al. 2019b) Jedes Objekt erhält einen Zeitstempel, der den Beginn und das Ende der Gültigkeit angibt. Diese Methode ist besser als sogenannte Snapshots, bei denen zu einem beliebigen Zeitpunkt Daten gespeichert werden. Der Nachteil von Snapshots ist, dass Objekte mehrfach geführt werden und man nicht genau sagen kann, wann sich etwas verändert hat, da die Veränderung zu jedem Zeitpunkt zwischen zwei Snapshots stattgefunden haben kann. Europäische Städte wie etwa Rotterdam, Hamburg oder Prag haben bereits 4D-Stadtmodelle, die einen Blick in die Vergangenheit ermöglichen und Verständnis für urbane Prozesse im Laufe der Zeit schaffen. (Ohori et al. 2015; Kocaman et al. 2024)

Unter 4D-Modellen wird manchmal auch etwas anderes verstanden. Manche Publikationen sprechen von 4D-Modellen, wenn es sich um ein 3D-Modell mit verschiedenen LoD-Stufen handelt. Also anstatt, ein Objekt zu verschiedenen Zeitpunkten darzustellen, kann es in verschiedenen Detailstufen abgebildet werden. (Ohori et al. 2015)

3.4.3 Erreichbarkeit

UDZ werden meistens als 3D- oder 4D-Modelle visualisiert. Für den Großteil der Bevölkerung ist das eine sehr ansprechende und leicht zugängliche Visualisierung. Dennoch gibt es bestimmte Gruppen, die durch diese Darstellungsform ausgeschlossen werden oder nur schwer Zugang dazu finden. Caporusso et al. (Caporusso et al. 2020) entwickelten die Idee des IDT (Interface Digital Twin), dessen Ziel es ist, die Oberflächen von Websites für blinde Personen nachzubauen. Ob dieser Ansatz allerdings auch bei UDZ eingesetzt werden kann, ist sehr fraglich. Neben blinden Personen werden auch Menschen mit anderen körperlichen oder geistigen Beeinträchtigungen, Personen mit niedriger digitaler Bildung und Bevölkerungsgruppen ohne Zugang zur nötigen Soft- oder Hardware oft ausgegrenzt. Coluccino et al. (2025) sprechen in diesem Zusammenhang von der “unsichtbaren Bevölkerung”. Bei der Gestaltung muss jedoch auf diese Personen Rücksicht genommen werden, damit ein UDZ seine Rolle als demokratisches Instrument erfüllen kann. Veröffentlichungen zu konkreten Lösungsansätzen sind bislang kaum vorhanden. (Coluccino et al. 2026; Sarbout et al. 2024)

3.4.4 Austausch- und Streamingformate

Der große Vorteil eines einheitlichen Formats für 3D-Stadtmodelle, besteht darin, dass mehrere Stakeholder und Städte ihre Daten ohne große Probleme miteinander austauschen können. (Weil et al. 2023) Das am weitesten verbreitete Format im Bereich

von 3D-Modellen ist CityGML. CityGML ist ein Open-Source-Datenmodell, das auf dem XML-Format basiert und für das Speichern und Transferieren von 3D-Stadtmodellen entwickelt wurde (Petrova-Antonova und Ilieva 2019). Es wurde vom “Open Geospatial consortium”, kurz OGC, entwickelt (Weil et al. 2023). Neben dem Austausch helfen Formate wie CityGML auch bei der Archivierung von Daten. (Petrova-Antonova und Ilieva 2019) Der Fokus von CityGML liegt auf Objekten in der Stadt. Dazu zählen beispielsweise Gebäude, Straßen, Brücken, Bäume und Gewässer. CityGML ist weit verbreitet und bietet viele Erweiterungen an. (Weil et al. 2023)

Ein weiteres gängiges Austauschformat für Vektordaten in UDZ ist IFC (Industry Foundation Classes). Im Gegensatz zu CityGML liegt der Fokus dieses Formats vor allem auf Gebäuden und ihren unmittelbaren Umgebungen. Man spricht hier auch gerne vom AEC-Bereich. AEC steht für Architecture, Engineering und Construction. IFC-Daten sind meist detaillierter, aber dafür nur selten großflächig verfügbar. In der neuesten Version dieses Formats, wurden zusätzliche Klassen (Straßen, Brücken, Tunnel etc.) ergänzt. Andere Austauschformate für 3D-Daten im UDZ-Bereich sind laut Weil et al. (2023) nicht weit verbreitet oder nur für bestimmte Anwendungsbereiche nützlich. Dazu zählen InfraGML, Green Building XML oder IndoorGML. (Abbildung 24) (Weil et al. 2023)

- Common exchange vector formats in the UDZ context

Format	CityGML	IFC	InfraGML - Landinfra	INTERLIS	LandXML	Green Building XML	IndoorGML
Maintainer / developer	OGC - Open Geospatial consortium	BuildingSMART	OGC - Open Geospatial consortium	Swissopo	LandXML.org Switzerland	Green Building XML organization	OGC - Open Geospatial consortium
Purpose	Representation, storage, and exchange of virtual 3D city models.	Data model and exchange format for building and infrastructure data.	Representation, storage, and exchange of land and civil engineering infrastructure facilities.	Representation and exchange of data.	Standard data exchange format for land development.	Describe the physical and thermal properties of a building.	Framework of representation and exchange of indoor spatial information.
Semantic	✓	✓	✓	✓	✓	✓	✓
Spatial (geometry)	✓	✓	✓	✓	✓	✓	✓
3D geometry	✓	✓	✓	?	×	✓	✓
Focus	Cities objects: Buildings, Roads, Bridges, Vegetation, Water bodies, Transportation, Land use, Relief, ...	Building & AEC (system, components), surrounding site.	Roads, rail ways, and waterways, as well as other types of infrastructure, such as buildings, bridges, and utility networks.	Depends on models.	Alignment, Surfaces, Profiles, Cross-sections, Dams, etc., Earthwork, ...	Building.	Building.
Conceptual Model	Conceptual Model Standard ("OGC City Geography Markup Language (CityGML) Part 1: Conceptual Model Standard,").	Specification of the IFC standard	Conceptual Model Standard ("OGC 15-111r1 - Landinfra Conceptual Model,").	Multiple models (Swiss official data models ("models.gro.admin.ch,").	Schema ("LandXML-2.0,").	Schema ("gbXML, Green Building - Current Schema,").	Schema ("IndoorGML, OGC,").
Extensions	Many extensions (Biljecki et al., 2018) available.	IFC v5 will bring more classes (rail, tunnel, road, bridge).	-	Multiple models	-	-	-
Widely used ?	✓	✓	-	✓, in CH	×	-	?

¹ Some papers mention the usage of 3D geometries within INTERLIS. However, it seems not to be the case in the current version of INTERLIS used in Switzerland.

Abbildung 24: Austauschformate für Vektordaten im UDZ Kontext (Weil et al. 2023)

Neben einem passenden Austauschformat für die Vektordaten ist auch ein Streaming-Format notwendig (Abbildung 25). Mit diesem Format wird der UDZ publiziert und ist für die Zielgruppe, meist Bürger, sichtbar. Weil et al. (2023) erwähnen sechs verschiedene Streaming-Formate. Einige davon werden hier kurz vorgestellt. Das Open Geospatial Consortium (OGC) hat “OGC 3D Tiles” entwickelt. Dieses Dateiformat ermöglicht die Übertragung von sehr großen 3D-Datensätzen. Dazu zählen photogrammetrische Daten, Gebäudemodelle, Punktwolken und BIM/CAD-Daten. Der große Nachteil von “OGC 3D Tiles” ist die fehlende semantische Information. GeoJSON-LD ist in der Lage, semantische Informationen darzustellen. Auch dieses Format wurde vom OGC entwickelt. Allerdings kann man keine 3D-Geometrien mit GeoJSON-LD darstellen. Deshalb hat das OGC das Format CityJSON entwickelt. Dieses Format soll eine leichtgewichtige Alternative zu CityGML sein. Wie CityGML liegt der Fokus dieses

Formats auf urbanen Objekten (Gebäude, Straßen, Brücken, etc.). Eine weitere Variante 3D-Daten zu präsentieren ist ein KML Network Link. Auch dieses Format wurde vom OGC entwickelt und verwendet eine spezielle Form der KML-Sprache namens KMZ. (Weil et al. 2023)

- Common streaming formats in the UDT context

Format	OGC 3D Tiles	Indexed 3D Scene Layers (I3S)	GeoJSON-LD	CityJSON	KML Network Link	GL Transmission Format (gltf)
Maintainer / developer	OGC - Open Geospatial consortium	OGC - Open Geospatial consortium Created by ESRI	OGC - Open Geospatial consortium	OGC - Open Geospatial consortium	OGC - Open Geospatial consortium	Khronos Group ("The Khronos Group," 2023)
Purpose	Streaming and rendering massive 3D geospatial (Photogrammetry, 3D Buildings, BIM/CAD, Instanced Features, and Point Clouds).	Streaming and storing massive amounts of geospatial content.	Lightweight Linked data format. Allows for the use of semantic web vocabulary and concepts, such as linked data URIs and RDF properties, in the encoding of geographic information.	Alternative to the CityGML standard and is designed to provide a more lightweight and efficient representation of 3D city models.	A KML network link is a way of publishing geographic data on your server using a compressed form of Keyhole Markup Language called KMZ.	Efficient transmission and loading of 3D scenes and models by applications. glTF minimizes both the size of 3D assets, and the runtime processing needed to unpack and use those assets. Mainly used in gaming and VR.
Semantics Spatial (geometry)	×	×	✓	✓	✓	?
3D geometry Focus	Multiple.	Multiple. The latest Version, I3S OGC 1.3, adds support for Building Scene Layers ("Building Viewer," 2023) (BSL).	Multiple.	Buildings, Roads, Bridges, Vegetation, Water bodies, Transportation, Land use, Relief, ...	Geographic visualization, including annotation of maps and images.	✓
Conceptual Model	-	-	-	CityJSON Specifications ("CityJSON Specifications 1.1.3,").	Schema ("Index of /kml,").	-
Extensions	-	-	-	Many extensions available ("Extensions,").	-	-
Link	3D Tiles Specification ("3D Tiles Specification,").	Format Community Standard ("Indexed 3D Scene Layers (I3S),").	geojson ("GeoJSON-LD,").	cityjson.org ("Home,").	Specification ("KML,").	Specification ("Khronos glTF Registry - The Khronos Group Inc,").

Abbildung 25: Streaming-Formate im UDZ Kontext (Weil et al. 2023)

3.4.5 Cloubasierte Plattformen

Es gibt bereits einige cloubasierte Lösungen, die für Smart Cities entwickelt wurden. Dazu zählt beispielsweise *Smart World Pro*. Dieses Tool verbindet das 3D-Modell einer Stadt mit Gebäude- und Geoinformationen. Auch Sensordaten und IoT-Geräte können mit dem Modell verknüpft werden. Alternativen zu Smart World Pro sind *Open Cities* und *Platform of Trust*. Vor allem die Zusammenarbeit mehrere Stakeholder, vereinfacht die Erstellung solcher cloubasierter Plattformen. (Hämäläinen 2020) Zu den größten Vorteilen von cloubasierten Lösungen zählen die Interoperabilität und Ortsflexibilität. Am effektivsten wären Städteübergreifende Lösungen, zum Beispiel für ganz Deutschland. Allerdings ist hier das Problem, dass aktuelle Zwillinge verschiedene Cloudlösungen verwenden und die Fusion dieser Systeme sehr aufwendig ist. Das Hosten der Anwendung auf eigener IT-Infrastruktur ist für die meisten Städte und Kommunen aus finanziellen und personaltechnischen Gründen nicht realisierbar. Deshalb setzen die meisten Projekte auf cloubasierte Lösungen, womit sie einen Teil der Arbeit auslagern. (Tegtmeyer et al. 2022) Der große Nachteil von cloubasierten Lösungen ist, dass man die Kontrolle über die Daten zum Teil abgibt. Im schlimmsten Fall verliert man selbst den Zugang zu den eigenen Daten. Auch in Bezug auf die Sicherheit der Daten muss man sich auf den Anbieter des Cloud-Speichers verlassen. Bei der Auslagerung von Daten oder Anwendungen für einen UDZ sollte die Cloud-Lösung auf alle Fälle intensiv auf mögliche Sicherheitsrisiken überprüft werden. (Syed et al. 2020)

3.4.6 Virtuelle Realität

UDZ können helfen, das Mitwirken von Bürgern zu erleichtern. (White et al. 2021) Die Darstellung von UDZ in virtuellen Realitäten ist eine Möglichkeit, die Kommunikation zwischen verschiedenen Stakeholdern zu erleichtern. Durch VR können Ergebnisse auf leichter zu verstehende Weise präsentiert werden. Dadurch kann man Zielgruppen erreichen, die mit traditionellen Darstellungsformen nur schwer zu erreichen sind. Dazu zählen Kinder, Teenager oder Gruppen mit niedriger Bildung. Auch Sprachbarrieren können durch diese Art der Präsentation überwunden werden. UDZ können die Auswirkungen von geplanten Projekten simulieren. In Kombination mit VR werden diese Konsequenzen für fast alle Stakeholder realitätsnah dargestellt. Gelingt es, vorab so viele Stakeholder wie möglich in die Planung einzubeziehen, sinkt das Risiko negativer Reaktionen während oder nach der Umsetzung der Projekte. (Dembski et al. 2020)



Abbildung 26: Miteinbeziehen von Bürgern in den Planungsprozess mittels VR (Dembski et al. 2020)

Ein Weg, um die jüngere Bevölkerung in Planungsprozesse einzubeziehen, ist das weltweit beliebte Computerspiel "Minecraft". In der Welt von Minecraft besteht alles aus verschiedenen Würfeln. Städte können ihren UDZ in eine eigene Minecraft-Welt umwandeln. In dieser können junge, aber auch ältere Menschen ihre Ideen für den urbanen Raum spielerisch umsetzen. Diese Art der Bevölkerungspartizipation wurde bereits in mehreren Städten eingesetzt (z. B. Zürich). (Schrotter und Hürzeler 2020)

3.4.7 Probleme und Herausforderungen

Eine effektive Visualisierung ist essenziell, um komplexe urbane Systeme verständlich und interaktiv zu präsentieren. Zu den größten Problemen bei der Visualisierung zählt einmal mehr die Heterogenität der verschiedenen Daten. Von 2D-Karten bis zu 3D-Punktwolken und semantischen Informationen, soll alles in einem Modell zusammengefügt werden. Weiters variiert der optimale Detaillierungsgrad des drei- oder

vierdimensionalen Modells stark von Anwendung zu Anwendung. Manche Analysen sind mit zu hohem LoD nicht durchführbar, andere wiederum bei zu niedriger Genauigkeit nicht aussagekräftig. Auch das Anreichern mit semantischen Informationen ist schwierig, da zuerst ein geometrisch korrektes Modell erstellt werden muss. Hinzukommen Herausforderungen in Bezug auf ausgegrenzte Bevölkerungsgruppen und die Speicherung des UDZ.

Neben den bereits genannten Problemen ist der Mangel an qualifizierten Arbeitskräften und Expertise im Bereich von UDZ ein weiteres Problem. Die Erstellung, Inbetriebnahme und Fortführung eines umfangreichen UDZ erfordert viele technische Fähigkeiten. Doch Entwickler und Ingenieure im Bereich der Infrastruktur sind rar, und die Interdisziplinarität bei einem UDZ erfordert zusätzliches Wissen und einen gut organisierten Austausch zwischen Experten. Diese Umstände führen oft dazu, dass Städte die Entwicklung und das Management von UDZ auslagern und extern vergeben. Das führt jedoch meist zu hohen Kosten und zu einem zu geringen Mitwirken der lokalen Stakeholder. Des Weiteren wird dann das Problem der mangelnden qualifizierten Arbeitskraft nur an Technologieunternehmen weitergegeben. Außerdem fehlt es an guten Austausch- und Lernplattformen, um Entwicklern das nötige Know-how beizubringen. Neben den hohen Erstellungskosten sind auch die Wartungskosten eines UDZ beträchtlich. Vor allem bei zuverlässigen UDZ mit Echtzeit-Funktionen sind die laufenden Kosten sehr hoch. Da die Auswirkungen von UDZ auf die mittel- und langfristigen Kosten einer Stadt noch nicht abschätzbar sind, ist noch unklar, wie rentabel UDZ sind. Effiziente UDZ-Systeme und Komponenten haben meist höhere Anschaffungskosten und erfordern mehr Knowhow als günstigere Alternativen. Deshalb sind sie oft schwer zu vermarkten und werden von vielen Städten nicht eingesetzt. Die Herausforderungen in diesem Bereich erklären, warum viele Wissenschaftler einheitliche Standards und Methoden für UDZ fordern. Eine einheitliche Vorgehensweise erleichtert den Wissenstransfer und den Einstieg für neue Entwickler. Außerdem können dadurch Entwicklungskosten reduziert werden. (Weil et al. 2023)

3.5 Baustein-Konzept

Immer mehr Städte und Regionen befassen sich mit UDZ und verwenden diese für unterschiedliche Anwendungsgebiete. Zu den häufigsten Anwendungsthemen zählen das Transportwesen, das Energiewesen, die Umwelt und die Stadtplanung. Bislang vernachlässigte Themen sind das Gesundheitswesen, die Bildung und die Wirtschaft. Meistens beschränken sich einzelne UDZ auf ein oder wenige Anwendungsgebiete, da eine Spiegelung der Realität in mehrere Bereiche noch nicht umsetzbar ist. (Wang et al. 2023b)

Einige Wissenschaftler beschäftigten sich bereits mit der Frage, ob es überhaupt sinnvoll beziehungsweise realisierbar ist, einen UDZ mit sehr vielen verschiedenen Anwendungsgebieten zu erstellen. Die Aufteilung in mehrere kleine UDZ ist für Städte

wesentlich leichter zu realisieren. Dadurch können beispielsweise die Herausforderungen der Interoperabilität von Big Data leichter bewältigt werden. Diese Modelle können entweder voneinander isoliert funktionieren oder miteinander verknüpft sein. (Shahat et al. 2021) Die Stadt Linz gliedert den UDZ in mehrere Bausteine. Diese werden unterschiedlich dargestellt und funktionieren isoliert voneinander. In diesem Kapitel werden einige der häufigsten Anwendungsgebiete und Bausteine von UDZ beschrieben.

3.5.1 Transport & Mobilität

UDZ spielen eine bedeutende Rolle bei der Erstellung von sogenannten „Mapping-Services“ wie etwa Google Maps oder TomTom. Diese Technologien benötigen eine regelmäßig aktualisierte digitale Repräsentation der echten Welt. Ein UDZ, der ausreichend Daten über den Verkehr in einer Stadt beinhaltet, kann eine Vielzahl von wichtigen Analysen und Prognosen durchführen und visualisieren. Diese Analysen können verwendet werden, um verschiedene Bereiche im Transportwesen einer Stadt zu verbessern. Dazu zählen beispielsweise, die Straßenraumplanung oder das Parkplatzmanagement. Vor allem im Kontext einer nachhaltigen Stadtentwicklung können durch die Integration von UDZ Verbesserungen erzielt werden. (Alvi et al. 2025)

Jafari et al. (2023) heben einige Anwendungen im Bereich des Transports in ihrer Arbeit hervor. Dazu zählt die Verkehrsvorhersage, die zu einem besseren Verkehrsmanagement beitragen kann. Auch die Analyse von Verhaltensmustern im Verkehr wird erwähnt. Hat man Informationen über das Verhalten der Verkehrsteilnehmer, kann die Sicherheit im Straßenraum verbessert werden. Dasselbe gilt für das Verhalten von Fußgängern. Ein weiteres Anwendungsgebiet ist das Management von Transportsystemen. Auch hier können Kosten reduziert werden. Genau wie Alvi et al. (2025) erwähnen auch Jafari et al. (2023) das Parkplatzmanagement.

Tabelle 2: Anwendungsgebiete im Bereich Transport & Mobilität (Jafari et al. 2023)

Literatur	Anwendung	Motivation
(Ketzler et al. 2020)	Verkehrsprognose	optimales Verkehrsmanagement
(Liu et al. 2020)	Fahrer-Verhaltensanalyse	Sicherheit und Nachhaltigkeit verbessern
(Yan et al. 2022)	Fußgänger-Verhaltensanalyse	Sicherheit verbessern
(Damjanovic-Behrendt 2018)	Erkennung von physischen und virtuellen Angriffen	Sicherheit verbessern
(Maurer 2017)	Transportsystem-Monitoring	Real-time Monitoring und niedrigere Wartungskosten
Vorschlag (Alvi et al. 2025)	Energievorhersage	Optimierung des Energiekonsums
Vorschlag (Alvi et al. 2025)	Parkplatz-Management	Effektiveres Parkplatz-Management

3.5.2 Klimawandel

Im Laufe der letzten Jahrzehnte hat sich das Klima signifikant verändert und Extremwetterereignisse treten immer öfter auf. Um diesen Ereignissen möglichst gut entgegenwirken zu können, sind Simulationen essenziell. Mit einem UDZ können diese extremen Ereignisse teilweise simuliert werden. So haben bereits mehrere Städte Simulationen für Hochwasserereignisse durchgeführt. Mit diesen Simulationen können besonders betroffene Gebiete erkannt und Gegenmaßnahmen in der digitalen Welt getestet werden. Natürlich sind für solche Simulationen und Analysen genaue Daten notwendig. Die Europäische Union arbeitet an einem Projekt mit dem Ziel, die Erde mit einer Auflösung von einem Kilometer möglichst genau zu replizieren. Dadurch sollen Extremwetterereignisse, Überflutungen, Dürren und Waldbrände mit höherer Genauigkeit vorhergesagt werden können. Auch die Auswirkungen des Menschen und bestimmter politischer Regelungen sollen in diesem Projekt dargestellt werden. (Alvi et al. 2025) Solche großmaßstäbigen Analysen können auch für kleinmaßstäbige urbane Zwillinge hilfreich sein. Das europa- oder weltweite Modell prognostiziert, wann und in welchem Ausmaß Extremereignisse auftreten. Der UDZ kann dann für den entsprechenden Zeitpunkt die Auswirkungen des Events auf den urbanen Raum simulieren. Neben Simulationen für ganze Städte sind solche Analysen vor allem für kritische Infrastruktur von großer Bedeutung. Zhu et al. (2025) haben in ihrer Arbeit zusammengefasst, welche Teile einer Stadt als kritische Infrastruktur bezeichnet werden können und welche Studien sich bereits mit UDZ in den jeweiligen Bereichen befasst haben. (Abbildungen 27 & 28)

CI sector	CI	Risks	Applications	Study
Built environment	Buildings structure	Extreme temperature	Informed decisions on infrastructure design and maintenance.	Alibrandi (2022)
	Open space		Forecasting urban overheat exposure to reduce its impact.	Mavrokapnidis et al. (2021)
	Urban Morphology		Simulating cool air flows for UHI mitigation.	Schrotter and Hürzeler (2020)
		Extreme temperature/ Flood	Enabling collaboration among stakeholders, including citizens, for proactive planning.	Dembski et al. (2019)
	Heritage	Flood/ Sea-level rise	Protecting the natural and cultural heritage and the unique waterways and pedestrian roads.	Villani et al. (2025)

Abbildung 27: UDZ für Kritische Infrastruktur (Zhu und Jin 2025)

(Alibrandi 2022; Mavrokapnidis et al. 2021; Schrotter und Hürzeler 2020; Dembski et al. 2020; Dembski et al. 2019 - 2019; Villani et al. 2025)

CI sector	CI	Risks	Applications	Study
Transportation	Rail	Extreme temperature/ Flood	Providing stakeholders with accurate and up-to-date information for infrastructure investments and safety measures.	Kaewunruen et al. (2022)
	Road	Flood	Improving the coordination among different actors involved in disaster response.	Fan et al. (2021)
			Pinpointing vulnerable locations across the city under various flood scenarios.	Ghaith et al. (2022a)
			Adjusting traffic control measures in response to changing conditions, enhancing resilience to disruptions.	Xu et al. (2023)
	Port	Sea-level rise	Identifying vulnerabilities in port infrastructure, supporting investments and operational adjustments.	Karatvuio et al. (2022)
Energy	Airport	Extreme temperature/ Strong wind	Fundamental data hub envisioning resilience planning.	Agapaki (2022)
	Grid	Storm	Identifying system vulnerabilities	Braik and Koliou (2023)
	Utility pole	Strong wind	Encouraging participatory sensing involves the community in monitoring and reporting infrastructure issues, fostering a collaborative approach to urban resilience.	Ham and Kim (2020)
Water	Gas pipe	Extreme temperature	Knowledge graph for integrating cross-domain data, enhancing collaboration among experts.	Savage et al. (2022)
	Reservoir	Storm	Utilising user-generated data from social media enables a more comprehensive understanding of local impacts during disasters.	Fan et al. (2020)
	Wetland	Flood	Supporting decision-making in pollution control and habitat preservation. Adjusting water flow and introducing filtration mechanisms.	Ahleroff et al. (2021)
	Drainage		Early warning systems enable mitigation implementation. Supporting land-use planning.	Ghaith et al. (2022b)
			Improving the accuracy of water level predictions, assessing the impact of different preventive measures.	Roudbari et al. (2024)
	Inland waterway	Generic	Forecasting responses to both anticipated and unforeseen events.	AlexandraMicu et al. (2025)

Abbildung 28: UDZ für Kritische Infrastruktur (Zhu und Jin 2025)

(Kaewunruen et al. 2022; Fan 2022; Ghaith et al. 2022a; Xu et al. 2023; Karatvuio et al. 2022; Agapaki 2022; Braik und Koliou 2023; Ham und Kim 2020; Savage et al. 2022; Fan et al. 2020; Ahleroff et al. 2021; Ghaith et al. 2022b; Roudbari et al. 2024; AlexandraMicu et al. 2025)

3.5.3 Verschmutzung

Im Zusammenhang mit der Luftverschmutzung gibt es bereits mehrere Projekte, die einen UDZ nutzen, um den Zustand der Luft in einer Stadt zu überwachen oder zu simulieren. Für solche Modelle spielt das IoT eine bedeutende Rolle, denn ohne eine Vielzahl an Sensoren und Daten können keine sinnvollen Analysen durchgeführt werden. Zum Teil haben diese Projekte bereits künstliche Intelligenz integriert. Dadurch können Muster erkannt und Prognosen erstellt werden. Außerdem können mithilfe von UDZ mögliche Ursachen für Unterschiede in der Luftverschmutzung innerhalb einer Stadt gefunden werden. Neben der Luftverschmutzung wird auch die Lärmverschmutzung bereits in vielen UDZ analysiert (z. B. Wien). Die Ergebnisse solcher Analysen oder Simulationen können in der Stadtplanung sehr hilfreich sein. Andere Arten von Verschmutzungen (Wasserverschmutzung, Bodenverschmutzung, Lichtverschmutzung) werden in UDZ bislang noch kaum behandelt. (Alvi et al. 2025)

3.5.4 Energiemanagement

Der Energiesektor war einer der ersten Bereiche, der die Idee des Digitalen Zwillings aufgriff. Jüngste Entwicklungen machen den Einsatz von UDZ noch sinnvoller. Zu diesen neuen Entwicklungen zählen intelligente Stromzähler und sogenannte Microgrids (kleine Energiesysteme, die sich selbst erhalten). UDZ sollen bei der Planung und Optimierung solcher Systeme helfen. (Alvi et al. 2025) Der Zwilling eines Microgrids kann als eigener kleiner Zwilling betrachtet werden, der mit dem UDZ in Verbindung steht. Der Zwilling kann bei der Vorhersage, beim Management und Monitoring sowie bei der Erkennung von Fehlern helfen. Ein weiterer Vorteil ist die rasche Wahrnehmung von Cyber-Attacken. Das ist möglich, da das Microgrid ständig Daten an den DZ sendet. (Jafari et al. 2023) Ein Beispiel ist die Arbeit von Austin et al. (2020). Sie erstellten einen UDZ für die Stadt Chicago. In diesem Modell sind Gebäudeblöcke nach ihrem Energieverbrauch kategorisiert. Für die Erstellung dieses UDZ wurde KI und ML eingesetzt. Das Ergebnis kann als Entscheidungsgrundlage für städtische Infrastruktur-Projekte dienen. (Alvi et al. 2025)

Das ist nur ein kleiner Ausschnitt bereits realisierter Anwendungsgebiete von UDZ. Die Liste weiterer Anwendungen ist lang. Tabelle 3 zeigt eine Übersicht über weitere Anwendungsgebiete und wissenschaftliche Publikationen zu den jeweiligen Bereichen.

Tabelle 3: weitere Anwendungsgebiete von UDZ

Umwelt & Klima	Urban Heat Island Management	(Omrany und Al-Obaidi 2025)	Application of digital twin technology for Urban Heat Island mitigation: review and conceptual framework
Umwelt & Klima	Kritische Infrastrukturen und Klimawandel	(Zhu und Jin 2025)	Data-Driven Urban Digital Twins and Critical Infrastructure Under Climate Change: A Review of Frameworks and Applications
Umwelt & Klima	Küstenresilienz und Katastrophenvorsorge	(Chen et al. 2025)	Integrating Urban Digital Twin with Cloud-Based Geospatial Dashboard for Coastal Resilience Planning: A Case Study in Florida
Umwelt & Klima	Starkregen- und Hochwassermanagement	(Kim et al. 2025)	Stormwater digital twin with online quality control detects urban flood hazards under uncertainty
Umwelt & Klima	Frühwarnsysteme für urbane Überschwemmungen	(Hlal et al. 2025)	Digital Twin Technology for Urban Flood Risk Management: A Systematic Review of Remote Sensing Applications and Early Warning Systems
Umwelt & Klima	Überflutungsprävention	(Ge und Qin 2025)	Urban flooding digital twin system framework
Umwelt & Klima	Kritische Infrastruktur	(Ahn et al. 2025)	Critical urban informatics for urban digital twin models

Umwelt & Klima	Erdbebenresilienz und nachhaltige Stadtentwicklung	(Habib et al. 2025)	Exploring the Sustainability Benefits of Digital Twin Technology in Achieving Resilient Smart Cities During Strong Earthquake Events
Umwelt & Klima	Abfallmanagement und Ressourcenoptimierung	(Cárdenas-León et al. 2024)	Urban digital twin-based solution using geospatial information for solid waste management
Umwelt & Klima	Wettervisualisierung und -prognose	(A Novel Digital Twin Framework for Adaptive Urban Weather Visualization Using IoT and Fuzzy Logic 2024)	A novel digital twin framework for adaptive urban weather visualization using IoT and fuzzy logic
Umwelt & Klima	Luftqualitätsüberwachung	(Cowell et al. 2025)	Moving from monitoring to real-time interventions for air quality: are low-cost sensor networks ready to support urban digital twins?
Umwelt & Klima	Brand- und Rauchereignis-Management	(Lewis et al. 2024)	Fire and smoke digital twin – A computational framework for modeling fire incident outcomes
Umwelt & Klima	Stadtwachstum und Vegetationsüberwachung	(Zhao et al. 2022)	Applying Digital Twins to Research the Relationship Between Urban Expansion and Vegetation Coverage: A Case Study of Natural Preserve
Umwelt & Klima	Klimagerechte Stadtentwicklung und Nullenergiegebäude	(Omar 2025)	Integrating Zero-Energy Buildings into Smart City Strategies

Infrastruktur	Management unterirdischer Infrastrukturen	(Babanagar et al. 2025)	Digital twins for urban underground space
Infrastruktur	Energiemanagement	(Xu et al. 2024)	Novel economic models for advancing urban energy management and transition: Simulation of urban energy system in digital twin
Infrastruktur	Brückenüberwachung und Instandhaltung	(Kang et al. 2021)	Multimedia knowledge-based bridge health monitoring using digital twin
Infrastruktur	Stadtbeleuchtung und Infrastrukturplanung	(Leplat et al. 2025)	NorDark-DT: A digital twin for urban lighting infrastructure planning and analysis
Infrastruktur	Prädiktive Wartung städtischer Infrastrukturen	(Prasath und Vishnupriya 2025)	AI-Enabled Digital Twin Framework for Predictive Maintenance in Smart Urban Infrastructure
Infrastruktur	Entwässerungsmanagement und Kanalnetzüberwachung	(Bartos und Kerkez)	Pipedream: An interactive digital twin model for natural and urban drainage systems
Infrastruktur	Gebäude-Lebenszyklus	(Aditya et al. 2025)	Digital Twin Implementation in Building Lifecycle for Sustainable Urban Development
Verkehrswesen	Verkehrsmonitoring und Mobilitätsmanagement	(Ali et al. 2025)	Developing Urban Digital Twin For Mobility
Verkehrswesen	Optimierung urbaner Logistikprozesse	(Xu et al. 2025)	Towards the Autonomous Optimization of Urban Logistics: Training Generative AI with Scientific Tools via Agentic Digital Twins and Model Context Protocol
Verkehrswesen	Intelligente Verkehrssteuerung	(Dasgupta et al. 2021)	A Transportation Digital-Twin

Verkehrswesen	Autonomes Fahren	(Xu et al. 2024)	Semi-Automatic Geographic Information System Framework for Creating Photo-Realistic Digital Twin Cities to Support Autonomous Driving Research
---------------	------------------	------------------	--

4 GeoAI

Künstliche Intelligenz, vor allem Deep Learning (DL) hat viele wissenschaftliche Disziplinen transformiert. Durch KI ist es möglich, unbekannte Muster und neues Wissen in einem datenbasierten Ansatz zu entdecken und bestehende Prozesse zu beschleunigen oder verbessern. Eine dieser Disziplinen ist die Geographie, insbesondere die Geoinformation. Sie eignet sich hervorragend für den Einsatz von KI, da in diesem Gebiet eine große und vielfältige Datenmenge untersucht werden muss. (Li et al. 2024b) Bereits vor über 30 Jahren erwähnten Wissenschaftler die Vorteile eines Zusammenspiels von Künstlicher Intelligenz und GIS. Zu Beginn lag der Fokus vor allem auf Machine Learning (ML) zur geographischen Analyse und Natural Language Processing (NLP), um Text zu georeferenzieren. In der jüngsten Vergangenheit wurde auch Deep Learning eingesetzt. Dadurch können Fernerkundungsdaten besser analysiert werden (Bsp.: Landcover-Klassifizierung) und neue geographische Daten gewonnen werden (Social-Media-Daten). (Scheider und Richter 2023) Betrachtet man eine Stadt, ist diese voll von komplexen Interaktionen und Mustern. Diese bleiben mit herkömmlichen Methoden oft unentdeckt oder die Daten sind zu umfangreich, um sie händisch zu analysieren. Die rasanten Entwicklungen im Bereich der KI auf der einen Seite und die immer komplexeren Fragestellungen in der Geographie auf der anderen Seite führen zu einer großen Relevanz von GeoAI. Die zur Verfügung stehenden geographischen Daten werden immer umfangreicher und hochauflösender. Diese Daten müssen durch robuste, skalierbare und intelligente Methoden verarbeitet werden, um anschließend sinnvoll analysiert werden zu können. (Li et al. 2024b)

4.1 Definition

Li et al. (2024) definieren GeoAI als neue Technologie, die KI, große räumliche Daten und performante Berechnungen einsetzt, um räumliche Probleme zu lösen. Im weiteren Sinne umfasst GeoAI den Einsatz bereits bestehender KI-Methoden für geographische Fragestellungen. Jedoch gibt es einige Eigenschaften von GeoAI, die diesen Bereich grundlegend von der allgemeinen KI abgrenzen. Li et al. (2024) heben vier Methoden hervor:

Explizites räumliches Modellieren: Räumliche Daten und Prozesse haben zwei bedeutende Eigenschaften. Sie korrelieren räumlich und sind räumlich heterogen. Die automatische räumliche Korrelation besagt nach “Tobler’s First Law of Geography”, dass räumlich nahegelegene Objekte sich ähnlicher sind als räumlich weit entfernte Objekte. Die räumliche Heterogenität bezieht sich auf die unvorhersehbare räumliche Varianz von Prozessen an der Erdoberfläche. Diese und weitere Eigenschaften von geographischen Daten müssen von GeoAI-Algorithmen berücksichtigt werden.

Eine weitere Kerneigenschaft von räumlichen Daten ist **der geographische Rahmen**. Dadurch können Daten aus verschiedenen Quellen miteinander verschnitten werden.

Dabei müssen verschiedene Formate (Raster, Vektor, Text) möglichst genau miteinander verknüpft werden. Im Englischen spricht man von “multi-source GeoAI”.

Ein gemeinsames Merkmal vieler geophysikalischer und sozialer Phänomene, wie z. B. Wirbelstürme oder eine sich ausbreitende Pandemie, besteht darin, dass es sich um **sich entwickelnde räumliche Prozesse** handelt. Beim Modellieren müssen also räumliche und zeitliche Aspekte gleichzeitig ins Auge gefasst werden. Jüngste Forschungen versuchen, große Modelle zu erstellen, die den räumlich-zeitlichen Zusammenhang von großen Datensätzen über lange Zeiträume darstellen. Dadurch können komplexe Simulationen wie Wetter- und Klimaprognosen effizienter durchgeführt werden. Denn diese Prognosen werden traditionell mit physikalischen Simulationsmodellen erstellt, die sehr viel Rechenkapazität und Rechenzeit benötigen.

Der letzte, von Li et al. (2024) erwähnte Punkt ist **das Training und die Validierung von GeoAI-Methoden**. Neben dem Erstellen von räumlichen KI-Modellen ist die Geographie auch notwendig, um die erstellten Modelle zu trainieren und zu validieren. Dafür gibt es verschiedene Ansätze. So können beispielsweise OpenStreetMap-Daten für die Validierung der automatischen Gebäude- und Straßenerkennung dienen. Bei der Validierung wird zwischen zufälliger und räumlicher Validierung unterschieden. Bei der zufälligen Validierung wird der geographische Kontext nicht berücksichtigt. Die räumliche Validierung eignet sich für noch ungetestete geographische Gebiete, da sie den räumlichen Kontext inkludiert. (Li et al. 2024b)

Verallgemeinert kann man sagen, dass man unter GeoAI die Synthese von Künstlicher Intelligenz mit räumlichen Daten versteht. Um dieses Zusammenspiel optimal nutzen zu können, muss die Disziplin der GeoAI mit GIS verbunden werden. Abbildung 29 zeigt das Zusammenspiel von GeoAI und GIS. Nur durch die Verknüpfung mit GIS können die Daten strukturierter verarbeitet und gespeichert werden. Das führt zu einer besseren Modellierung und einer höheren Interoperabilität mit anderen Systemen. Abbildung 29 geht auch grob auf die unterschiedlichen Begriffe innerhalb der KI ein. Diese werden in den Methoden jeweils kurz beschrieben. Der Begriff KI/AI beziehungsweise GeoAI ist der Überbegriff und kann weiter unterteilt werden. (Boutayeb et al. 2024)

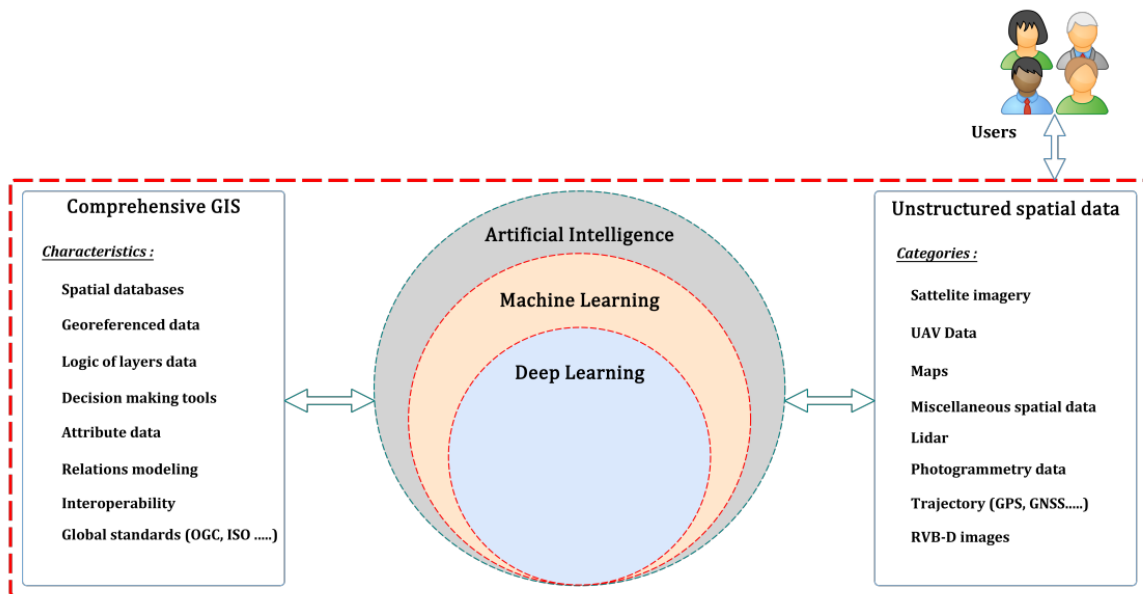


Figure 1: GeoAI and GIS.

Abbildung 29: Zusammenspiel von GeoAI und GIS (Boutayeb et al. 2024)

4.2 Historie

Vor mehr als 30 Jahren begannen die ersten Wissenschaftler, Vorschläge zu machen, wie man Künstliche Intelligenz im Zusammenhang mit räumlichen Problemstellungen einsetzen kann (Scheider und Richter 2023). Openshaw et al. (Openshaw und Openshaw 1997) sprechen in ihrer Arbeit von einem neuen Toolkit, das aus dem Zusammenspiel von KI und GIS entstehen kann. Sie erwarteten, dass manche GIS-Analysen in Zukunft auf KI angewiesen sein werden. Und damit sollten sie Recht behalten. Heutzutage sind einige GIS-Methoden eng mit KI verbunden. Vor allem der Einsatz von Machine Learning für geographische Fragestellungen ist in vielen Organisationen fest verankert. (Scheider et al. 2023) Bereits 1984 erwähnte Smith das Potenzial von KI für die Geographie. Mit einem effizienten Einsatz von KI-Methoden, müssen sich Forscher weniger Zeit mit der technischen Umsetzung auseinandersetzen und können sich auf kreative Arbeitsprozesse fokussieren, so Smith. (Smith 1984) Nystuen (Nystuen 1984) sah die Entwicklung und das Zusammenspiel von KI und Geographie etwas kritischer. Er sah die von Smith erwähnten Vorteile als fragwürdig an. In den nächsten Jahrzehnten beschäftigten sich immer wieder Wissenschaftler mit dem Einsatz von KI in der Geographie. Openshaw ((Openshaw 1992), 1998) arbeitete an KI-Methoden für räumliche Analysen und Modelle. Miller und Goodchild (Miller und Goodchild 2015) diskutieren inwiefern KI helfen kann, mit Big Data in der Geoinformation umzugehen. Schließlich kam der Begriff GeoAI auf. Dieser wird von Janowicz et al. (Janowicz et al. 2020) erwähnt und ist eine Abkürzung für "Geospatial Artificial Intelligence". Traditionelle KI ist schon lange Teil der Geoinformation. Ob als unüberwachte Lernalgorithmen zur Datenbeschaffung oder als überwachte Methoden zur räumlichen

Autokorrelation. Jedoch gibt es immer wieder Wellen, die neue Entwicklungen und Möglichkeiten mit sich bringen. (Sabbata et al. 2023)

Nach den ersten Überlegungen und Publikationen zum Thema KI im geographischen Kontext (Smith 1984) vergingen mehr als 30 Jahre bis zur ersten Erwähnung von GeoAI. Diese erfolgte im Rahmen eines Workshops der SIGSPATIAL-Konferenz 2017, die von der Association for Computing Machinery (ACM) veranstaltet wurde. Bei dieser Konferenz wurde GeoAI als Verbindung von Künstlicher Intelligenz mit geographischen und räumlichen Daten definiert. Die erste Erwähnung des Begriffs gilt als großer Meilenstein und führte zu vielfältigen und raschen Entwicklungen im Bereich der GeoAI. (Boutayeb et al. 2024)

Abbildung 30 zeigt die Anzahl an wissenschaftlichen Arbeiten zum Thema GeoAI im urbanen Kontext. Vor allem in den letzten fünf Jahren ist hier ein starker Anstieg zu erkennen. In grundlegenden, urbanen Thematiken (Verkehr, Umwelt, etc.) können mit GeoAI wertvolle, neue Informationen gewonnen werden. Dies geschieht durch neue Technologien wie etwa neuronale Netzwerke oder sogenannte Knowledge Graphs. In Abbildung 31 fassen Lu et al. (2025) die Entwicklungsphasen der GeoAI zusammen. (Lu et al. 2025)

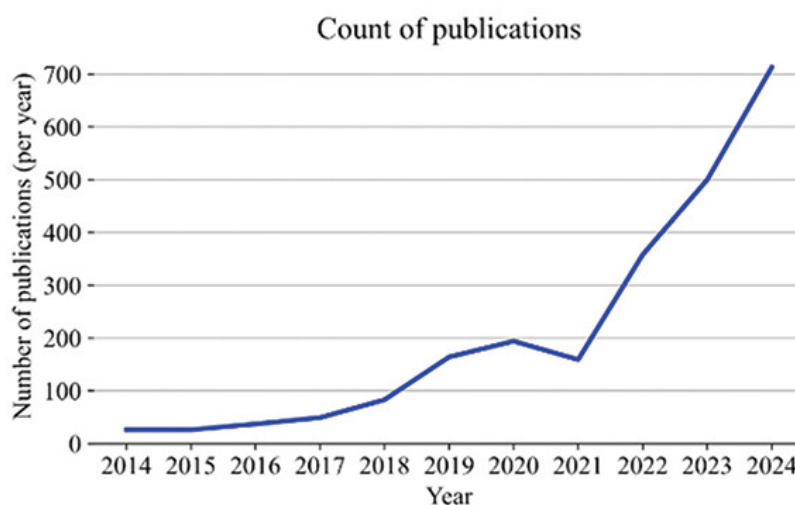


Abbildung 30: Publikationen zum Thema GeoAI (Lu et al. 2025)

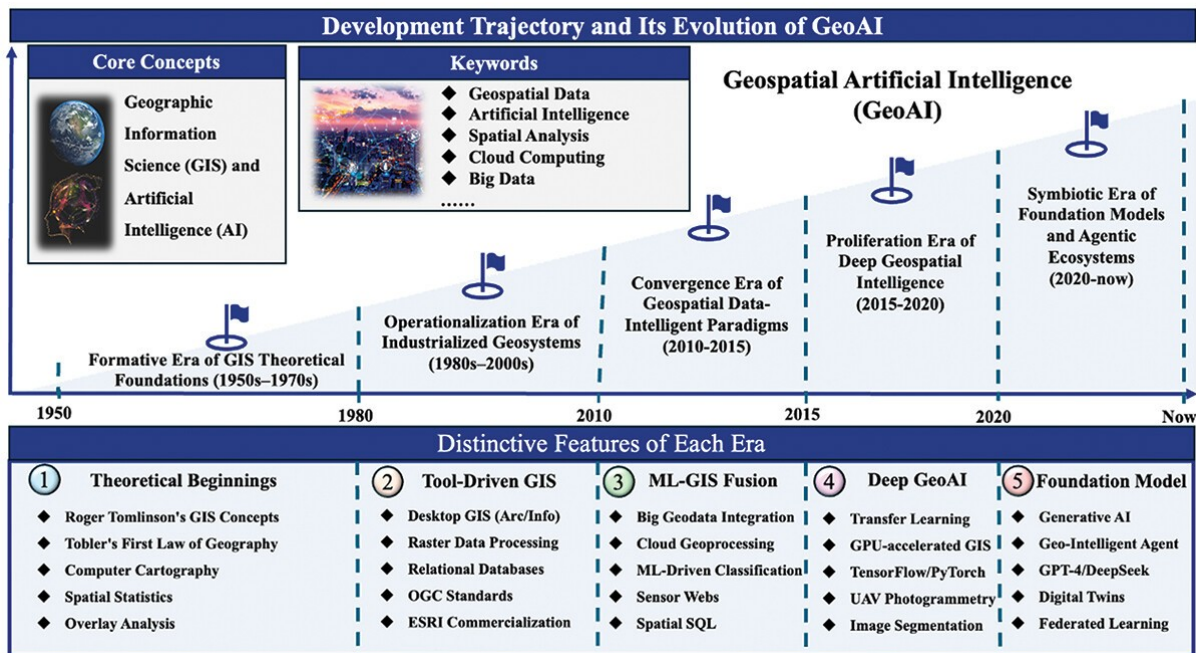


Abbildung 31: Entwicklungsphasen von GeoAI (Lu et al. 2025)

4.3 Methoden

Wie bereits in Kapitel 4.1 kurz erwähnt, besteht die GeoAI aus verschiedenen Subdisziplinen und Methoden. In diesem Kapitel werden die prominentesten Methoden der GeoAI kurz vorgestellt.

4.3.1 Machine Learning (ML)

Machine Learning (ML) im Bereich der Datenverarbeitung lernt die Struktur einer Datenquelle kennen und kann dadurch Muster erkennen und bei Vorhersagen hilfreich sein. (Austin et al. 2020) ML wird bereits seit über zehn Jahren im Bereich der Geoinformation eingesetzt. Santosh et al. (Santosh et al. 2021) definierten ML als mehrere Techniken zur Annäherung an eine Funktion, die aus räumlichen Eingabedaten räumliche Ausgabedaten generiert und dabei aussagekräftige, nicht redundante Informationen aus Beispieldaten extrahiert. ML kann überwacht oder unüberwacht trainiert werden. Das überwachte Training besteht normalerweise aus zwei Schritten. Zum einen dem Training und zum anderen der Vorhersage. So werden beispielsweise bei automatischen Kategorisierungen die Objekte im Trainingsdatensatz händisch zugewiesen. Anhand dieser Informationen kann der ML-Algorithmus andere Gebiete automatisch klassifizieren. Meist gibt es beim ML auch einen Testdatensatz, der zur Überprüfung der Genauigkeit dient. Auch für den Testdatensatz werden die Objekte händisch klassifiziert. Jedoch fließen diese Informationen nicht in den ML-Algorithmus ein, sondern werden mit den Ergebnissen der automatischen ML-Klassifizierung verglichen. (Austin et al. 2020) Zu den bekanntesten Berechnungen des überwachten ML zählen die lineare und logistische Regression, sowie Decision Trees oder Random Forest. Bei Decision Trees werden die Daten immer weiter, aufgrund von bestimmten Kriterien aufgeteilt, bis die finale Vorhersage (z.B. Kategoriezuweisung) erfolgt. Random Forest

verwendet mehrere Decision Trees und verwendet den Durchschnitt der Bäume für die Vorhersage. Bei der linearen Regression wird der Zusammenhang zwischen einer erklärenden und einer abhängigen Variable analysiert. Dadurch können Werte für bestimmte Punkte oder Objekte vorhergesagt werden. Im Gegensatz zur linearen Regression ist das Ergebnis einer logistischen Regression ein Prozentwert. Dies kann zum Beispiel die Wahrscheinlichkeit für die Zugehörigkeit zu einer Objektklasse sein. Außerdem wird überwachtes ML für die Clustererstellung mittels K-Nearest Neighbors verwendet. (Boutayeb et al. 2024)

Beim unüberwachten ML werden keine Ergebnisse händisch erfasst beziehungsweise keine Trainingsdaten verwendet (Austin et al. 2020). Diese Form scheint sehr praktisch zu sein, da es in vielen Fällen keine Trainingsdaten gibt. Beispiele für unüberwachtes ML sind die Clustering-Methoden K-means und DBSCAN. (Boutayeb et al. 2024) Allerdings hat das unüberwachte ML auch Nachteile. So kann das Modell beispielsweise kein sinnvollen Muster erkennen und dementsprechend kategorisieren. Da man keine Daten händisch klassifiziert, ist es außerdem schwieriger, das Ergebnis zu validieren. Außerdem erfordert die Erstellung eines unüberwachten ML-Algorithmus viel Know-how und Zeit. Denn die Algorithmen reagieren oft sehr sensibel auf die Berechnungsparameter. Ein Beispiel dafür ist die Anzahl an Clustern oder Klassen. (Zhou 2021)

Es gibt auch noch das semi-überwachte ML, bei dem zuerst die unüberwachte Methode und anschließend das überwachte ML angewandt wird. Dabei werden zuerst Muster mittels unüberwachtem ML erkannt. Diese Muster werden dann verwendet, um durch überwachtes ML Vorhersagen zu treffen. (Austin et al. 2020) Bei dieser Methode werden sowohl gelabelte als auch ungelabelte Daten verwendet. Dadurch entsteht eine hohe Flexibilität und eine effiziente Nutzung der Eingabedaten. Doch auch hier gibt es einige Nachteile. Dazu zählen, wie beim unüberwachtem ML, die fehlerhafte Mustererkennung und die erschwerte Evaluierung des Modells. Der Aufbau solcher kombinierter ML-Algorithmen ist meist noch komplexer als beim unüberwachten ML und dementsprechend mit hohem Zeit- und Kostenaufwand verbunden. Prinzipiell kann man sagen, dass unüberwachtes und semi-überwachtes ML vor allem bei Daten verwendet wird, bei denen ein händisches Labeln sehr schwierig und aufwendig ist. (Zhou 2021)

4.3.2 Deep Learning

Machine Learning ist der Überbegriff für Methoden, bei denen Computer aus Daten lernen und Zusammenhänge herstellen, ohne explizit programmiert zu werden. Eine bedeutende Sparte des ML ist Deep Learning (DL). Unter DL versteht man Modellarchitekturen, die versuchen, das menschliche Gehirn zu imitieren. Dazu werden Deep Neural Networks (DNN) als Grundansatz verwendet. DNN sind künstliche neuronale Netzwerke mit vielen Schichten, die in der Lage sind, komplexe Muster in

großen Datenmengen zu erkennen. Dieses Kapitel beschreibt die häufigsten DL-Modelle. (Boutayeb et al. 2024)

Die grundlegendste Form des DL ist das Multi-Layer Perceptron (MLP), das aus mehreren Schichten von künstlichen Neuronen besteht: einer Eingabeschicht, einer Ausgabeschicht und mehreren versteckten Schichten dazwischen. Die Eingabeschicht überträgt die Darstellung der Eingabedaten an die versteckten Schichten, wobei jedes Neuron mit allen Neuronen der nächsten Schicht verbunden ist. Dieser Informationstransfer basiert auf nichtlinearen Aktivierungsfunktionen, die es dem Modell erlauben, Abhängigkeiten zwischen den Daten besser zu verstehen. Die Aktivierungsfunktionen können beispielsweise dazu dienen, die Ausgabeschicht zu klassifizieren oder eine lineare Regression zu erstellen. Am Ende steht die Ausgabeschicht, die das Ergebnis des Algorithmus ausgibt. (Boutayeb et al. 2024)

Eine andere Form des DL sind Convolutional Neural Networks (CNN), die vor allem für die Bild- und Videoverarbeitung entwickelt wurden. Ein CNN besteht normalerweise aus drei wesentlichen Elementen. Dabei handelt es sich um **Convolutional Layers** für das Extrahieren von einzelnen Objekten, **Pooling Layers** für die Reduzierung der Dimensionen und **Fully Connected Layers** (FC) für die Klassifizierung der Ergebnisse. CNN werden von vielen Wissenschaftlern behandelt. Die erste Erwähnung stammt von Lang (Lang et al. 1990). Seither wurden CNN kontinuierlich weiterentwickelt. CNN haben bewiesen, dass sie einige grundlegende geo-relevante Aufgaben effizient umsetzen können. Dazu zählen die Objekterkennung, die Objektklassifizierung und die Objektsegmentierung. (Boutayeb et al. 2024)

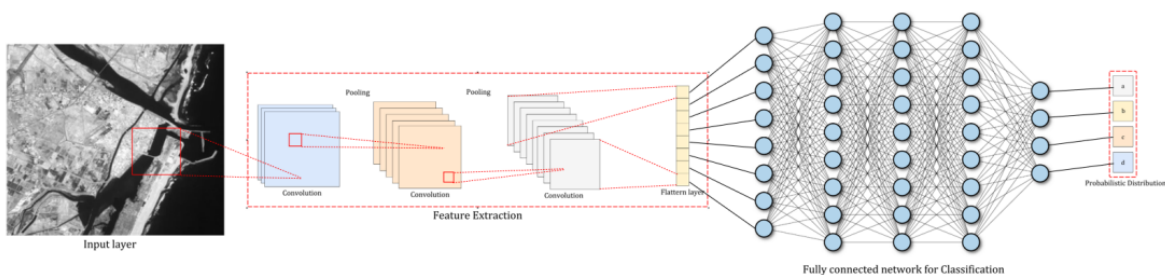


Abbildung 32: CNN-Architektur (Boutayeb et al. 2024)

Eine besondere Form des DL sind Recurrent Neural Networks (RNN). Diese Netzwerke eignen sich besonders für sequenzielle Daten. Dazu zählen Texte, Audioaufnahmen und sonstige Daten in Zeitreihen. RNN sind dazu in der Lage, sich vorherige beziehungsweise ältere Dateneinträge zu merken und für die Verarbeitung aktueller Eingaben zu verwenden. Allerdings stoßen RNN bei sehr langen Zeitreihen an ihre Grenzen, da sie beginnen, frühere Informationen zu vergessen. Man spricht vom Vanishing-Gradient-Problem. (Boutayeb et al. 2024) Weiterentwicklungen wie Long Short-Term Memory (LSTM) und Gated Recurrent Units (GRU) adressieren dieses Problem. (Salem 2022; Sherstinsky 2020)

Eine weitere Modellart sind Autoencoder (AE) (Abbildung 33). Diese Form wird vor allem für die Komprimierung und Rekonstruktion von Daten verwendet. Ein Encoder komprimiert die Daten und behält dabei die wichtigsten Informationen. Ein Decoder hingegen versucht, aus den komprimierten Daten die ursprünglichen Informationen zu rekonstruieren. Diese Methode wird zum Beispiel in der Bildkomprimierung oder Anomalie-Detektion angewandt. (Li et al. 2023; Boutayeb et al. 2024)

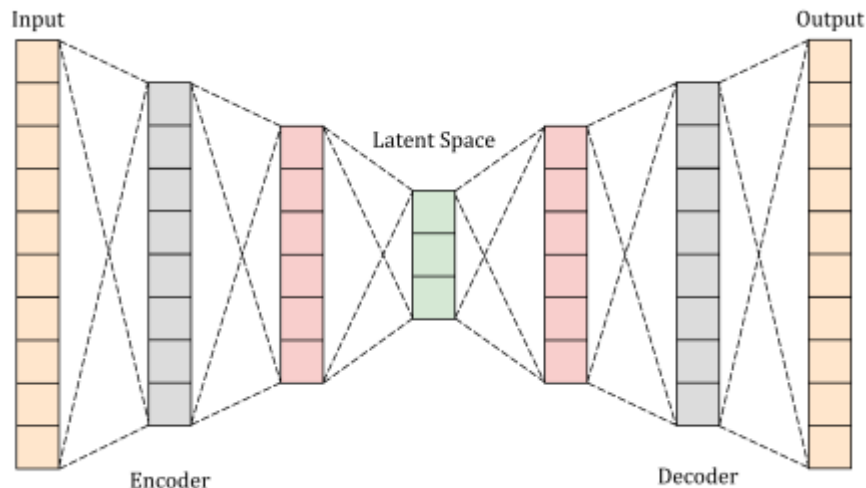


Abbildung 33: Auto-Encoder-Architektur (Boutayeb et al. 2024)

Neben den grundlegenden Architekturformen von DL gibt es noch einige fortschrittliche Konzepte im Zusammenhang mit diesem Thema. Diese Konzepte können bestehende Formen des DL entscheidend verbessern. Eines dieser Konzepte ist das Reinforcement Learning. Hierbei verbessert sich die KI aufgrund von “Belohnungen” und “Bestrafungen”, die sich aus den Ergebnissen ableiten. Die KI lernt also aus ihren Fehlern. Diese Form findet auch in der Robotik und beim autonomen Fahren Anwendung. (Boutayeb et al. 2024)

Eine andere Form sind Attention Mechanism. Diese Technologie ermöglicht es neuronalen Netzen, sich gezielt auf die wichtigsten Teile der Eingabedaten zu konzentrieren (Abbildung 34). Besonders erfolgreich ist der Einsatz dieser Methode in sogenannten Transformer-Modellen, die beispielsweise die Grundlage für Large Language Models (LLMs) bilden. Ein Transformer besteht aus einem Encoder und einem Decoder und ist in der Lage, komplexe Zusammenhänge in Texten zu erfassen und zu erstellen. Der Attention-Mechanism hilft, relevante Informationen aus langen Eingabesequenzen zu erkennen. (Boutayeb et al. 2024)

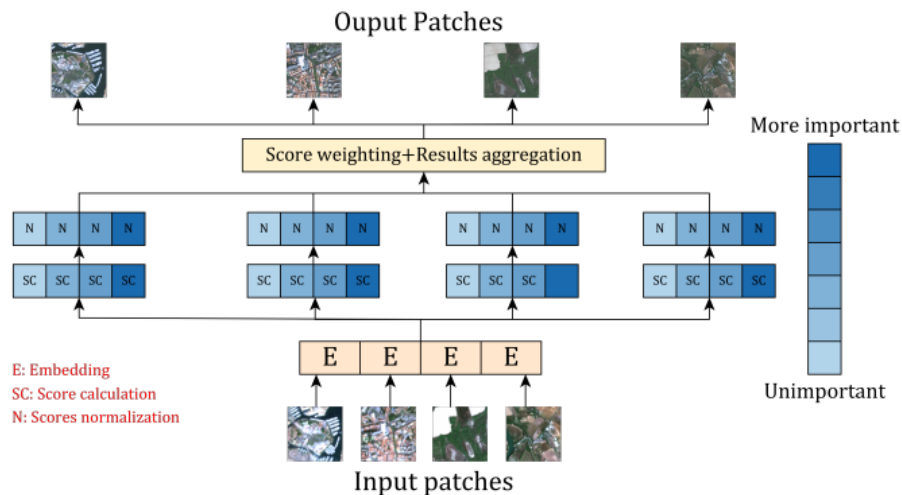


Abbildung 34: Attention-Mechanism

4.3.3 Räumliche Grundmodelle (Prithvi)

Jüngste Entwicklungen im Bereich der GeoAI gehen in Richtung der Erstellung von Grundmodellen (Foundation Models). Diese sind das Gegenteil von problem- oder aufgabenspezifischen Modellen. Ein Grundmodell wird mit sehr großen Datenmengen trainiert und ist in der Lage, verschiedenste räumliche Aufgaben zu erfüllen. Im Vergleich zu den rasanten Entwicklungen von Large Language Models (LLMs) hinkt die Entwicklung von visuellen Grundmodellen jedoch stark hinterher. Zu den Herausforderungen bei der Erstellung von visuellen Grundmodellen zählen die unterschiedlichen Ansprüche an das Ergebnis und die Komplexität von Bild-zu-Bild-Algorithmen. Außerdem sind für das Training solcher Modelle eine große Datenmenge und erhebliche Rechenkapazität notwendig. Deshalb ist es akademischen Forschungseinrichtungen meist nicht möglich, in diesem Gebiet zu forschen und zu entwickeln. Die nötige Datenmenge und Rechenkapazität haben hauptsächlich technologische Großkonzerne. Ein Beispiel für ein solches visuelles Grundmodell ist „Prithvi“. Prithvi wird von der US-amerikanischen Aeronautik- und Raumfahrtbehörde NASA (National Aeronautics and Space Administration) und dem internationalen Konzern IBM (International Business Machines) entwickelt. Prithvi basiert auf einer fortgeschrittenen KI-Architektur namens „Vision Transformer“ und verwendet einen selbstüberwachten Lernalgorithmus. Das Besondere an Prithvi ist der Fokus auf räumlichen Daten. Es wird vor allem mit Landsat- und Sentinel-2-Daten trainiert. Dadurch steigt die Qualität von Analysen wie etwa der Kartierung und Vorhersage von Naturgefahren. Ein großer Vorteil von Prithvi ist die kostenlose Verfügbarkeit der Daten und des Modells. Dieser Aspekt wirkt sich positiv auf die weitere Verbesserung von GeoAI-Modellen aus. (Li et al. 2024b)

4.3.4 XAI-Modelle

Da unsere Abhängigkeit von intelligenten Maschinen immer weiterwächst, ist es besonders wichtig, transparente Modelle zu verwenden, um die Verarbeitungs- und Interpretationsschritte der KI nachvollziehen zu können. Ein erklärbares, transparentes Modell wird als Goldstandard angesehen. In der Fachsprache spricht man von Explainable Artificial Intelligence (XAI). Dwivedi et al (2023) fassen in ihrer Arbeit die wichtigsten Schritte von XAI zusammen. Der erste Schritt unterscheidet sich noch nicht von herkömmlichen KI-Modellen. Hier wird das KI-Modell gebaut und trainiert. Anschließend wird eine Erklärungsmethode ausgewählt. Dazu zählen beispielsweise SHAP oder LIME. Mit diesen Erklärungsmethoden werden menschenlesbare Texte generiert, die die Vorgehensweise der KI beschreiben. Diese Beschreibungen müssen evaluiert werden, um ihre Sinnhaftigkeit zu überprüfen. Abschließend wird das Modell je nach Ergebnis der Evaluierung angepasst. Dieser Ablauf muss mehrmals wiederholt werden, um ein qualitativ hochwertiges XAI-Modell zu erhalten. (Dwivedi et al. 2023)

4.4 Einsatzgebiete

GeoAI wird in einer Vielzahl von wissenschaftlichen Disziplinen eingesetzt. Dazu zählen die Klimamodellierung, das Gesundheitswesen, die Analyse menschlicher Bewegungsmuster, die Fernerkundung und Bildanalyse, die Stadtplanung, die Katastrophenhilfe und viele mehr. (Li et al. 2024b) Boutayeb et al. (2024) unterteilen die Einsatzgebiete von GeoAI in sechs unterschiedliche Kategorien (Abbildung 35). Alle diese Themen können Bestandteil eines UDZ sein, wobei vor allem die urbane Planung (Urban Planning) am bedeutendsten für UDZ ist. Auch das Disaster Management ist bereits Bestandteil vieler UDZ. Hier ist die Hochwassersimulation ein häufiges Anwendungsfeld. Auch die Überwachung von umweltrelevanten Kennzahlen und die Durchführung von Simulationen in diesem Bereich werden in vielen UDZ umgesetzt oder entwickelt. (Boutayeb et al. 2024) Je nach Einsatzgebiet und Anwendungsfall müssen die GeoAI-Methoden mit unterschiedlichen Datensätzen zurechtkommen. Dazu zählen Satelliten-, Luft- und Drohnenbilder, LIDAR-Punktwolken und verschiedene Vektordaten, wie beispielsweise OpenStreetMap-Daten. Weiters werden Trajektorien, Adressdaten und Namensgütern von GeoAI-Technologien verarbeitet. (Boutayeb et al. 2024)

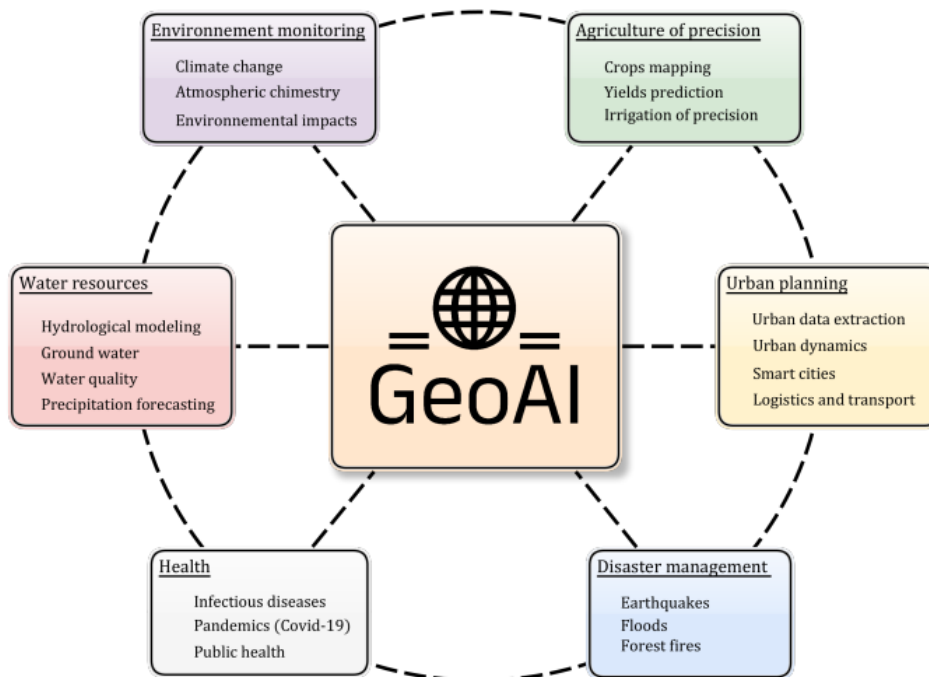


Abbildung 35: Einsatzgebiete von GeoAI (Boutayeb et al. 2024)

4.4.1 Nationale Kartenwerke/Datahubs

Viele Länder arbeiten an einer Art nationaler Karte. Dabei handelt es sich um eine Sammlung räumlicher Daten. Dazu zählen unter anderem Höhenmodelle, hydrographische Daten, Orthofotos und Textbeschriftungen. Die Erstellung solcher Sammlungen ist sehr aufwendig und zeitintensiv. Das händische Einarbeiten der Daten dauert mehrere Jahre und lohnt sich deshalb oft nicht. Auch herkömmliche KI-Algorithmen stoßen hier an ihre Grenzen, denn die genaue Lokalisierung von Objekten an der Erdoberfläche gelingt oft nicht. GeoAI hat das Potenzial, die Erstellung Nationaler Karten, um ein Vielfaches zu erleichtern und zu beschleunigen. Die Kombination von räumlichen Daten aus verschiedenen Quellen und das Anwenden von geographischem Fachwissen zeichnet GeoAI aus. GeoAI kann die Daten georeferenzieren und skalieren und dadurch mit verschiedenen Auflösungen und Projektionen kombinieren. (Li et al. 2024b)

4.4.2 urbane Planung

GeoAI beginnt bei der Extraktion relevanter Informationen aus bestehenden Daten. Es können bestehende Fernerkundungsdaten oder Kartenwerke automatisch kategorisiert werden. Diese automatische Klassifizierung ermöglicht eine hohe Präzision und eine erheblich schnellere Verarbeitung. Für die Klassifizierung gibt es verschiedene Modelle. Alem und Kumar (Alem und Kumar 2022) haben drei Klassifizierungsmodelle miteinander verglichen. Außerdem erscheinen laufend Studien, die neue Methoden für die Klassifikation entwickeln (z. B. Zhang et al 2021). Neben der Kategorisierung von urbanen Daten, ist GeoAI in der Lage komplexe urbane Dynamiken zu verstehen. Hat die

GeoAI vergangene und aktuelle Dynamiken verstanden, kann sie Prognosen für zukünftige Entwicklungen tätigen. Der große Vorteil von GeoAI ist die Möglichkeit der Automatisierung. Immer mehr Städte verwenden GeoAI, um die urbane Infrastruktur noch „smarter“ zu machen. (Boutayeb et al. 2024) So ist es beispielsweise möglich, Werbeflächen anhand von Google-Street-View-Bildern automatisch zu erkennen. Dabei gelingt es mehr als 85 % der Flächen zu identifizieren. (Ye et al. 2019) Natürlich können auch andere Flächen und Objekte automatisiert erkannt und klassifiziert werden. Diese Methoden ermöglichen es, aus bestehenden Daten, neue Informationen zu gewinnen. Ein weiteres bedeutendes Anwendungsgebiet im urbanen Raum ist die Optimierung des Verkehrs- und Transportsystems. Dazu zählt die Vorhersage von Fahrt- oder Ankunftszeiten, die Verbesserung der Sicherheit und die nachhaltigere Nutzung von Verkehrswegen und -mitteln. (Boutayeb et al. 2024)

4.4.3 Semantische Anreicherung

Unter Semantik versteht man die Bedeutung von Daten, die es Informationssystemen erlaubt, nicht nur die Attribute von Objekten, sondern auch deren Verhalten und Beziehungen zu anderen Objekten zu beschreiben. Im geographischen Kontext können Objekte im urbanen Raum klassifiziert und ihre Interaktionen dargestellt werden. Eine semantische Struktur ist von großer Bedeutung, um automatische Schlussfolgerungen zu generieren. Denn selbst wenn geographische Daten oft hochauflösend und umfassend sind, fehlt häufig ein konzeptioneller Rahmen, der den Daten inhärente Bedeutung verleiht. Unter semantischer Anreicherung versteht man den Prozess, der den geographischen Rohdaten Bedeutung und Kontext verleiht. Dafür werden Metadaten, Ontologien und Beziehungen hinzugefügt, die gemeinsam die Interpretierbarkeit und den Informationsgehalt erhöhen. Jüngste Forschungen beschäftigen sich auch mit der Erstellung von zeitlichen Semantiken zusätzlich zur räumlichen Semantik. Dadurch können zeitliche Entwicklungen und dynamische Prozesse im urbanen Raum besser erfasst werden. GeoAI wird in vielen Anwendungsfällen und Modellen eingesetzt, um die geographischen Daten semantisch anzureichern. Dies kann entweder explizit über Ontologien oder Knowledge Graphs oder implizit durch Embeddings und Deep Learning geschehen. Explizite Semantik ist vom Menschen interpretierbar, logisch strukturiert und formal definiert. Diese Semantik wird bewusst modelliert und nicht nur gelernt. Der Mensch kann sie überprüfen, und sie basiert auf Regeln, Konzepten und Definitionen. Implizite Semantik wird in Zahlenvektoren angegeben und entsteht automatisch durch Lernprozesse wie etwa DL. Sie ist für den Menschen nicht direkt sichtbar oder erklärbar. Die implizite Semantik entsteht durch die Mustererkennung in den Trainingsdaten. Sie ist im Vergleich zur expliziten Semantik leistungstärker, aber intransparenter. (Wang et al. 2025) Im praktischen Teil dieser Arbeit wird GeoAI eingesetzt, um ein Gebäudemodell semantisch anzureichern.

4.5 Zukunft von GeoAI

Sabbata et al. (2023) streichen drei Themen hervor, die in Zukunft von großer Bedeutung für die Entwicklung von GeoAI im urbanen Raum sein werden. Erstens sollten ethische Bedenken im Zentrum von GeoAI stehen. Bereits Openshaw (1992) behandelt das Problem der Erklärbarkeit. Die heutigen Modelle werden immer komplexer. Deshalb ist es umso wichtiger, dass die Funktionsweise der Modelle verständlich und transparent ist. Die vorhin beschriebenen XAI-Modelle versuchen, dieses Problem zu lösen. Neben der technischen Umsetzung solcher XAI-Algorithmen sollte man jedoch auch nicht auf semantische und soziale Aspekte vergessen. GeoAI-Modelle sind nicht frei von Vorurteilen. Diese entstehen durch ungleiche Aufteilung der Trainingsdaten. Diese Vorurteile wirken sich auf das Ergebnis der Algorithmen aus. Da KI-Modelle immer öfter eingesetzt werden, erhöht sich ihr Einfluss. Deshalb müssen ethische Bedenken im Zentrum stehen. (Sabbata et al. 2023)

Weiters ermöglicht die GeoAI gänzlich neue Wege, um mit prominenten Datengrundlagen umzugehen. Dazu zählen beispielsweise Texte oder Street-View-Bilder. (Sabbata et al. 2023) Zu Beginn der Entwicklung von Künstlicher Intelligenz wurde vor allem symbolisch analysiert. Symbolische KI fokussiert sich auf klassische, logische, von Menschen verständliche Prinzipien und Verbindungen. Ein Beispiel für diesen Ansatz ist das Bilden von Ontologien. Der momentan dominierende Ansatz ist, die KI nicht mit Grundwissen zu füttern. Hier ist das sogenannte Deep Learning ein Beispiel. In jüngster Vergangenheit versuchen immer mehr Forscher die beiden Ansätze zu kombinieren. Diese Modelle verwenden symbolisches Wissen und neuronale Netze. (Mai et al. 2022)

Zuletzt sollten geographische Theorien und Konzepte als Eckpfeiler für neue Entwicklungen in der GeoAI dienen. Fortschritte in der GeoAI können in der Lage sein, bestehende zentrale Probleme in der Geographie neu zu denken oder zu lösen. Dazu zählt beispielsweise eine effektive, gemeinsame Kodierung von Raum und Zeit oder die Forschung an Nachbarschaftseffekten durch neuronale Netze. Mit GeoAI können die Nachbarschaftseffekte um viele weitere Aspekte, wie etwa soziale, ökonomische oder geopolitische Umstände, erweitert werden. (Sabbata et al. 2023)

Zusammenfassend lässt sich sagen, dass bei der Erstellung und Weiterentwicklung räumlicher Grundmodelle nicht nur technologische Aspekte, sondern auch geographische Theorien und ethische Grundlagen eine zentrale Rolle spielen werden. (Sabbata et al. 2023)

5 Synthese

In Kapitel 3 werden einige Probleme und Herausforderungen von UDZ angeführt und näher beschrieben. Kapitel 4 gibt einen Überblick über die verschiedenen Disziplinen der GeoAI und zeigt einige Anwendungsbereiche. Dieses Kapitel versucht, die Brücke zwischen Kapitel 3 und 4 zu schlagen. GeoAI ist bereits fester Bestandteil vieler UDZ. Doch der Einsatz von GeoAI in diesem Bereich kann noch optimiert und erweitert werden. Dadurch kann es gelingen, bestehende Probleme und Herausforderungen zu meistern. Wie bereits Kapitel 3 ist auch dieses Kapitel in vier verschiedene Bereiche von UDZ aufgeteilt. In jedem Bereich werden zunächst die Probleme wiederholt und anschließend diskutiert, inwieweit GeoAI bei diesen Problemen helfen kann. Wichtig anzumerken ist, dass es sich nur um Vorschläge und Ideen handelt, wie GeoAI helfen könnte. Ein praktisches Austesten ist im Rahmen dieser Arbeit nicht möglich. Teilweise gibt es jedoch schon erste Projekte oder Versuche. Man kann die folgenden ausgearbeiteten Vorschläge und Gedanken also auch als Ideen für künftige Forschungen und Projekte betrachten.

5.1 Datenquellen

Im Bereich der Datenquellen, liegt der Fokus auf den folgenden drei Herausforderungen:

1. Verfügbarkeit hochwertiger Daten
2. Sicherheit und Privatsphäre bei der Datenerhebung
3. Datentransfer

GeoAI eignet sich hervorragend für die Gewinnung von hochwertigen Daten. Denn dank künstlicher Intelligenz können aus bestehenden Grunddaten, neue Informationen gewonnen werden. Vor allem die Ergänzung von semantischen Informationen ist für die Weiterverarbeitung in einem UDZ besonders wertvoll. Mittels GeoAI werden also keine Daten aufgenommen, sondern aus bestehenden Daten, neue Informationen abgeleitet. Bei diesen bestehenden Daten kann es sich beispielsweise um Satellitenaufnahmen, Luftbilder, Lidar-Daten oder Social-Media-Daten handeln. Ein sehr bekanntes Beispiel für die Datengewinnung mittels GeoAI ist die Landcover-Klassifizierung. Diese kann auf Basis von Satelliten- oder Luftbildern mittels ML erstellt werden. Neben der Gewinnung von hochwertigen Daten eignet sich die GeoAI auch dafür, bestehende Daten qualitativ aufzuwerten. Das geschieht, indem Qualitätsprobleme vom Algorithmus erkannt und korrigiert werden. (Rozony et al. 2024)

Da die Daten eines UDZ oft sehr heikel sind, muss die Sicherheit und Privatsphäre gewährleistet sein. GeoAI beziehungsweise KI an sich kann teilweise helfen, die Daten ausreichend zu anonymisieren. Le Yang et al. (Le Yang et al. 2024) haben mittels ML einen neuen Algorithmus zum Schutz der Privatsphäre entwickelt. Dabei kombinieren sie die Methode der Differential Privacy (DP) mit Machine Learning (ML). DP anonymisiert Daten, indem Rauschen hinzugefügt wird. ML dient dazu, die statistischen Eigenschaften

der Daten zu erhalten. In Kombination werden die Daten so anonymisiert, dass sie für Analysen und weitere Verarbeitungen brauchbar sind. Chen et al. (2021) haben einen KI-Algorithmus verwendet, um die Sicherheit und Privatsphäre von UDZ-Daten zu erhöhen. Tama et al. (Tama und Rhee 2017) verwenden ein CNN, um Angriffe auf das System zu erkennen. Eine Alternative (Aloqaily et al. 2019) ist der Einsatz von DL, um feindliche Elektronik im UDZ-System zu entdecken. Es gibt auch Methoden, um mögliche Schwachstellen des Systems zu finden. Ein Beispiel sind sogenannte “Stacked Autoencoder”(SAE). (Jafari et al. 2023)

Eine sehr große Herausforderung ist der Datentransfer. Also der Weg vom Entstehungsort der Daten zum Ort der Weiterverarbeitung. Gerade bei Sensoren blickt man hier oft Problemen entgegen. Im Grunde kann man hier zwischen zwei Ansätzen unterscheiden. Zum einen kann die Technologie für die Übertragung verbessert werden. Zu dieser Kategorie zählen Technologien wie 5G oder LoRaWAN. Dies sind zwei komplett unterschiedliche Systeme, die beide ihre Stärken und Schwächen haben. Sie können entweder geringe Datenmengen mit geringem Aufwand oder große Datenmengen mit hohem Aufwand transportieren. Der zweite Ansatz ist die Daten bereits am Ort der Aufnahme zu bearbeiten. Durch diese Datenmanipulation nahe an der Datenquelle, verringert sich die Datenmenge, die später transferiert werden muss. Wie bereits in Kapitel 3.2.4 erwähnt, spricht man in diesem Fall von Edge Computing. GeoAI vereinfacht die Umsetzung von Edge Computing. ML-Algorithmen können bereits direkt bei der Datenquelle angewandt werden (eML = edge Machine Learning). In Zukunft könnte die KI, je nach Datenmenge und äußeren, aktuellen, technologischen Umständen, selbst entscheiden, wie umfangreich die Daten bereits am Ursprung manipuliert werden. Neben ML können auch Neuronale Netze oder DL in Kombination mit Edge Computing eingesetzt werden. Der Vorteil von ML ist, dass es wenig Rechenleistung braucht und deshalb gut auf kleinen Geräten (Sensoren) funktioniert. Neuronale Netze benötigen zwar viel Rechenleistung, um trainiert zu werden, sind aber dann ebenfalls ressourcenschonend. Diese beiden Formen der GeoAI eignen sich gut für Edge Computing. DL ist eine Sonderform von Neuronalen Netzen. Aus diesem Grund ist auch bei DL eine hohe Rechenleistung notwendig. DL-Algorithmen sind jedoch noch komplexer und vielschichtiger aufgebaut. Deshalb ist der Einsatz im Edge-Computing schwieriger umzusetzen, da die DL-Modelle stark optimiert werden, müssen, um auf rechenschwachen Geräten zu funktionieren. Dafür gibt es verschiedene Tools (TensorFlow Lite, TinyML). Verbesserungen in der Komprimierung von DL-Modellen können den großflächigen Einsatz dieser Technologie im Edge Computing ermöglichen und den späteren Datentransfer erleichtern. (Merenda et al. 2020; Simic et al. 2020)

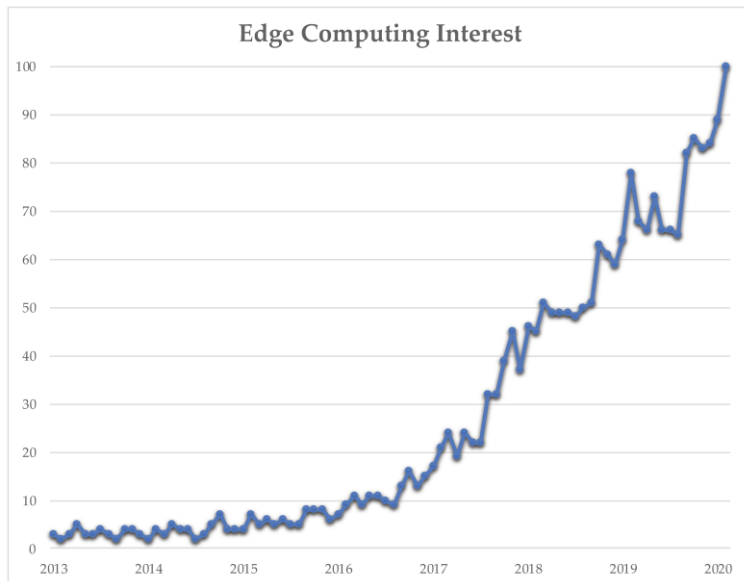


Abbildung 36: Publikationen zum Thema "Edge Computing" (Merenda et al. 2020)

5.2 Datenverarbeitung und –management

Im Bereich der Datenverarbeitung werden drei Probleme sehr häufig genannt:

1. Heterogenität der Daten
2. Big Data
3. fehlende semantische Information

Es gibt bereits vielversprechende Entwicklungen im Bereich von ML, DL und AI, um große Datenmengen zu vereinheitlichen. Doch trotz der sehr großen Datenmenge sind viele Forscher der Meinung, dass es für viele AI-Methoden noch nicht genug Daten gibt. (Ballouch et al. 2022; Brasil et al. 2022; Wang et al. 2022b; Weil et al. 2023)

Viele KI-Tools wurden bereits eingesetzt, um die Verarbeitung von Big-Data zu erleichtern. Auch im Bereich der UDZ wurde dies bereits angewandt. (Wang et al. 2023b; Braga und Rawal 2024) Vor allem ML hat sich im Umgang mit Big Data etabliert. ML wird eingesetzt, um verschiedene Datenschemata miteinander zu matchen, Fehler und Datenlücken zu erkennen und die Daten zu bereinigen. Viele Studien erwähnen die Vorteile von ML im Umgang mit großen Datenmengen. Allerdings gibt es auch Herausforderungen. Um einen ML-Algorithmus effektiv trainieren zu können, benötigt man Trainingsdaten mit einer sehr hohen Qualität. Diese benötigte Qualität kann zu lückenhaften oder auf Vorurteilen aufbauenden Ergebnissen führen. (Rozony et al. 2024)

Rozony et al. (2024) haben in einer Mindmap (Abbildung 37) die verschiedenen Herausforderungen von Big Data gesammelt. In dieser Mindmap kommt künstliche Intelligenz mehrmals vor. Man kann anhand dieser Grafik sehr gut den Zusammenhang von GeoAI und Big Data erkennen. Die Heterogenität der Daten wird in zwei Teilen der Mindmap erwähnt. Zum einen im Bereich "Standardization" und zum anderen unter "Semantic Heterogeneity". Die fehlende semantische Information wird nur teilweise im

Bereich "Semantic Heterogeneity" dargestellt. Wobei Rozony et al. (2024) weniger auf fehlende semantische Informationen, als auf unterschiedliche semantische Informationen eingehen. Dazu zählen zum Beispiel verschiedene Datumsformate oder Wege, die ID von Objekten anzugeben und zu speichern. In der Mindmap wird AI als Tool zur Transformation von Daten, Automatisierung von Prozessen, Verbesserung der Skalierbarkeit und Erhöhung der Datenqualität dargestellt. (Rozony et al. 2024) Um UDZ mit semantischen Informationen anzureichern, gibt es verschiedene Methoden. In Kapitel 8 wird genauer auf die semantische Anreicherung von UDZ eingegangen.

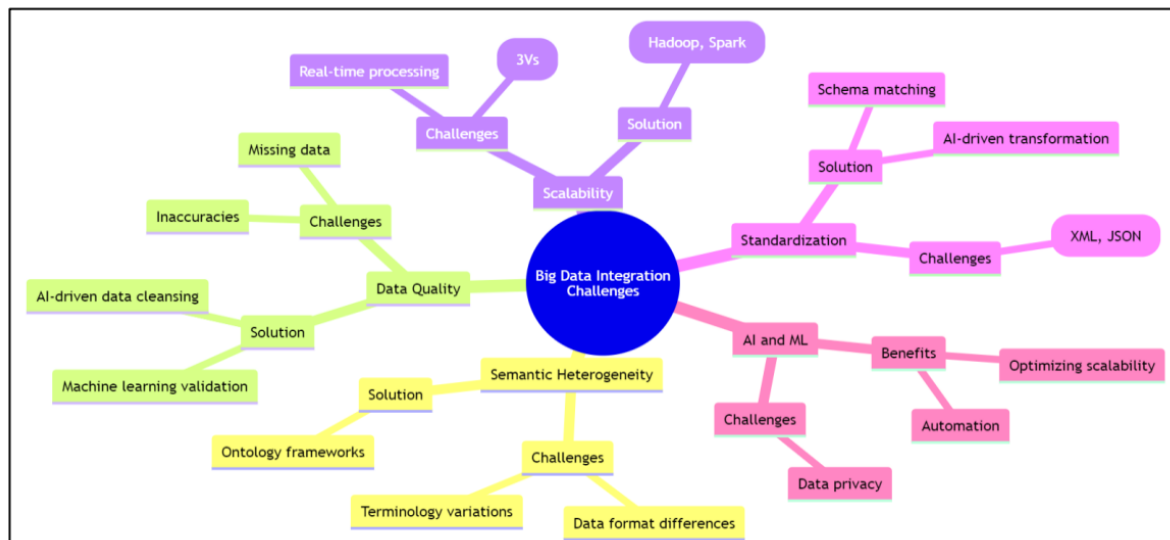


Abbildung 37: Herausforderungen bei der Integration von Big Data (Rozony et al. 2024)

Um verschiedenen Dateiformaten eine einheitliche Struktur zu verleihen, wird oft Schema Matching eingesetzt. Dabei werden Attribute, Datentypen, Datenstrukturen und semantische Informationen von verschiedenen Datenquellen miteinander verglichen und Verknüpfungspunkte gefunden. Jüngste Entwicklungen im Bereich der Künstlichen Intelligenz können einen Großteil dieser Arbeit automatisieren. Dadurch steigt die Effizienz der Datentransformation und weniger manuelle Eingriffe sind notwendig. Jedoch verlangen die wachsende Menge und steigende Komplexität der Daten nach weiteren Entwicklungen in diesem Bereich. (Rozony et al. 2024)

Neben der automatisierten Verarbeitung von Daten eignet sich GeoAI auch für das Management von Ressourcen. Das geschieht, indem die KI die voraussichtlich notwendige Rechenkapazität kalkuliert und die Ressourcen dementsprechend verteilt. (Rozony et al. 2024)

5.3 Simulation & Analyse

Im Zusammenhang mit Simulationen und Analysen innerhalb von UDZ zählen folgende Aspekte zu den größten Herausforderungen:

1. soziale Analysen
2. Analyse von Interaktionen dritter Ordnung
3. Skalierbarkeit

Um komplexe soziale und ökonomische Prozesse zu analysieren und prognostizieren, führt fast kein Weg an GeoAI vorbei. Für solche Analysen ist eine große Datenmenge notwendig, damit Muster erkannt und Prognosen abgeleitet werden können. Vor allem während der Corona-Pandemie gab es in diesem Bereich große Fortschritte. Es wurden Bewegungsmuster von Bürgern analysiert und in weiterer Folge Vorhersagen über die Ausbreitung des Virus abgeleitet. Dafür wurden Social-Media-Daten, Sensordaten und Satellitenbilder verwendet. (Roy et al. 2021) Auch bei Katastrophen und Unruhen kann GeoAI hilfreich sein. Mittels Social-Media-Daten können Bereiche in der Stadt erkannt werden, in denen große Unruhe herrscht. Weiters kann man aus den Daten ableiten, was der Grund für die Unzufriedenheit ist, und Gegenmaßnahmen setzen. (Fauzi 2023; Li et al. 2024a) In diesen beiden Fällen, aber auch in den meisten anderen sozialen Fragestellungen, benötigt man sensible Daten für die Vorhersage. Deshalb spielen hier der Datenschutz und die Sicherung der Privatsphäre eine bedeutende Rolle. Die große Herausforderung ist, diese Aspekte zu gewährleisten und gleichzeitig die Qualität und Auflösung der Daten weiter zu erhöhen, um aussagekräftige Prognosen zu erhalten. (Roy et al. 2021)

In den meisten UDZ werden Interaktionen dritter Ordnung noch nicht dargestellt oder analysiert. Doch es gibt bereits einige Projekte, die sich in diese Richtung bewegen. Die notwendige, hohe Rechenleistung bleibt weiterhin das größte Problem. (Batty 2018) Gelingt es, Interaktionen dritter Ordnung in die Analyse zu integrieren, kann die Vorhersagegenauigkeit deutlich erhöht werden. (Yi et al. 2024) Wang et al. (2023) entwickelten ein Multi-Order Spatial Interaction Network (MSIN), das die Analyse von Interaktionen höherer Ordnung ermöglicht. Mit dem Modell von Wang et al. Können menschliche Körperteile in Videos und Bildern besser erkannt werden. Das Ganze basiert auf einer Kombination eines neuronalen Netzwerkes mit mehreren Transformern. Die Ergebnisse des MSIN sind deutlich genauer als herkömmliche Methoden, die nur Interaktionen erster und zweiter Ordnung berücksichtigen. (Wang et al. 2023a) Solche Methoden werden jedoch in wesentlich kleinerem Maßstab eingesetzt. Die Analyse von Interaktionen höherer Ordnung für eine ganze Stadt und mehrere Anwendungsfelder (Verkehr, Bewegungsmuster etc.) erfordert eine sehr hohe Rechenkapazität. Die damit verbundenen Kosten sind weiterhin der Hauptgrund, warum diese Interaktionen noch nicht in UDZ analysiert werden. Doch dank GeoAI können Methoden zur Analyse von Interaktionen dritter oder vierter Ordnung weiter verfeinert und optimiert werden.

Dadurch könnte sich ein Einsatz in UDZ bald lohnen. (Batty 2024) Lucas et al. (Lucas et al. 2026) haben eine Methode entwickelt, die bei der Entscheidung helfen soll, ob sich die Analyse von Interaktionen dritter Ordnung lohnt oder ob das untersuchte System auch durch Interaktionen zweiter Ordnung repräsentiert werden kann.

Durch Konzepte wie Hadoop und Spark wird die Datenverarbeitung aufgeteilt und somit dem Problem der 3Vs entgegengewirkt. Hadoop und Spark beinhalten selbst keine GeoAI. Allerdings bauen viele GeoAI-Methoden auf diesen oder ähnlichen Konzepten auf, um große Datenmengen zu verarbeiten. Da die Datenmenge, die Komplexität der Daten und die Geschwindigkeit immer weiter steigen, benötigen Abfragen und Analysen zunehmend mehr Zeit. Hier kann GeoAI helfen, die Abfragezeit zu reduzieren. Im Vergleich zu traditionellen Methoden können Abfragen mit KI durchschnittlich um bis zu 40 Prozent schneller erfolgen. Im Zusammenhang mit geographischen Daten gibt es noch wenige Publikationen, doch die Untersuchungen in anderen Bereichen zeigen, dass GeoAI das Potenzial hat, die Herausforderungen der “3Vs” zu meistern. (Gadde 2023)

KI-gestützte Analysen können jedoch auch negative Auswirkungen haben. Ein Problem ist, dass KI-Methoden oft Vorurteile der Eingangsdaten oder sozialen Strukturen kopieren. So kann zum Beispiel eine ungleiche Verteilung der Eingangsdaten zu solchen Vorurteilen beim KI-Algorithmus führen. Ist eine bestimmte Region einer Stadt in den Trainingsdaten unterrepräsentiert, kann das dazu führen, dass die KI Ressourcen unfair verteilt. Dasselbe Problem kann auch für bestimmte Altersschichten oder Berufsgruppen auftreten. Deshalb sind Studien in diesem Bereich notwendig, um die ethischen Bedenken und Vorurteile von KI-Methoden genauer zu untersuchen. (Alvi et al. 2025)

5.4 Visualisierung

Bei der Visualisierung von UDZ gibt es einige Herausforderungen. Zu den wichtigsten zählen:

1. optimaler Detaillierungsgrad
2. geometrisch sauberes Modell
3. leichter Zugang

Die Wahl des optimalen Detaillierungsgrades (LoD) für einen UDZ ist von Anwendung zu Anwendung unterschiedlich und stellt viele UDZ vor große Herausforderungen. Im Optimalfall kann zwischen den Detaillierungsgraden hin- und hergeschaltet werden. Dadurch wird für jede Analyse der passende LoD verwendet. Allerdings ist das Überführen eines 3D-Modells von einem LoD in einen anderen LoD oft kompliziert. Grundsätzlich gilt, dass der Wechsel in einen niedrigeren Detaillierungsgrad leichter erfolgt, da keine zusätzlichen Daten ergänzt werden müssen. Jedoch können beispielsweise aus einem LoD2-Modell verschiedene LoD1-Modelle erstellt werden. Abbildung 38 zeigt, welche unterschiedlichen LoD1-Modelle aus einem detailreichen

Gebäudemodell abgeleitet werden können. Welches LoD1-Modell am geeignetsten ist, unterscheidet sich erneut je nach Anwendungsfall. Ein UDZ sollte also nicht nur zwischen LoD-Stufen wechseln können, sondern auch innerhalb einer Stufe das jeweils richtige Modell auswählen. (Lehner und Dorffner 2020) (Biljecki et al. 2015) GeoAI kann bei der Wahl der optimalen LoD-Stufe helfen, indem die Anforderungen der jeweiligen Anwendung analysiert werden. Auch bei der Wahl der Transformationsmethode kann GeoAI helfen, das beste Modell auszuwählen. Allerdings gibt es in diesem Bereich noch keine Publikationen. Neben der Wahl des Detaillierungsgrades, kann GeoAI vor allem bei der Erstellung von Gebäudemodellen in unterschiedlichen Detaillierungsgraden hilfreich sein. (Döllner 2020; Paden et al. 2024)



Abbildung 38: verschiedene Konstruktions-Möglichkeiten für ein LoD1-Gebäude (Lehner und Dorffner 2020)

Die Konstruktion eines Basiszwillings auf Basis von vorab klar definierten 3D-Schemata ist ein zeitintensiver Ansatz, der häufig zu Fehlern im Modell führt. Ein Beispiel für solche Modelle sind 3D-Stadtmodelle im CityGML-Format mit einem Level of Detail von 3 oder höher. Oft haben solche Schemata ein Problem mit Datenlücken und Ausreißern, die nicht ins Schema passen. Egal wie sehr man den Algorithmus und das Schema zur Erstellung anpasst, werden immer Fehler auftreten, die eine regelmäßige Aktualisierung und das Arbeiten mit besonderen Objekten unmöglich machen. Hier kann die Analyse von 3D-Punktwolken mittels Machine- oder Deep Learning eine sinnvolle Alternative sein. Denn diese Verfahren sind nicht durch bestimmte Schemata eingegrenzt. Mit einem guten Algorithmus kann der ursprüngliche große Nachteil von Punkten, das Fehlen von semantischen Informationen, ausgemerzt werden. Dadurch kann eine flexible Verarbeitung der Eingangsdaten mit regelmäßiger Aktualisierung erfolgen. Ein weiterer Vorteil des ML-/DL-Ansatzes ist das Beibehalten der Rohdaten und der damit verbundenen Originalität des Modells und die leichtere Speicherung der Daten. Ohne

Weiterentwicklungen im Bereich der GeoAI wäre diese Alternative nicht möglich, da die Interpretation und Segmentierung der Punktwolken nicht so effizient durchgeführt werden könnten. Um ein gutes Ergebnis mittels ML/DL zu erreichen, sind gute Trainingsdaten notwendig. (Döllner 2020)

Um die “unsichtbare Bevölkerung” in die Arbeit mit UDZ zu integrieren, kann GeoAI sehr nützlich sein. (Coluccino et al. 2026) Ein Ansatz ist der Einsatz von Sprachsteuerung in Kombination mit Large-Language-Modellen oder anderer Künstlicher Intelligenz. So können Menschen, die aufgrund einer Beeinträchtigung oder mangelnder EDV-Bildung ein digitales System nicht bedienen können, dennoch teilnehmen. Anstatt Eingaben am Display wird mit einem Chatbot kommuniziert, der die Audio-Eingabe des Nutzers in Aktionen im UDZ umwandelt. Das System muss in der Lage sein, Audio in maschinenlesbaren Text und umgekehrt umzuwandeln. Bisher gibt es noch keine Publikationen im Zusammenhang mit UDZ, doch ein Einsatz wäre auf jeden Fall denkbar. (Basha et al. 2024) Alternativ kann der Output auch als Video gestaltet werden. Durch KI ist es möglich, den Output beliebig und vor allem massenhaft zu produzieren. Natürlich gibt es Einbuße in der Qualität, aber es kann für jeden Nutzer eine, je nach Fragestellung, individuelle Video- oder Audiodatei erstellt werden. (Watanabe et al. 2022) Aus dem Input “Wie komme ich von A nach B?” kann beispielsweise eine Verkehrsanalyse als Video-Output entstehen.

5.5 Fazit

In den Kapiteln 5.1 bis 5.4 werden zwölf Herausforderungen und ihre möglichen Lösungsansätze beleuchtet. Im praktischen Teil (Kapitel 8) dieser Arbeit wird einer dieser Lösungsansätze näher behandelt: das Fehlen semantischer Informationen. Dabei handelt es sich um eines der häufigsten Probleme bei 3D-Modellen. Den meisten 3D-Stadtmodellen mangelt es an semantischen Informationen. Dabei ist das Anreichern mit semantischen Informationen einer der ersten und wichtigsten Schritte in Richtung eines UDZ. Weitere Herausforderungen genauer zu behandeln, würden den Rahmen dieser Masterarbeit sprengen, weshalb der Fokus auf ein Problem gelegt wird.

6 UDZ in verschiedenen Städten

6.1 Digitaler GeoZwilling Wien

Die Stadt Wien hat sich das Ziel gesetzt, einen GeoZwilling der Stadt zu entwickeln. Dieser Zwilling soll alle bestehenden Rohdaten und eventuell noch weitere Datenquellen in einem semantischen, virtuellen 3D-Modell der Stadt zusammenfassen. In Kombination mit sozioökonomischen, ökologischen und anderen Daten kann ein CIM erstellt werden. Außerdem sollen auch ALS-Daten verwendet werden, um Datenlücken der terrestrischen und photogrammetrischen Vermessung auszufüllen. Bestehende Produkte, wie etwa das Digitale Geländemodell, sollen mit automatischen Workflows aus dem GeoZwilling erstellt werden. Für alle Produkte, die aus dem Zwilling abgeleitet werden, kann vollständige Kohärenz gewährleistet werden. Der GeoZwilling fungiert als Datenplattform, die geographische Produkte für Analysen und Simulationen zur Verfügung stellt. (Lehner und Dorffner 2020)

6.2 Digital Twin Zürich

Die Stadt Zürich veröffentlichte 2011 ein 3D-Stadtmodell, das helfen sollte, die Herausforderung der steigenden Bevölkerungszahl zu bewältigen. Das Züricher Modell besteht aus drei Ebenen: einem Digitalen Geländemodell, einem Blockmodell und einem Dachmodell. Zusätzlich wurden die Geometrien der Straßen hinzugefügt. Das Meshmodell für das Gelände wurde mittels FME erstellt und weist einen durchschnittlichen Höhenfehler von 30-40 cm auf. Das Blockmodell basiert auf Gebäudedaten des offiziellen Katasters, die mit photogrammetrischen Höhendaten (Dachtraufe, Dachgiebel) verknüpft wurden. Hier beträgt die durchschnittliche Abweichung bei den Höhen 50 cm. Die Positionsgenauigkeit weicht im Schnitt 10 bis 15 cm ab. Das Dachmodell wurde durch stereografische Luftbilder erstellt. Insgesamt beinhaltet das 3D-Modell mehr als 50.000 Gebäude und bildet die Basis für die Erstellung des UDZ der Stadt. Dieser Zwilling ist mit vielen Sensoren verknüpft und legt seinen Fokus auf die Bereiche Umwelt und Energie. (Alvi et al. 2025) Außerdem wurden ein Geoportal entwickelt, das das Sammeln und automatische Aktualisieren der Geodaten erleichtert. Dort kann man für einen beliebigen Bereich bestimmte Daten herunterladen. Dabei kann man aus über 200 räumlichen Datensätzen im Raster- und Vektorformat wählen (Schrotter und Hürzeler 2020). Der Digitale Zwilling kann in einem dreidimensionalen Viewer betrachtet werden. In diesem Viewer können aktuelle Informationen und Projekte dargestellt werden. (Shahat et al. 2021) Neben einem öffentlichen Viewer besteht auch die Möglichkeit "exclusive rooms" zu generieren. Diese sind nur für eine bestimmte Gruppe zugänglich und werden oft für Projektplanungen eingesetzt. Um die Bevölkerung besser in die urbane Planung zu integrieren, wurde der UDZ der Stadt in die Computerspielumgebung "Minecraft" überführt. So können Bürger ihre Ideen für die Stadt spielerisch umsetzen, ohne dass umfangreiche IT-Kenntnisse notwendig sind. Die Stadt Zürich setzt ihren UDZ in vielen verschiedenen Bereichen ein. Neben der urbanen Planung liegt der Fokus vor allem auf umweltrelevanten Themen

(Lärm, Luftqualität, Solarpotenzial, Handystrahlung). Für das räumliche Entwicklungskonzept wurde ein Modell erstellt, das die maximale Bebauung darstellt. Dieses konnte von Experten mit dem aktuellen Stand verglichen werden. Daraus leitete man das Entwicklungskonzept ab. Die Stadt Zürich will ihren UDZ weiter verbessern. Man will noch mehr dreidimensionale Daten generieren und diese besser miteinander verknüpfen. Außerdem will man sich mehr Richtung Echtzeit-Übertragung bewegen und das Modell häufiger updaten. Ein weiterer Fokus liegt auf dem Stadtmobiliar. Dieses soll systematisch erfasst und in den UDZ integriert werden. Zusätzlich will man den UDZ um die unterirdische Infrastruktur der Stadt erweitern. Zuletzt ist ein besseres Zusammenspiel mit GIS- und BIM-Systemen ein Ziel der Schweizer. (Schrotter und Hürzeler 2020)



Abbildung 39: Stadtmodell in der Videospielumgebung "Minecraft" (Schrotter und Hürzeler 2020)

6.3 Virtual Singapore

2011 wurde Singapur von neun starken Überflutungen heimgesucht. Aufgrund dieser Tragödien begann die Stadt, nach Möglichkeiten zu suchen, um solche Überflutungen in Zukunft früher zu erkennen. 2012 begann man, an einer 3D-Karte der Stadt zu arbeiten. Mithilfe dieser Karte sollten potenzielle Überflutungsgebiete früh erkannt werden. Für die Erstellung des Modells wurden Befliegungen mit 3D-Laserscannern durchgeführt. 2016 wurde das dreidimensionale Modell fertiggestellt. Im nächsten Jahr begann man Daten auf Straßenebene zu sammeln. Auf Fahrzeugen montierte Laserscanner nahmen Informationen über Straßen, Mobiliar, Brücken und Tunnel auf. Im nächsten Schritt wurden die Informationen der Befliegung und der Befahrung in einem passiven UDZ kombiniert. Um den UDZ weiterzuentwickeln, wurde eine Vielzahl sehr aktuellen

beziehungsweise Live-Daten integriert. Im UDZ von Singapur können viele Simulationen und Analysen durchgeführt werden. Ein Beispiel ist die Simulation von verschiedenen Verkehrsszenarien. Für diese Szenarien werden Verkehrssensoren und Wetterdaten von einem KI-Modell analysiert. Auch im Bereich der Mikroklima-Analyse, wurden mit der Simulationsplattform BESCAM, bereits große Fortschritte erzielt. Der UDZ von Singapur bietet neben der simplen Repräsentation auch die Möglichkeit zur Analyse bestehender Daten und zur Prognose künftiger Ereignisse. Damit entspricht er einem programmierbaren UDZ. (Alvi et al. 2025; Syamimi et al. 2020)

6.4 Kalasatama Helsinki

Die Hauptstadt Finnlands, Helsinki, startete ihre erste UDZ-Initiative im Kalasatama-Viertel. Dieser Bereich der finnischen Stadt soll als Vorzeigebispiel für eine intelligente Stadt fungieren. Das primäre Ziel des Projekts war das Erstellen und Publizieren eines hochwertigen 3D-Modells. Im nächsten Schritt soll dieses Modell beziehungsweise dieser UDZ für urbane Entwicklungsprojekte und das Testen neuer Technologien herangezogen werden. Es wurde zum einen ein semantisches Stadtmodell im Format CityGML und zum anderen ein aus Luftbildern und Laserscanndaten erstelltes Mesh-Modell für die Öffentlichkeit produziert. Für den zweiten Schritt wurde eine Applikation namens “Open Cities Planner” erstellt. Dieses Tool ermöglicht Stakeholdern das Testen von neuen Projekten im digitalen Raum. Aber auch die Auswirkung von externen Faktoren kann im UDZ des Viertels simuliert werden. So wurde beispielsweise getestet, wie sich das Viertel aufgrund von bestimmten Wetterbedingungen (Wind, Sonnenlicht) im Laufe der Zeit verändert. Des Weiteren wurden die möglichen Auswirkungen von Stürmen und das Solarpotenzial berechnet. (Hämäläinen 2020)

6.5 Herrenberg

Für den UDZ der Stadt Herrenberg wurden verschiedene Methoden eingesetzt. Das Ergebnis des Projekts ist ein Prototyp eines UDZ, der von verschiedenen Stakeholdern mit VR-Brillen interaktiv erlebt werden kann. Der Prototyp besteht aus einem 3D-Modell, einem Straßennetzwerk, einer Mobilitätssimulationssoftware (SUMO), einer Luftstrom- und Verschmutzungssimulation, empirischen Daten und Bildern. Das 3D-Stadtmodell wurde aus drei verschiedenen Datenquellen generiert. Einem digitalen Höhenmodell, BIM-Modellen und 3D-Laserscandaten. Der Detailgrad unterscheidet sich innerhalb des Stadtmodells. Gebiete und Gebäude, bei denen man von einer großen Auswirkung auf ihre Umgebung ausgeht, wurden genauer modelliert und teilweise durch BIM-Modelle ergänzt. Für die Darstellung des Straßennetzes wurden zwei verschiedene Methoden eingesetzt. Zum einen wurde die sogenannte NACH-Methode eingesetzt. Dieses Verfahren berechnet die Zentralität einzelner Straßenabschnitte. Dadurch kann das Nutzungs- bzw. Verkehrspotenzial dieser Straße vorausgesagt werden. Diese Analyse steht im Kontrast zu Zensusdaten, die die tatsächliche Nutzung des Verkehrsnetzes wiedergeben. Mit diesen beiden, sehr unterschiedlichen Varianten können verschiedene Szenarien mit dem aktuellen Status quo verglichen werden. Die Straßennetzsimulation

wurde mit einer Mobilitätssoftware erweitert. SUMO (Simulation of Urban Mobility) ist in der Lage, die Verkehrsbewegung einzelner Verkehrsteilnehmer dreidimensional im Stadtmodell darzustellen. Für das Projekt in Herrenberg wurde eine Erweiterung von SUMO entwickelt, die es möglich macht Simulationen und Visualisierungen der Veränderungen im Modal Split umzusetzen. Um empirische Daten aufzunehmen, wurde eine Applikation entwickelt. Diese App ermöglicht es Nutzern, ihre täglichen Routen und Aktivitäten zu tracken, Bilder und Tonaufnahmen zu verorten und Orte zu bewerten. Die persönliche Empfindung von öffentlichen Orten kann so sehr genau dokumentiert werden. Der gesamte UDZ wurde in ein Visualisierungs- und Simulationsprogramm namens COVISE integriert. Diese Software ermöglicht es mit einem Renderingmodul eine Umgebung für virtuelle Realität zu erstellen. (Dembski et al. 2020)

6.6 Probleme und Herausforderungen

Wissenschaftler sind sich einig, dass es einheitliche Dateiformate und Standards für UDZ geben sollte. Denn nur so kann es in Zukunft gelingen, UDZ miteinander zu verknüpfen. Außerdem können UDZ, die ähnlich aufgebaut sind und funktionieren, von neuen Entwicklungen anderer UDZ profitieren. Es gibt bereits Versuche, solche Standards zu integrieren. Die britische Regierung hat ein Projekt gestartet, dessen Ziel es ist, UDZ zu standardisieren. Diese Standards sollen alle UDZ in Großbritannien betreffen. Jedoch ist man aktuell noch sehr weit von einer einheitlichen Vorgehensweise entfernt. (Alvi et al. 2025) Die oben genannten Beispiele zeigen auch, dass sich jeder UDZ nur auf einige wenige Anwendungsgebiete spezialisiert. Die Analyse und Simulationen nahezu aller urbaner Prozesse in einem einzigen System ist für Städte aktuell nicht umsetzbar. Eine mögliche Lösung ist die Erstellung mehrerer UDZ, die für bestimmte Themenbereiche optimiert sind, aber dennoch miteinander kommunizieren und auf denselben Basisdaten aufbauen. Die Alternative ist ein System, das alles können soll und dementsprechend sehr leistungsstark sein muss. Ob ein solches System in naher Zukunft oder überhaupt realistisch ist, ist zu hinterfragen. (Yu und Merritt 2022)

7 Der Geozwilling der Stadt Linz

7.1 Status quo

Seit den 1990er-Jahren betreibt die Stadt Linz ein Geoinformationssystem, indem Geobasisdaten und Geofachdaten in zahlreichen Anwendungen mittels GIS verarbeitet, analysiert und visualisiert werden. Aktuell sind mehr als 80 verschiedene Geo-Datenschichten visuell miteinander verknüpft. Um diese Geodateninfrastruktur übersichtlich und effektiv betreiben zu können, wurde eine Geodatenplattform entworfen, die eine Vernetzung von Geodaten, Metadaten und Datendiensten sicherstellt. Der nächste Schritt ist das Schaffen einer Urbanen Datenplattform (UDP). Im Gegensatz zur Geodatenplattform sind hier auch vermehrt Nicht-Geodaten vertreten. Die UDP soll eine ganzheitliche Datenplattform für urbane Datensätze bilden. Die Daten sollen möglichst aktuell gehalten werden, und zusätzlich historisch gespeichert werden. Dadurch steigt das Nutzungspotenzial urbaner Daten für verschiedenste Akteure. Die UDP soll dazu führen, dass es weniger isolierte Fachanwendungen (Datensilos) gibt und Anwendungen sowie Daten miteinander vernetzt werden. Die Erstellung dieser Plattform ist ein laufender Prozess, der immer wieder vor Herausforderungen steht. Das Verknüpfen von verschiedenen Systemen ist vor allem bei Fachanwendungen, die seit Jahren in Anwendung sind, oft nicht einfach. Ziel ist es, vor allem bei neuen Anwendungen, Projekten und Daten darauf zu achten, eine Verknüpfung zur UDP herzustellen und keine Insellösung zu kreieren.

Die UDP steht in engem Zusammenhang mit der Erstellung eines UDZ. In Linz wird dieser "Geobasis-Zwilling Linz" genannt. Die UDP bildet die Grundlage für den UDZ, der dafür sorgen soll, die Geodaten der Stadt Linz räumlich, transparent und nachhaltig zu präsentieren. Bürger sollen so einen einfacheren Informationszugang erhalten. Der UDZ soll als transparentes, interaktives Kommunikationstool dienen. Fachbereiche können die Daten des UDZ für fachspezifische Simulationen nutzen. Außerdem soll der UDZ Redundanzen vermeiden. Auf Führungsebene dient der UDZ als Entscheidungshilfe und zur verbesserten Vermittlung von Entscheidung an die Bürger. Ziel des "Geobasis-Zwilling" ist alles darzustellen, was in der realen Welt sichtbar ist und sich nicht bewegt. Dazu zählen Bauwerke, Grünflächen, der Straßenraum, Gewässer und das Stadtmobiliar. Der Geobasis-Zwilling bildet die Grundlage für Fachzwillinge. In diesen Fachzwillingen können dann Sensoren verknüpft und fachspezifische Simulationen durchgeführt werden. Beispiele für Fachzwillinge sind der Planungszwilling oder der Umweltzwilling. Im Planungszwilling können geplante Projekte und deren Auswirkungen visualisiert werden. Im Umweltzwilling werden umweltrelevante Daten, die nicht sichtbar sind, ergänzt (Temperatur, Feinstaub, Wind etc.).

7.2 Bausteine

Wie viele andere Städte hat sich Linz dazu entschieden, den UDZ auf einem Bausteinsystem aufzubauen. Als zentraler Baustein, der mit allen anderen Bausteinen direkt verknüpft ist, fungiert das GIS. Weitere Bausteine sind das 3D-Stadtmodell, das Adress-, Gebäude- und Wohnungsregister (AGWR), die Gebäude-Informations-Karte (GIK) und die Naturbestandsdaten. Ein zukünftiger Baustein soll beispielsweise ein Straßenraum-Zwilling sein.

Das GIS der Stadt Linz ist seit über 25 Jahren in Betrieb und dient der Vernetzung, Verwaltung, Visualisierung und nachhaltigen Speicherung von Geodaten und deren Metadaten. Verschiedene Dienste gewährleisten den einfachen und nutzungsbringenden Umgang mit diesen Daten. Das GIS ist eng mit der UDP verknüpft und ermöglicht durch verschiedene Workflows Analysen in diversen Bereichen (Baumkataster, Pflegeflächen, Liegenschaften etc.). Das GIS der Stadt wird laufend aktualisiert und um zusätzliche Datenschichten erweitert. Das wichtigste Tool zur Visualisierung der GIS-Daten ist das WebGIS der Stadt. Hier können Mitarbeiter des Magistrats verschiedene Datenquellen der UDP visualisieren und Abfragen durchführen. Weitere Anwendungen sind Geo-Dashboards und öffentliche Kartendarstellungen für die Bürger Linz, die mit der UDP verknüpft sind und laufend aktualisiert werden. So können beispielsweise die E-Scooter-Fahrten tagesaktuell abgefragt werden.

Das 3D-Stadtmodell bildet einen weiteren Baustein. Das Modell ist seit 2011 in Betrieb und wurde 2020 veröffentlicht. Es dient vor allem der Visualisierung der Stadt und ihrer Sehenswürdigkeiten. Außerdem sind einfache Analysen (z. B. Schattensimulation) möglich. Ein Nachteil des 3D-Modells ist der Aktualisierungszeitraum und die fehlende Semantik in den Daten. (mehr Infos 8.1.1)

Der Baustein AGWR hängt eng mit der Gebäudeinformationskarte (GIK) zusammen. Das AGWR ist mit vielen Datensätzen verknüpft und kann wertvolle Informationen für die GIK oder andere Bausteine und Anwendungen liefern. So ist das AGWR beispielsweise mit der Energieausweis-Datenbank der Bundesländer verknüpft. Die GIK ermöglicht es, die Gebäude "intelligenter" zu machen, indem es Attribute aus dem AGWR oder anderen Quellen mit den Gebäudepolygonen verschneidet. Ziel ist es, aus der aktuellen 2D-GIK 3D-Gebäude abzuleiten (mehr Infos 8.1.1).

Um zukünftige Bausteine zu ermöglichen, wird daran gearbeitet, die Vermessung des Naturbestands in Zukunft datenbankbasiert durchzuführen. Der Naturbestand wird dann flächenhaft vermessen und direkt in die Datenbank eingespielt, die die Daten mit anderen Bausteinen verbindet. Bis dato sind dafür noch Zwischenschritte notwendig.

7.3 “Intelligente” Gebäude

Dank der GIK liegen nun geometrisch korrekte Gebäude mit einigen Attributen aus dem AGWR vor. Ziel ist es, die Gebäude mit weiteren Attributen und semantischen Informationen anzureichern. Dazu sollen verschiedene Datenquellen genutzt werden. Dann können Gebäude nach verschiedenen Attributen klassifiziert werden. Um Gebäuden allerdings semantische Informationen zuzuweisen, die in keiner Datenquelle vorhanden sind (z. B. INSPIRE-Gebäudeklasse) reicht die reine Verschneidung nicht aus. Ziel des praktischen Teils ist es herauszufinden, ob ein ML-Algorithmus dazu in der Lage ist, Gebäude anhand ihrer Attribute automatisch zu klassifizieren. Wichtig anzumerken ist, dass die Klassifikationsart des praktischen Teils nur eine von vielen Möglichkeiten ist, Gebäude zu klassifizieren (Übersicht siehe Kapitel 8.4.1). Für eine andere Variante der Klassifikation müssten der Trainingsdatensatz und die Auswahl der Attribute angepasst werden.

8 KI-gestützte Klassifizierung von Gebäuden

Ziel des praktischen Teils ist die semantische Anreicherung von Gebäudedaten. Dies geschieht mithilfe eines KI-Algorithmus. Es wird überwachtes Machine Learning eingesetzt, um die Gebäudepolygone der Stadt Linz einer von fünf Gebäudekategorien zuzuordnen. Die Kategorien entsprechen dem INSPIRE-Standard. Ziel ist es herauszufinden, in welcher Qualität der ML-Algorithmus die Gebäude klassifiziert. Dabei wird darauf eingegangen welche Faktoren das Ergebnis maßgeblich beeinflussen.

8.1 Verwandte Theorien

Es gibt einige Theorien, die Gebäudekategorien direkt aus Fernerkundungsdaten ablesen. Dafür werden meist CNN verwendet. Der Vorteil dieser Verfahren ist, dass sie direkt aus Luftbildern die Gebäude und seit einigen Jahren auch Gebäudekategorien ableiten. Jedoch ist für ein funktionierendes CNN ein umfangreicher Trainingsdatensatz notwendig und es werden lediglich Faktoren berücksichtigt, die am Bild zu erkennen sind. Attributive Aspekte werden in diesen Varianten nicht verwendet. Ein weiteres Problem ist, dass in einigen Verfahren die Segmentierung nicht von der Klassifizierung getrennt werden kann. Auch die Genauigkeit lässt in vielen Fällen zu wünschen übrig und die zu verarbeitende Datenmenge ist enorm. Hertrich (Hertrich 2024) hat in seiner Arbeit bestehende Publikationen zu diesem Thema miteinander verglichen. (Abbildung 40) (Hertrich 2024) Alternativ können Gebäudetypen auch mit Hilfe von regelbasiertem Lernen erkannt werden. (Hussain und Shan 2016) Bei solchen regelbasierten Verfahren ist allerdings eine sehr gute Datengrundlage notwendig und Ausreißer werden oft fehlerhaft bewertet. Alternativ kann unüberwachtes Machine Learning eingesetzt werden. (Hertrich 2024) Hier ist das große Problem, dass die Gebäudekategorien nicht vorab festgelegt werden können. Zu den häufigsten Methoden zählt das überwachte ML, das auch in dieser Arbeit Anwendung findet. Mit dieser Variante kann eine hohe Genauigkeit erzielt werden und die Rechenkapazität hält sich in Grenzen. Wichtig für dieses Verfahren ist ein qualitativ hochwertiger Ground-Truth-Datensatz. Ein Nachteil dieses Verfahrens ist, dass die vom Algorithmus erkannten Muster nicht in allen Städten, sondern nur in Städten mit ähnlichen Strukturen und ähnlichen Datengrundlagen funktionieren. In dieser Arbeit wird überwachtes ML eingesetzt, weil es die Rechenzeit begrenzt hält und separat von der Gebäudesegmentierung stattfindet. Die bestehenden Publikationen verwenden unterschiedliche Gebäudeklassen, die meist nicht mit üblichen Formaten für UDZ (CityGML) kompatibel sind oder Gebäude nicht nach Nutzung, sondern nach anderen Aspekten klassifizieren (z. B. Klima, Alter). (Kim et al. 2022) Andere Arbeiten legen den Fokus auf POI-Daten, um die Nutzung der Gebäude zu klassifizieren. Bei diesen Beispielen ist zwar die Genauigkeit für Gebäude mit POIs hoch (z. B. Bildungseinrichtungen – 96 %), Wohngebäude oder Verkaufsgebäude werden allerdings schlecht erkannt (79 % & 82 %). (Hu et al. 2026) Das Ziel dieser Arbeit ist es Gebäude so zu klassifizieren, dass die Klassen direkt in die CityGML-Datenstruktur eingefügt und die Gebäude in einem UDZ dargestellt werden können.

Autor	Klassen	Algorithmus	Trennung	Daten (Herkunft, Auflösung)	Gütemaße
HOFFMANN et al. (2021)	Wohngebäude, Nicht-Wohngebäude	Bayesian FCN (U-Net ähnlich)	nein	Luftbilder (Google Maps, ~ 0,6 – 2,4 m)	IoU je Klasse: 0,036 – 0,706
DROIN et al. (2020)	Wohngebäude, Nicht-Wohngebäude, Nebengebäude	FCN-vgg19	nein	Luftbilder (o. A., 0,4 m)	OA: 0,93 % Kappa: 0,73
DIMASSI et al. (2021)	Wohngebäude, Nicht-Wohngebäude	RexNet und 4 andere CNNs	ja	Satellitenbilder (Mapbox, ~ 0,5 m)	OA: 0,93 – 0,95 %
STRELTSOV et al. (2020)	Wohngebäude, Wirtschaftsgebäude	Segmentierung: U-Net Klassifizierung: ResNet-152	ja	Luftbilder (USGS, 0,3 m)	Segmentierung: mIoU 0,76 und 0,81 Klassifizierung: IoU 0,99 und 0,74
HOFFMANN et al. (2019)	Wirtschaftsgebäude, Wohngebäude, öffentliche Gebäude, Industriegebäude	Inceptionv3, VGG16 und diverse Fusionsstrategien	nein	Luftbilder (BingMaps, ~ 0,3 – 2 m) + GSV	Nur Luftbilder: F1: 0,51 – 0,66 Luftbilder + GSV: F1: 0,68 – 0,73
CAO & QIU (2018)	Ein- und Zweifamilienhäuser, begehbare Mehrfamilienhäuser, Mehrfamilienhäuser mit Aufzug, gemischte Wohn- und Geschäftsgebäude, Geschäfts- und Bürogebäude, Industrie- und Produktionsgebäude, Verkehrs- und Versorgungseinrichtungen, öffentliche Einrichtungen und Institutionen, + 3 Landnutzungsklassen	Basiert auf SegNet und Teilen von VGG16	nein	Luftbilder (BingMap, 0,3 m) + GSV	Nur Luftbilder: mIoU: 0,47, Genauigkeit: 0,78 Luftbilder + GSV: mIoU: 0,48, Genauigkeit: 0,78
PAN et al. (2020)	Alte Gebäude, alte Fabriken, Gebäude mit Eisendach, neue Gebäude	U-Net	nein	Satellitenbilder (Worldview-2, 2 m durch pan-sharpening 0,5 m)	OA: > 0,83 %, IoU: > 0,86, F1: 0,63 – 0,88
HUANG et al. (2021)	Niedrige Wohngebäude, mittelhohe Wohngebäude, hohe Wohngebäude, Industriegebäude, + 5 andere Klassen (Bodenbedeckung + Schatten)	M2-3-DCNN	nein	Satellitenbilder (Ziyuan 3, 2,1 und 3,5 m)	Genauigkeit der Klassen: 0,679 – 968

Abbildung 40: Studien zur semantischen Segmentierung und Klassifizierung von Gebäudetypen (Hertrich 2024)

(Hoffmann et al. 2021; Hoffmann et al. 2019; Droin et al. 2020; Dimassi et al. 2021; Streltsov et al. 2020; Cao und Qiu 2018; Pan et al. 2020; Huang et al. 2021)

8.2 Methodik

Abbildung 41 skizziert den Ablauf des praktischen Teils. Zu Beginn steht die Erstellung der verschiedenen Gebäude-Datensätze. Hier besteht ein großer zeitlicher Aufwand bei der händischen Klassifizierung der Gebäude für den Trainings- und Validierungsdatensatz. Vor diesem Schritt musste noch die Wahl der Klassifizierungsart festgelegt werden, die in Kapitel 8.4.1 genauer behandelt wird. Die restlichen, unklassifizierten Gebäude werden zu einem späteren Zeitpunkt des Workflows vom trainierten ML-Algorithmus einer der fünf INSPIRE-Kategorien zugewiesen. Mit den händisch gelabelten Daten erfolgt ein Trial-and-Error-Prozess, um die optimalen Rahmenbedingungen für die automatische Klassifizierung zu finden.

Wie bereits in Kapitel 8.1 erwähnt, wird für die semantische Anreicherung der Gebäude ein ML-Ansatz angewendet. Dabei sind zwei Aspekte essenziell, um ein optimales Ergebnis zu erzielen. Zum einen die Auswahl der Attribute, mit denen die Gebäude angereichert und die ML-Algorithmen trainiert werden, und zum anderen die Wahl der ML-Algorithmen selbst und deren Parametereinstellungen. Diese beiden Schritte erfordern einige Testdurchläufe und werden in Kapitel 8.5 näher beschrieben. Wichtig anzumerken ist, dass diese Testdurchläufe nur mit den händisch gelabelten Daten durchgeführt werden, da hier eine Validierung möglich ist. Am Ende dieses sehr aufwendigen Schrittes steht das vorerst fertige Modell, mit dem anschließend das gesamte Stadtgebiet klassifiziert werden kann.

Am Ende wird das Ergebnis für die gesamte Stadt noch stichprobenartig überprüft und visualisiert. Außerdem wird darauf eingegangen, wie andere Städte diese Methode anwenden können.

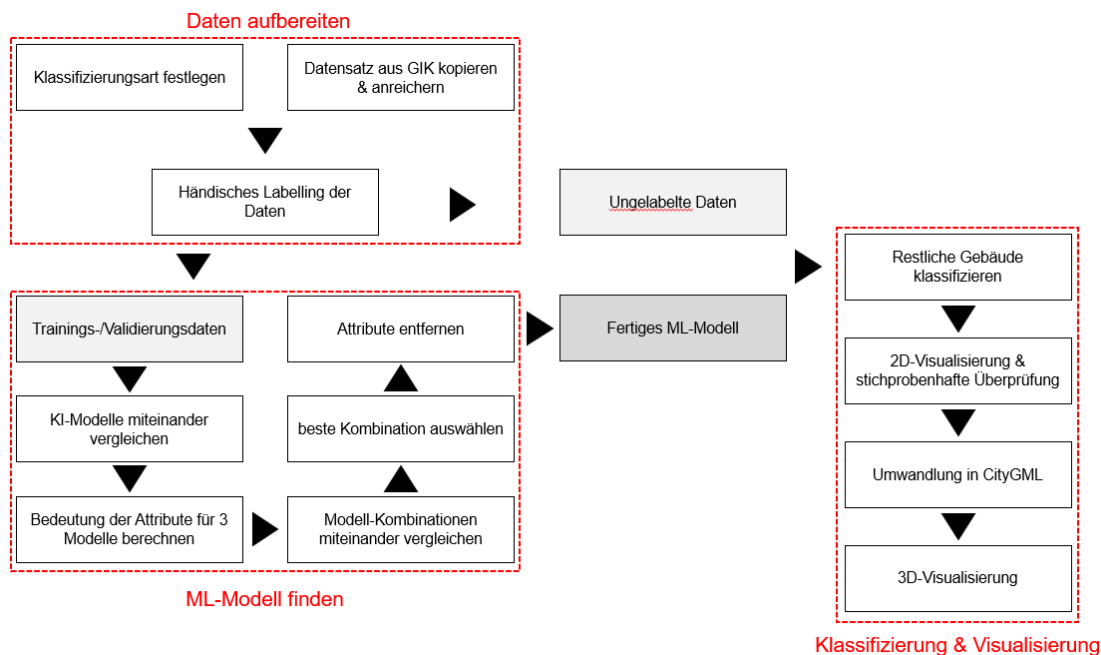


Abbildung 41: Workflow praktischer Teil

8.3 Datengrundlage

8.3.1 Gebäude

Datengrundlage für die automatische Klassifizierung sind die Gebäude der Gebäude-Informationen-Karte (GIK) der Stadt Linz. In diesem Datensatz sind mehr als 76.000 Gebäude als Polygone enthalten. Diese Gebäude werden als 2D-Polygone verarbeitet und klassifiziert. Alternativ wurde auch die Verwendung von 3D-Daten aus dem 3D-Stadtmodell der Stadt Linz in Erwägung gezogen. Allerdings sind diese Daten sehr schlecht gegliedert und werden nur unregelmäßig aktualisiert. Ein Problem bei der Gliederung sind beispielsweise übereinanderliegende Polygone, die das Verschneiden mit 2D-Daten verkomplizieren. Im Gegensatz dazu haben die GIK-Gebäude eine wesentlich bessere geometrische Qualität, eine hohe Aktualität und liegen adressscharf vor. Ein Gebäudepolygon kann also nur eine Adresse haben und jedes Polygon im Datensatz ist einer Adresse zugewiesen. Dadurch können manche Attribute, die auf Adressebene vorliegen, mit den Gebäuden verknüpft werden. Wichtig anzumerken ist jedoch, dass nicht jede Adresse, sondern jedes Gebäude klassifiziert wird. Es können also mehrere Gebäude aus unterschiedlichen Kategorien einer Adresse zugeordnet sein. Das Ziel der Stadt Linz ist es, die 2D-GIK-Gebäude in 3D-Gebäude umzuwandeln. Es gibt hierzu bereits erste Ergebnisse im LoD1 und LoD2. Das Klassifizieren der 2D-Daten und spätere Umwandeln in 3D-Gebäude verringert auch die Rechenzeit bei der Klassifizierung. Da die 3D-Gebäude mit der GIK als Datengrundlage noch nicht

flächendeckend vorliegen, können manche dreidimensionale Aspekte noch nicht für das ML verwendet werden. Dazu zählt beispielsweise die Dachform der Gebäude.

8.3.2 AGWR

Das “Adress-, Gebäude- und Wohnungsregister”(AGWR) umfasst zwei Register, die getrennt geführt werden. Zum einen das Adressregister und zum anderen das Gebäude- und Wohnungsregister. Für jede Adresse gibt es eine Vielzahl an Attributen im AGWR. Ein Teil dieser Attribute (Tabelle 4) wurde für die Klassifizierung der Gebäude verwendet. In Grün sind die Attribute markiert, die analysiert wurden und für das finale Modell verwendet werden. In Rot werden Attribute dargestellt, die nach der Analyse nicht im Modelltraining verwendet wurden. Dazu zählt unter anderem die Gebäudeeigenschaft. Hier wird für jede Adresse eine Gebäudeeigenschaft gemäß EU-Gebäudeklassifikation zugewiesen. Es gibt jedoch keine Verpflichtung zur Aktualisierung dieses Attributs und die Zuweisung erfolgt auch nur für das Gebäude, auf dem der Adresspunkt liegt. Eine Übertragung der Gebäudeeigenschaft auf alle zur Adresse gehörenden Gebäude macht keinen Sinn, da sich die Nutzung oft sehr stark unterscheidet. So haben beispielsweise ein Pfarrheim, in dem jemand wohnt und die dazugehörige Kirche denselben Adresspunkt. Liegt der Adresspunkt auf dem Pfarrheim wird die Adresse als “Gebäude mit einer Wohnung” oder “Gebäude mit 2 oder mehr Wohnungen” klassifiziert. Die richtige EU-Gebäudeklassifikation für das Kirchengebäude ist jedoch “Kirchen, sonstige Sakralbauten”. Deshalb macht eine automatische Klassifizierung auf Gebäudeebene auf jeden Fall Sinn, da solche Beispiele separat betrachtet und klassifiziert werden können. Es werden jedoch einige der Attribute aus dem AGWR für die Klassifizierung verwendet. (Statistik Austria 2022a)

Tabelle 4: verwendete Attribute aus dem AGWR

Beschreibung	Attributname
Gebäudeeigenschaft	GDAEIGENS
Wohnzweck	WOHNZW
Adresscode	ADRCD
Hauptgebäude der Adresse	ADRHAUPTGB
Postleitzahl	PLZ
Gebäudeunterscheidung	GEBUNTERSC
Straßenname	STRASSE
Hausnummer	HAUSNUMMER
Straßenkennzahl	SKZ
Nutzung der Adresse	NUTZUNG

8.3.3 True Orthofoto

Im Gegensatz zu einem normalen Orthofoto sind True Orthofotos ein besseres Produkt, da sie die tatsächliche Position und Ausdehnung der Gebäude und sonstiger Flächen zeigen. Dadurch wird auch die unmittelbare Umgebung rund um das ganze Gebäude sichtbar. (Fras et al. 2021) Das verwendete Digitale Orthofoto (DOP) von Linz wurde auf Basis von Luftbildern in Kombination mit einem Gelände- und Oberflächenmodell erstellt. Anschließend wurde für jedes Gebäude der jeweilige Ausschnitt aus dem Orthofoto ausgeschnitten und auf Bildeigenschaften untersucht. (mehr Info 8.4.4)

8.4 Vorverarbeitung der Daten

8.4.1 Art der Kategorisierung

Im ersten Schritt muss die Art der Kategorisierung festgelegt werden. Es wurden mehrere Möglichkeiten untersucht, Gebäude zu klassifizieren. Tabelle 5 zeigt die untersuchten Kategorisierungsarten und deren Eliminationsgründe. Besonders wichtig für die semantische Anreicherung ist die Kompatibilität mit dem CityGML-Format, da dies das häufigste Austauschformat für 3D-Modelle und UDZ ist. Zwei dieser Klassifikationsarten stellten sich als geeignet heraus, da sie häufig in Kombination mit CityGML zum Einsatz kommen. Zum einen die INSPIRE-Gebäudeklassen und zum anderen die SIG3D-Klassen. Abbildung 42 zeigt die einzelnen Gebäudeklassen der beiden Standards. Am Ende fiel die Wahl auf den INSPIRE-Standard, da dieser weniger Klassen hat und somit weniger Gebäude händisch klassifiziert werden müssen. Ein Training auf alle SIG3D-Klassen wäre wesentlich aufwendiger und würde den Rahmen dieser Arbeit überschreiten. Generell sind viele Klassen in den beiden Standards gleich (Wohnen, Landwirtschaft, Industrie, Nebengebäude). Nur die INSPIRE-Klasse "COMMERCEandSERVICES" wird in SIG3D in fünf verschiedene Klassen aufgeteilt. Dadurch hält sich auch der Aufwand in Grenzen, falls man zu einem späteren Zeitpunkt auf SIG3D umstellen will. Denn dann müsste man nur die als "COMMERCEandSERVICES" bezeichneten Gebäude im Trainingsdatensatz ändern und auf die verschiedenen SIG3D-Klassen aufteilen. Natürlich müssten in diesem Fall zusätzliche Gebäude für die neuen Kategorien händisch klassifiziert werden.

Tabelle 5: untersuchte Klassifizierungs-Standards

Standard	Eliminierungsgrund	Quelle
INSPIRE	keiner	(INSPIRE Maintenance and Implementation Group (MIG) 2024)
SIG3D	zu viele Klassen	(SIG3D AG Qualität 2014)
OpenStreetMap Tags	keine Standardisierung	(Open Geospatial Consortium 2021)
CORINE Land Cover	Fokus auf Flächen	(European Environment Agency 2021)
IFC	Kein Fokus auf Nutzungsarten	(buildingSMART International 2024)

BauNVO	National begrenzt (Deutschland)	(Bundesministerium für Justiz 2023)
Urban Atlas (Copernicus)	Fokus auf Flächen	(Land Monitoring Service 2026)
Nationale Statistikämter (Bsp. AGWR)	National begrenzt	(Statistik Austria 2022b)

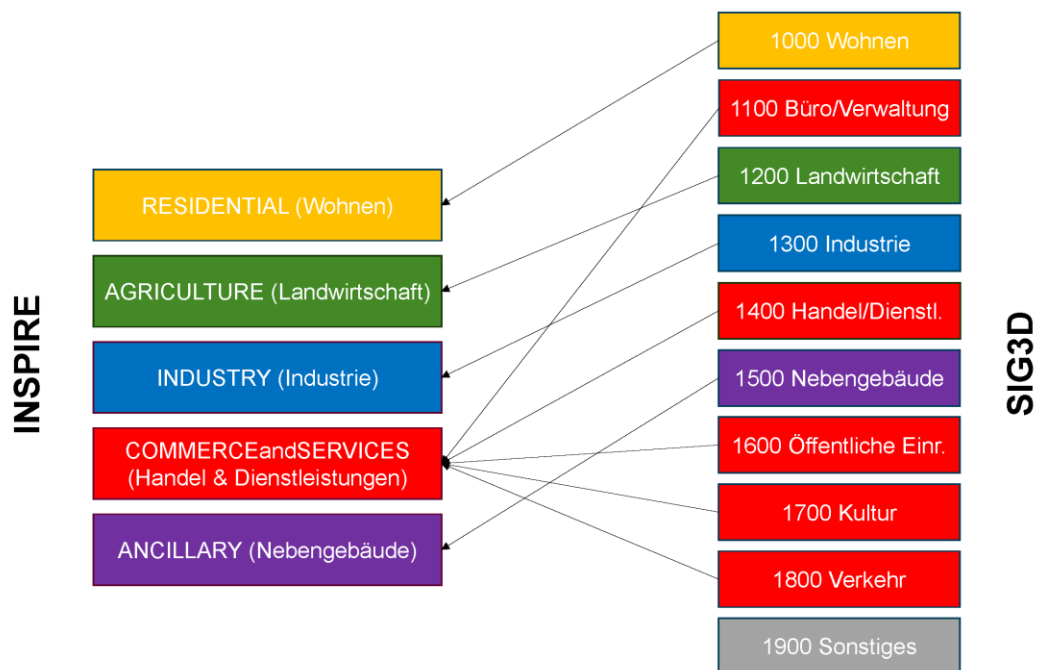


Abbildung 42: INSPIRE-/SIG3D-Gebäudekategorien

8.4.2 Händische Klassifizierung

Insgesamt wurden 615 Gebäude händisch einer Kategorie zugewiesen (Tabelle 6). Die Zuweisung erfolgte mittels QGIS. Dabei wurde darauf geachtet, von jeder Kategorie mindestens 100 Gebäude zu labeln. Eine Ausnahme bilden die landwirtschaftlichen Gebäude, da es nur sehr wenige Gebäude dieser Kategorie im urbanen Raum gibt. Am häufigsten wurden Wohn-, Neben- und Handels- und Dienstleistungsgebäude klassifiziert, da es auf den ersten Blick am meisten Gebäude dieser Kategorien gibt.

Tabelle 6: Aufteilung der händisch gelabelten Gebäude

INSPIRE-Gebäudeklasse	Händisch gelabelte Gebäude
RESIDENTIAL	165
AGRICULTURE	18
INDUSTRY	102
COMMERCEandSERVICES	170
ANCILLARY	160
No Label	75.433
Total	76.051

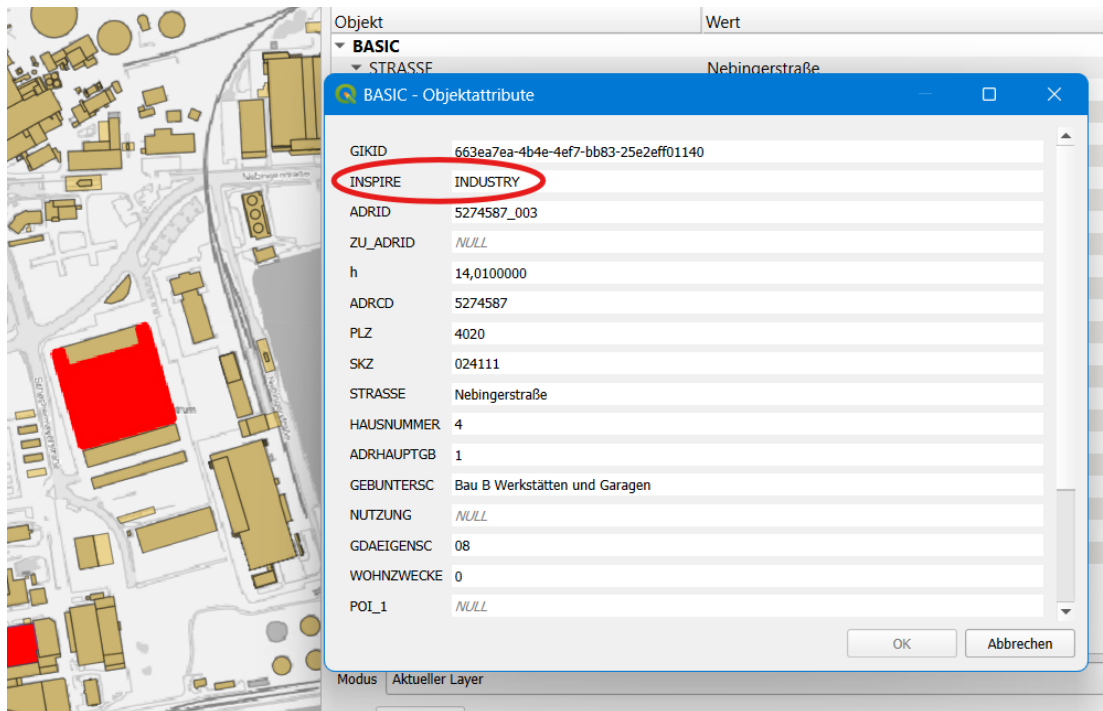


Abbildung 43: Händisches Labelling der Trainings- und Validierungsdaten

8.4.3 Datensätze

Nach dem händischen Labeln gab es zwei verschiedene Datensätze. Einen Datensatz mit allen händisch zugewiesenen Objekten (618) und einen Datensatz mit allen ungelabelten Gebäuden (75.433). Für die Erstellung des Algorithmus wurden die händisch zugewiesenen Objekte herangezogen. Dazu wurden 80 Prozent der 618 Gebäude als Trainingsdaten und 20 Prozent als Validierungsdaten eingesetzt. Wobei Cross-Validation angewendet wurde (Kapitel 8.6), und somit nicht immer dieselben Gebäude zum Trainieren beziehungsweise Validieren verwendet wurden. Abbildung 43 zeigt die Aufteilung der Gebäude in die verschiedenen Datensätze.

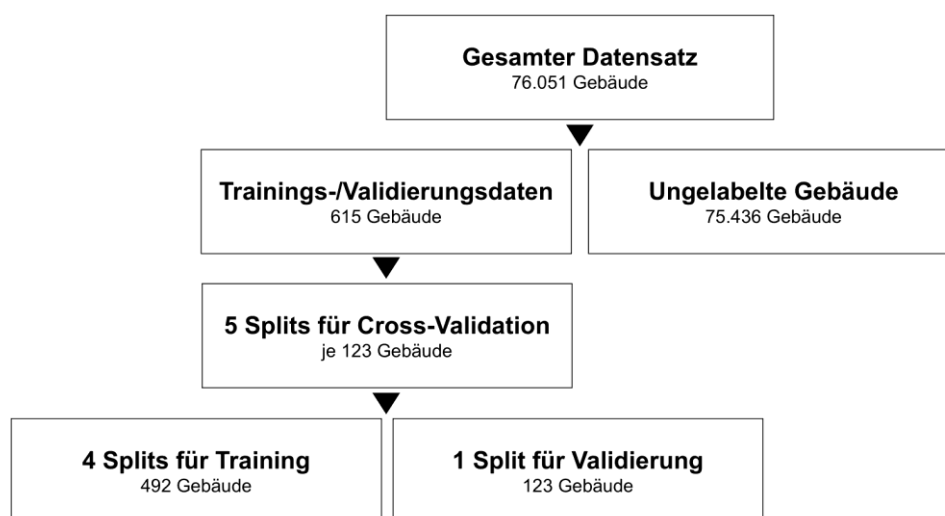


Abbildung 44: Generierung der verschiedenen Datensätze

8.4.4 Anreicherung mit zusätzlichen Attributen

Alle Gebäude, ob Trainings-, Validierungs- oder ungelabelte Daten, werden mit Informationen aus anderen Datengrundlagen (Orthofoto, AGWR etc.) verknüpft und angereichert. Außerdem werden verschiedene Attribute berechnet. Dazu zählen beispielsweise die Fläche oder auch der Umfang der Polygone. Um herauszufinden, welche Attribute bei der Klassifizierung hilfreich sind und wie der Algorithmus aufgebaut werden soll, werden zu Beginn nur die Trainings- und Validierungsdaten bearbeitet. Die FULL-Daten werden erst nach einer Festlegung der gewünschten Attribute angereichert und anschließend dem ML-Algorithmus übergeben. Dadurch kann die Rechenzeit geringer gehalten werden. Im Kapitel 8.5.2 werden die Attribute, die in Erwägung gezogen wurden, näher beschrieben. In Kapitel 8.5.4 wird auch erklärt, warum manche dieser Attribute nicht für die finale Klassifizierung verwendet wurden. Die meisten Arbeitsschritte zur Anreicherung der Gebäude erfolgten in Python. Lediglich die Informationen aus dem AGWR und der GIK waren bereits vorher Bestandteil des Datensatzes. Die Attribute aus dem AGWR wurden mittels FME (Feature Manipulation Engine) mit den GIK-Gebäuden verbunden.

8.5 Modelltraining und Klassifikation

Dieses Kapitel beschreibt den Prozess näher, mit dem die am besten geeigneten Attribute und ML-Algorithmen gefunden wurden. Zuerst werden die Ergebnisse der unterschiedlichen ML-Methoden näher beschrieben (8.5.1). Anschließend wird auf die Bedeutung der verschiedenen Attribute eingegangen (8.5.2). In Kapitel 8.5.3 werden mehrere ML-Modelle miteinander kombiniert, und im Kapitel 8.5.4 wird herausgefunden, welche Attribute für das finale Modell nicht notwendig sind.

8.5.1 Modell-Evaluierung

Um das bestmögliche ML-Modell zu finden, ist eine genaue Validierung erforderlich. In dieser Arbeit wurde Cross-Validation eingesetzt, um die Genauigkeit der verschiedenen Modelle zu bewerten. Der händisch gelabelte Datensatz wurde in fünf Teile aufgeteilt, wobei jeweils vier Teile zum Training und ein Teil zur Validierung dienen. Im nächsten Validierungsdurchlauf wird ein anderer der fünf Teil-Datensätze zur Validierung eingesetzt. Am Ende ergeben sich so fünf Durchgänge mit fünf Genauigkeitswerten. Der Mittelwert der fünf Genauigkeitswerte ist die Genauigkeit des Modells.

Um das beste ML-Modell zu finden, wurden zunächst mehrere bekannte Modelle getestet. Es wurden auch unterschiedliche Parametereinstellungen miteinander verglichen. Die Ergebnisse dieser Validierungen werden in Tabelle 7 dargestellt. Die Modelle wurden in diesem Schritt mit allen Attributen trainiert.

Tabelle 7: Performance der verschiedenen Modelle

							Genauigkeit				F1-Scores				Mean F1 pro Klasse				
Modell	n_estimators	random_state	n_jobs	class_weight	Attribute	Splits	Ø	Std	Max	Min	Ø	Std	Max	Min	AGR	ANC	COS	IND	RES
RandomForest	500	42	1	balanced	alle	5	84,6%	1,6%	86,3%	82,1%	79,9%	3,7%	85,0%	75,3%	63,1%	90,0%	81,1%	75,2%	89,7%
RandomForest	250	42	1	balanced	alle	5	84,6%	1,8%	86,3%	81,3%	79,7%	3,5%	85,0%	75,3%	63,1%	89,7%	81,1%	74,4%	90,3%
RandomForest	200	42	1	balanced	alle	5	85,1%	1,9%	87,9%	82,1%	80,2%	3,7%	85,0%	75,4%	63,1%	90,2%	81,7%	75,2%	90,7%
RandomForest	150	42	1	balanced	alle	5	84,8%	1,4%	87,1%	82,9%	80,0%	3,9%	84,4%	74,3%	63,1%	90,0%	81,0%	75,0%	90,4%
RandomForest	100	42	1	balanced	alle	5	84,1%	1,8%	86,3%	81,3%	79,4%	3,8%	84,4%	73,9%	63,1%	89,4%	80,0%	74,2%	90,3%
RandomForest	50	42	1	balanced	alle	5	83,6%	2,1%	87,1%	80,5%	79,0%	3,7%	83,2%	73,6%	63,1%	88,9%	79,4%	73,4%	89,7%
ExtraTrees	500	42	1	balanced	alle	5	78,6%	2,8%	81,5%	74,0%	71,2%	5,6%	76,1%	60,2%	50,5%	87,4%	73,3%	55,5%	89,1%
ExtraTrees	250	42	1	balanced	alle	5	78,5%	2,7%	82,3%	75,0%	70,1%	6,0%	77,6%	59,9%	50,5%	87,4%	72,6%	55,4%	89,1%
ExtraTrees	200	42	1	balanced	alle	5	78,1%	3,1%	84,5%	73,2%	70,8%	5,5%	76,1%	60,3%	50,5%	87,7%	71,8%	55,0%	88,8%
ExtraTrees	150	42	1	balanced	alle	5	78,0%	2,1%	80,6%	74,8%	71,8%	3,1%	75,5%	67,1%	57,1%	87,7%	71,8%	54,1%	88,3%
ExtraTrees	100	42	1	balanced	alle	5	78,6%	2,9%	81,5%	73,2%	72,5%	2,0%	76,2%	70,4%	57,1%	88,5%	71,7%	56,8%	88,6%
ExtraTrees	50	42	1	balanced	alle	5	78,2%	3,1%	81,5%	73,1%	72,5%	3,2%	77,7%	68,4%	63,1%	88,7%	72,2%	50,5%	88,2%
BaggingClassifier	500	42	1		alle	5	85,9%	1,5%	87,9%	83,7%	83,5%	2,4%	86,6%	79,1%	74,6%	89,1%	82,4%	81,6%	90,0%
BaggingClassifier	250	42	1		alle	5	86,6%	1,5%	88,7%	84,6%	85,0%	1,8%	87,2%	83,0%	80,3%	90,0%	83,2%	81,2%	90,4%
BaggingClassifier	200	42	1		alle	5	86,6%	1,5%	88,7%	84,6%	85,4%	1,4%	87,5%	83,3%	82,4%	89,5%	83,6%	80,7%	90,6%
BaggingClassifier	150	42	1		alle	5	86,7%	1,7%	88,7%	84,6%	85,5%	1,6%	87,5%	83,3%	82,4%	89,7%	83,6%	80,7%	90,1%
BaggingClassifier	100	42	1		alle	5	86,1%	1,4%	87,9%	83,7%	84,3%	1,2%	86,4%	83,2%	78,1%	89,0%	82,0%	80,6%	91,6%
BaggingClassifier	50	42	1		alle	5	86,1%	1,4%	87,9%	83,7%	84,1%	1,2%	86,4%	83,0%	78,1%	89,8%	82,2%	79,2%	91,3%

							Genauigkeit				F1-Scores				Mean F1 pro Klasse				
Modell	n_ estimators	random_ state	n_jobs	class_ weight	Attribute	Splits	Ø	Std	Max	Min	Ø	Std	Max	Min	AGR	ANC	COS	IND	RES
XGBClassifier	500	42	1	balanced	alle	5	86,9%	2,2%	90,3%	84,6%	83,8%	4,8%	88,9%	75,2%	71,4%	91,7%	82,5%	83,4%	90,1%
XGBClassifier	250	42	1	balanced	alle	5	86,6%	2,2%	90,3%	84,6%	83,5%	4,7%	89,0%	75,2%	71,4%	91,4%	82,2%	82,5%	90,1%
XGBClassifier	200	42	1	balanced	alle	5	86,6%	2,4%	90,3%	83,7%	84,0%	4,7%	89,0%	75,2%	74,1%	91,0%	82,3%	82,5%	90,1%
XGBClassifier	150	42	1	balanced	alle	5	86,2%	2,6%	91,1%	83,7%	83,7%	4,8%	89,9%	75,1%	74,1%	91,0%	81,5%	81,5%	90,3%
XGBClassifier	100	42	1	balanced	alle	5	85,9%	2,4%	90,3%	83,7%	83,5%	4,9%	89,3%	74,5%	74,1%	90,4%	80,9%	82,1%	90,0%
XGBClassifier	50	42	1	balanced	alle	5	86,4%	2,8%	91,1%	83,7%	84,0%	5,2%	90,0%	74,5%	74,1%	90,7%	81,6%	83,5%	90,0%
LGBMClassifier	500	42	1	balanced	alle	5	84,3%	1,3%	86,3%	83,1%	81,5%	2,5%	85,7%	79,0%	69,1%	91,1%	77,5%	82,0%	87,8%
LGBMClassifier	250	42	1	balanced	alle	5	84,3%	1,3%	86,3%	83,1%	81,5%	2,5%	85,7%	79,0%	69,1%	91,1%	77,5%	82,0%	87,8%
LGBMClassifier	200	42	1	balanced	alle	5	84,3%	1,3%	86,3%	83,1%	81,5%	2,5%	85,6%	79,0%	69,1%	91,0%	77,8%	81,6%	87,8%
LGBMClassifier	150	42	1	balanced	alle	5	85,1%	1,4%	87,1%	83,1%	82,2%	2,3%	86,3%	79,8%	69,1%	91,0%	79,6%	83,3%	88,1%
LGBMClassifier	100	42	1	balanced	alle	5	84,8%	1,1%	86,3%	83,1%	81,9%	2,7%	85,7%	79,6%	69,1%	91,6%	78,3%	82,2%	88,2%
LGBMClassifier	50	42	1	balanced	alle	5	85,0%	1,3%	87,1%	83,1%	82,6%	2,4%	84,9%	78,3%	72,7%	90,0%	79,8%	82,2%	88,4%
	max_ iter																		
MLPClassifier	500	42					30,7%	2,8%	35,5%	27,6%	16,2%	3,9%	20,2%	10,7%	0,0%	18,6%	38,9%	18,7%	5,2%
MLPClassifier	5000	42					30,7%	2,8%	35,5%	27,6%	16,2%	3,9%	20,2%	10,7%	0,0%	18,6%	38,9%	18,7%	5,2%

Es wurden fünf verschiedene ML-Modelle getestet. Außerdem wurde zum Vergleich ein Multi-Layer-Perceptron (MLP) getestet. Dabei handelt es sich um ein neuronales Netzwerk. Dieser performte, aber wesentlich schlechter als die fünf ML-Modelle. Die Zahl an verschiedenen ML-Modellen ist sehr groß, weshalb es schwerfällt sich auf fünf bestimmte festzulegen. In dieser Arbeit wurde GPT-4.0 verwendet, um eine Shortlist von 5 verschiedenen ML-Modellen zu erstellen, die vor allem im Ensemble-Learning zum Einsatz kommen (Abbildung 45). Der Vorteil beim Ensemble-Learning ist, dass mehrere Entscheidungsbäume miteinander kombiniert werden und so ein robusteres Modell entsteht. Folgende Modelle wurden von GPT-4.0 vorgeschlagen und in dieser Arbeit untersucht:

- Random Forest Classifier
- Extra Trees Classifier
- Bagging Classifier
- XG-Boost Classifier
- LightGBM Classifier



Abbildung 45: KI-Prompt

RandomForest wurde 1995 von Ho konzeptualisiert und wird seitdem laufend verbessert. RandomForest besteht aus mehreren Entscheidungsbäumen, die sowohl kleine als auch große Datensätze überwacht klassifizieren. RandomForest ist eines der bekanntesten und populärsten ML-Modelle in verschiedenen Anwendungsbereichen (Finanzwesen, Geoinformatik, Bioinformatik etc.). ExtraTrees ist die Kurzform für „Extremely Randomized Trees“ und baut wie RandomForest mehrere Entscheidungsbäume auf. Allerdings betrachtet ExtraTrees zusätzliche zufällige Untergruppen von Merkmalen und generiert zufällige Schwellenwerte für die Aufteilung der Objekte. Dieses hohe Maß an Zufälligkeit trägt zu einer starken Generalisierung bei. (Hussein und R. M. Zeebaree 2024)

Bagging ist die Kurzform für „Bootstrap Aggregating“ und trainiert mehrere Modelle, meist Entscheidungsbäume, um Datensätze zu klassifizieren. Jedes der trainierten Modelle erhält dasselbe Gewicht und am Ende wird die Vorhersage durch Mehrheitsvoting bestimmt. Bagging, ExtraTrees und RandomForest sind sich in der Regel sehr ähnlich, da sie alle Entscheidungsbäume verwenden. Beim Bagging wird jedoch – anders als beim RandomForest oder bei ExtraTrees – keine zusätzliche Zufälligkeit bei der Auswahl der Merkmale hinzugefügt. (Yaman und Subasi 2019)

XGBoost wurde 2016 von Chen und Guestrin präsentiert und ist eine signifikante Verbesserung von davor bestehenden Gradient-Boosting-Techniken. Beim Gradient Boosting versucht jedes neue Modell, die Fehler der Vorgängermodelle auszubessern. Der Begriff Gradient beschreibt die Richtung des stärksten Fehlers, der zuerst ausgebessert werden soll. XGBoost zeichnet sich durch hohe Effizienz und Flexibilität aus und vermeidet Overfitting. XGBoost hat bereits in verschiedenen Bereichen gezeigt, dass es ein Meilenstein in der Entwicklung von Gradient-Boosting-Modellen ist. (Imani et al. 2025)

Wie XGBoost gehört auch LightGBM zu den Gradient-Boosting-Methoden. Es verwendet zwei zentrale Techniken: Zum einen GOSS (Gradient-based One-Side Sampling), das dafür sorgt, dass nur die wichtigsten Datenpunkte mit großem Gradienten verwendet werden und die Rechenzeit verkürzt. Zum anderen EFB (Exclusive Feature Bundling), das Merkmale zusammenfasst, um die Anzahl der Features zu minimieren. Der Vorteil gegenüber XGBoost liegt in der höheren Geschwindigkeit bei der Auswertung. (Ke et al. 2017)

Somit wurden drei RandomTree-Techniken mit Unterschieden in der Zufälligkeit und zwei Gradient-Boosting-Methoden untersucht und verglichen.

Je nach Modell gibt es unterschiedliche Parameter. In dieser Arbeit wurde der Fokus auf einen Parameter gelegt: „n_estimators“. Bei RandomForest, ExtraTrees und Bagging beschreibt dieser Parameter die Anzahl an Bäumen, die gebaut werden. Bei XGBoost und LGBM gibt „n_estimators“ die Anzahl an aufeinanderfolgenden Bäumen an, die sich gegenseitig korrigieren. Dieser Parameter wurde deshalb ausgewählt, da hier keine Einstellung von Anfang an als logisch erschien und unterschiedliche Werte zum Teil deutlichen Einfluss auf das Ergebnis haben. Andere Parametereinstellungen waren entweder von Anfang an klar oder hatten auf den ersten Blick wenig Auswirkung auf die Genauigkeit. Zu ersteren zählt beispielsweise die Einstellung „random_state=42“, die für reproduzierbare Ergebnisse sorgt, oder die Einstellung „class_weight=balanced“, die dafür sorgt, dass unterrepräsentierte Klassen stärker gewichtet werden. Änderungen bei Parametern wie „max_depth“ oder „learning_rate“ hatten bei einer ersten oberflächlichen Analyse kaum Auswirkungen auf die Genauigkeit. Ein weiterer Grund, warum nicht alle Parametereinstellungen auf ihre Auswirkung untersucht wurden, ist die zeitliche Kapazität. Alle fünf Modelle mit allen Parameter-Kombinationen zu analysieren, würde den Rahmen dieser Masterarbeit weit überschreiten.

Folgende Einstellungen für den Parameter „n_estimators“ wurden für jedes Modell getestet: 50, 100, 150, 200, 250, 500. 200 gilt in vielen Modellen als „Default-Value“. Für das Modell XGBoost wurde zusätzlich „n_estimators = 1000“ getestet, da für dieses Modell „n_estimators=500“ die höchste Genauigkeit aufwies und mit der Erhöhung des Wertes auf 1000 überprüft wurde, ob eine weitere Verbesserung möglich ist.

Die Ergebnisse zeigen, dass “n_estimators” keine großen Auswirkungen auf die Genauigkeit der Modelle hat. Die Unterschiede zwischen der niedrigsten und höchsten Genauigkeit betragen für die fünf Modelle:

- RandomForest: 1,5 %
- ExtraTrees: 0,6 %
- Bagging: 0,8 %
- XGBoost: 1,0 %
- LightGBM: 0,8 %

Die Ergebnis-Tabelle ist in drei Teile gegliedert. Die Statistiken im ersten Abschnitt betreffen die generelle Genauigkeit der Modelle. Der zweite Teil behandelt den gesamten F1-Score der Modelle und der dritte die F1-Scores der einzelnen Klassen. Für die Spalte “Mean F1” und die F1-Scores der Klassen wurden die besten Werte je Modell in Hellgrün und die besten Werte modellübergreifend in Dunkelgrün eingefärbt.

Der F1-Score ist eine sehr bekannte Methode, um die Genauigkeit von Klassifizierungsmodellen zu bestimmen. Im ersten Schritt wird für jede Klasse ein F1-Score berechnet, der angibt, welcher Anteil der Features einer Klasse richtig klassifiziert wird. Der insgesamt F1-Score ist der Durchschnitt der F1-Scores je Klasse. In dieser Auswertung ist der F1-Score stets niedriger als die Genauigkeit, da die Klasse “AGRICULTURE” mit sehr wenigen Features den niedrigsten F1-Score hat. Diese fehlende Genauigkeit in der kleinsten Klasse hat eine größere Auswirkung auf den F1-Score, als auf die allgemeine Genauigkeit.

Insgesamt liefern XGBoost (86,9 %) und Bagging (86,7 %) die besten Ergebnisse. ExtraTrees (78,6 %) fällt im Vergleich zu den anderen vier Modellen deutlich ab. Die stabilsten Ergebnisse liefert LightGBM (Standardabweichung: 1,1 %). XGBoost ist das einzige Modell, das in vereinzelt Durchläufen (siehe Max) Genauigkeiten von über 90 % erreicht hat. Im Gegensatz dazu ist ExtraTrees das einzige Modell mit vereinzelt Durchläufen (siehe Min) mit Genauigkeiten unter 80 %. Den besten durchschnittlichen F1-Score weist der BaggingClassifier auf (85,5 %). Diese Methode hat auch den mit Abstand besten F1-Wert in der Kategorie “AGRICULTURE” (82,4 %). Diese Kategorie erkennen andere Modelle (RandomForest, ExtraTrees) deutlich schlechter (jeweils 63,1 %). Nebengebäude (ANCILLARY) werden von allen fünf Modellen gut erkannt (> 87 %). Dasselbe gilt für Wohngebäude (RESIDENTIAL, > 87 %). Diese beiden Kategorien werden von manchen Modellen zu mehr als 90 % richtig erkannt.

Die Kombination an Parametern mit dem besten Ergebnis ist für jedes Modell in Gelb hervorgehoben. Haben zwei unterschiedliche “n_estimators”-Werte dieselbe Gesamtgenauigkeit, wird die Kombination mit dem höheren F1-Score und in weiterer Folge dem niedrigeren “n_estimators”-Wert hervorgehoben. Die Modelle RandomForest, Bagging, XGBoost und LightGBM werden für die weiteren Analyseschritte

verwendet. Das Modell ExtraTrees wird aufgrund der deutlich schlechteren Performance nicht weiter betrachtet. Dasselbe gilt für den MLPClassifier, der bereits in diesem Kapitel kurz behandelt wurde. Für die weiteren Schritte im praktischen Teil werden die jeweils optimalen Kombinationen je ML-Modell verwendet.

8.5.2 Attribute analysieren

Insgesamt wurden 33 Attribute auf ihre Eignung untersucht. Tabelle 8 zeigt alle Attribute. In diesem Absatz werden die Attribute und deren Herkunft zunächst beschrieben, und anschließend wird gezeigt, wie wertvoll sie für das ML waren.

Zentroid

Für jedes Gebäude wurde in Python ein Zentroid berechnet und als X- und Y-Koordinate gespeichert.

Umfang & Fläche

Diese beiden Attribute wurden ebenfalls direkt in Python für jedes Gebäudepolygon berechnet.

Höhe

Dieses Attribut wurde berechnet, indem die GIK mit den digitalen Höhenmodellen der Stadt Linz verschnitten wurde. Aus dem Digitalen Geländemodell (DGM) und dem Digitalen Oberflächenmodell (DOM) wurde ein neutralisiertes Oberflächenmodell (nDOM) erstellt. Dieses Modell wurde dann mit den GIK-Gebäuden verschnitten, um verschiedene Höhen pro Gebäude zu berechnen. Für die Arbeit wurde die maximale Höhe pro Gebäude verwendet.

Gebäudeeigenschaft

Dieses Attribut stammt aus dem AGWR und liegt, wie bereits in Kapitel 8.3.2 erwähnt, pro Adresse und nicht pro Gebäude vor. Somit haben alle Gebäude, die zu einer Adresse gehören, die entsprechende Gebäudeeigenschaft. Die Gebäudeeigenschaft liegt sowohl als Text als auch in numerischer Form vor. Für diese Arbeit wurde die numerische Form verwendet, da sie maschinenlesbar ist. Abbildung 44 zeigt die möglichen Gebäudeeigenschaften laut AGWR.

8.3 Gebäudeeigenschaft gemäß EU-Gebäudeklassifikation

Insgesamt werden folgende Gebäudeeigenschaften unterschieden

- Gebäude mit einer Wohnung
- Gebäude mit 2 oder mehr Wohnungen
- Wohngebäude für Gemeinschaften
- Hotels und ähnliche Gebäude
- Bürogebäude
- Groß- und Einzelhandelsgebäude
- Gebäude des Verkehrs- und Nachrichtenwesens
- Industrie- und Lagergebäude
- Gebäude für Kultur- und Freizeit Zwecke sowie das Bildungs- und Gesundheitswesen
- landwirtschaftliches Nutzgebäude
- freistehende Privatgarage
- Kirchen, sonstige Sakralbauten
- Pseudobaulichkeit
- sonstiges Bauwerk

Abbildung 46: Gebäudeeigenschaften im AGWR

Wohnzweck

Dieses Attribut wird im AGWR geführt und gibt an, ob die Adresse für Wohnzwecke genutzt wird. Alle Gebäude erhalten das Attribut der Adresse, zu der sie gehören. Diese Vorgehensweise hat einen Vor- und einen Nachteil. Der Vorteil ist, dass Gebäudeteile, die keinen Adresspunkt haben, aber zu einem Gebäudeteil mit Adresspunkt gehören und dieselbe Nutzung haben, dieselben Attribute erhalten. Der Nachteil ist, dass auch Gebäudeteile mit einer anderen Nutzung (z. B. Nebengebäude) das Attribut der Adresse erhalten. Das Attribut WOHNZWECK wird mit 1="ja", oder 0="nein" angegeben.

ADRID & ZU_ADRID

Jeder Adresspunkt des AGWR hat eine ID (ADRID). Die Zugehörigkeit der Gebäude zu den Adresspunkten wird durch die Attribute ADRID (wenn der Punkt innerhalb des Gebäudes liegt und ZU_ADRID (wenn kein Adresspunkt innerhalb des Gebäudes liegt) angegeben.

GIKID

GIKID ist die eindeutige ID jedes Gebäudes der Gebäude-Informations-Karte der Stadt Linz. Bei diesem Attribut handelt es sich um einen sogenannten GUID (Globally Unique Identifier). Dieses Attribut wird so generiert, dass kein Feature dieselbe GIKID haben kann.

ADRCN, PLZ, SKZ

Diese Attribute stammen ebenfalls aus dem AGWR und liegen pro Adresse vor. ADRCN steht für Adresscode und ist der Vorgänger der ADRID, denn mehrere Adresspunkte können denselben Adresscode haben. Um Adresspunkte mit demselben Adresscode voneinander unterscheiden zu können, gibt es die ADRID. SKZ steht für Straßenkennzahl und ist für jede Straße in Österreich einzigartig.

ADRHAUPTGB

Dieses Attribut gibt an, ob das Gebäude ein Hauptgebäude ist. Liegt ein Adresspunkt auf dem Gebäude, handelt es sich um ein Hauptgebäude.

GEBUNTERSC & NUTZUNG

Diese beiden Attribute stammen aus dem AGWR und werden nur gelegentlich befüllt. GEBUNTERSC beschreibt, warum sich das Gebäude von anderen Gebäuden mit derselben Adresse unterscheidet. Beispiele sind "Bau 501", "Bibliothek" oder "Büro und Zentrallager". Das Attribut NUTZUNG beschreibt die Verwendung des Gebäudes. Beispiele für dieses Attribut sind "Technisches Gebäude", "Neues Rathaus" oder "Seniorenzentrum".

STRASSE & HAUSNUMMER

Der Straßenname und die Hausnummer der Adresse, zu der das Gebäude gehört.

Bildeigenschaften

Wie bereits in Absatz 8.3.3 erwähnt, wurde für jedes Gebäude der jeweilige Ausschnitt aus dem True Orthofoto ausgeschnitten. Für jeden Ausschnitt, wurden anschließend neun Bildeigenschaften berechnet. Es wurden die drei Bänder (Rot, Grün, Blau) für jedes Bild analysiert. Für jedes Band wurde der Anteil heller Pixel (> 180), der Anteil dunkler Pixel (< 50) und der Mittelwert berechnet.

POI1 bis POI6

Aus mehreren magistratsinternen Quellen wurde eine Liste an POIs erstellt. Diese liegt adressscharf vor. Diese Liste wurde mit den Gebäuden aus der GIK verknüpft. Um ein Gebäude mehreren POI-Kategorien zuordnen zu können, wurden die Attribute POI1 bis POI6 erstellt. Beispiele für Werte in diesen sechs Attributfeldern sind "Polizei", "Museum/Galerie", "Private Krabbelstube" oder "Seniorenzentren". Manche dieser Attribute sind aber nur sehr selten befüllt. Die Menge an Objekten mit einem Eintrag in diesen Spalten nimmt mit der steigenden Zahl hinter dem "POI" ab. Das führt dazu, dass in den Trainingsdaten keine Gebäude einen Wert in den Attributen "POI3", "POI4", "POI5" und "POI6" haben.

Im nächsten Schritt wurde die Bedeutung der einzelnen Attribute für drei unterschiedliche Modelle berechnet. Man spricht von der sogenannten Feature Importance. Diese kann für die Modelle XGBoost, RandomForest und LightGBM mithilfe einer Funktion der jeweiligen Python-Module berechnet werden. Der BaggingClassifier besitzt diese Funktion nicht, weshalb er in Tabelle 8 fehlt. Die Bedeutung der Attribute wurde jeweils mit den Modelleinstellungen berechnet, die die höchste Genauigkeit lieferten. Bei der Feature-Importance wird gezählt, wie oft ein Attribut bei allen Entscheidungsbäumen im Modell Kriterium für einen Split ist. Die Werte werden als Relativwerte dargestellt und ergeben pro Methode in Summe 100 Prozent. Für jeden Classifier werden die Features nach ihrer Bedeutung sortiert. Daraus wird in der Spalte "Rg Mittelw." (Rang Mittelwert) ein Durchschnitt gebildet. Dieser Mittelwert zeigt

modellübergreifend die wichtigsten Attribute. Die in Python berechneten Attribute (X-Koordinate, Y-Koordinate, Umfang, Fläche) zählen zu den wichtigsten Attributen. Ebenfalls von großer Bedeutung sind die Gebäudeeigenschaft und die Angabe, ob die Adresse des Gebäudes einen Wohnzweck erfüllt.

Die Genauigkeit der einzelnen Modelle zeigt, dass bei XGBoost die Genauigkeit mit steigender Anzahl an Features leicht abnimmt. Beim RandomForest-Classifizier variieren die Werte stärker. Die höchste Genauigkeit erzielt das Modell, bei dem nur Attribute mit einer Feature-Importance von mehr als 3 Prozent verwendet wurden. Bei LightGBM bleibt die Performance stabil. Da aber beim insgesamt genauesten Modell die höchste Genauigkeit mit allen Attributen erreicht werden konnte, wurden für den nächsten Schritt alle Attribute berücksichtigt.

Tabelle 8: Bedeutung der einzelnen Attribute in Prozent

	XGBClassifier (500-86,9)					RandomForest (200-85,1)						LightBGM (150-85,1)					Rang				
Total	alle	- 0er	- <0,5	- <1	- <3	alle	- 0er	- <0,5	- <1	- <3	- <10	alle	- 0er	- <1	- <3	- <5	XGB	RF	LBGM	Ø	Rang
centroid_y	5,70	5,70	5,69	3,68	3,33	14,29	15,44	16,70	15,91	21,82	30,43	13,26	13,26	13,41	14,09	17,53	5	1	1	2,3	1
perimeter	8,01	8,01	8,03	6,47	3,82	10,39	10,13	11,25	10,99	15,78	19,47	6,82	6,82	6,82	7,08	8,94	4	2	6	4,0	2
centroid_x	3,92	3,92	3,93	3,06	2,72	8,54	8,63	8,94	9,74	13,79	17,82	10,78	10,78	10,85	11,58	14,17	8	5	2	5,0	3
h	2,39	2,39	2,31	1,88		9,07	8,42	7,78	8,05	12,43	16,06	9,86	9,86	10,05	10,74	14,38	11	3	3	5,7	4
GDAEIGENS	8,84	8,84	8,46	10,81	13,58	6,19	6,30	6,43	6,29	8,73		3,34	3,34	3,74	5,38	6,14	3	6	10	6,3	5
area	1,30	1,30	1,28	1,21		8,82	9,53	9,64	10,52	13,83	16,21	7,17	7,17	7,07	7,52	10,16	12	4	4	6,7	6
WOHNZW	29,22	29,22	31,35	39,12	48,88	3,49	4,03	4,03	4,46	7,07		1,05	1,05	1,30			1	7	20	9,3	7
band1_dark	1,15	1,15	1,08	1,07		2,86	2,64	2,58	2,46			4,22	4,22	4,20	4,85		14	10	8	10,7	8
ZU_ADRID	0,83	0,83	0,82			3,42	3,35	3,19	3,15	6,55		6,99	6,99	7,08	7,37	9,86	21	8	5	11,3	9
ADRID	1,19	1,19	1,28	1,48		2,44	2,38	2,36	2,41			2,97	2,97	3,06	4,12		13	12	14	13,0	10
GIKID	0,64	0,64	0,62			2,90	2,83	2,73	2,77			6,32	6,32	6,22	6,51	10,48	23	9	7	13,0	10
band3_brig	0,87	0,87	0,78			2,57	2,37	2,20	2,18			2,88	2,88	3,02	4,76		18	11	15	14,7	12
band2_brig	0,97	0,97	0,94			2,27	2,37	2,18	2,17			2,38	2,38	2,53			15	14	16	15,0	13
band3_mean	0,84	0,84	0,91			2,01	1,78	1,71	1,74			3,43	3,43	3,66	5,04	8,35	19	18	9	15,3	14
band2_dark	0,70	0,70	0,59			2,30	2,03	1,96	1,92			3,02	3,02	3,18	3,31		22	13	12	15,7	15
ADRCDD	4,97	4,97	4,37	0,70		1,55	1,49	1,50	1,34			1,09	1,09	1,52			7	22	19	16,0	16
ADRHAUPT GB	2,89	2,89	2,66	3,48	7,22	2,03	1,92	2,17	2,07			0,65	0,65				10	17	23	16,7	17
band1_brig	0,88	0,88	0,89			1,83	1,72	1,60	1,59			3,02	3,02	3,1	4,26		17	20	13	16,7	17
PLZ	12,61	12,61	13,67	17,58	4,83	1,38	1,30	1,21	1,20			0,33	0,33				2	23	26	17,0	19
band3_dark	0,39	0,39				2,22	2,24	2,01	1,87			3,18	3,18	3,25	3,40		26	15	11	17,3	20
band1_mean	0,84	0,84	0,68			1,93	1,92	1,66	1,79			2,31	2,31	2,38			19	19	17	18,3	21
GEBUNTER SC	5,19	5,19	4,67	5,36	7,15	0,74	0,74	0,69				0,58	0,58				6	26	24	18,7	22
band2_mean	0,63	0,63	0,67			2,05	1,93	1,84	1,82			2,06	2,06	2,31			24	16	18	19,3	23
STRASSE	0,94	0,94	0,73			1,37	1,02	1,03	1,08			0,76	0,76				16	24	22	20,7	24
POI_1	3,22	3,22	2,86	4,09	8,47	0,36	0,40					0,06	0,06				9	27	27	21,0	25
HAUSNUMM ER	0,54	0,54	0,74			1,18	1,40	1,16	1,06			0,96	1,01	1,19			25	25	21	23,7	26
SKZ	0,35	0,35				1,61	1,64	1,45	1,44			0,52	0,52				27	21	25	24,3	27

NUTZUNG	0,00					0,09						0,00					28	28	28	28,0	28
POI_2	0,00					0,05	0,04					0,00					28	29	28	28,3	29
POI_3	0,00					0,00						0,00					28	30	28	28,7	30
POI_4	0,00					0,00						0,00					28	30	28	28,7	30
POI_5	0,00					0,00						0,00					28	30	28	28,7	30
POI_6	0,00					0,00						0,00					28	30	28	28,7	30
Genauigkeit	86,90	86,90	86,20	86,20	85,40	85,10	84,50	85,40	84,80	86,10	81,70	85,10	85,10	85,10	85,00	85,00					

8.5.3 Modelle kombinieren

Im nächsten Schritt wurden mehrere Modelle miteinander kombiniert, um die Vorhersage weiter zu verbessern. Die Vorhersage mit mehreren Modellen erfolgt nach dem Prinzip des Ensemble Learnings. Dabei wird jedes Feature von mehreren, unabhängig voneinander trainierten Modellen vorhergesagt. Jedes Modell liefert neben der Vorhersage auch einen Wert, der angibt, wie sicher sich das Modell mit der Vorhersage ist. Die Vorhersage mit der höchsten Sicherheit wird für die finale Vorhersage ausgewählt. Alternativ kann auch die Vorhersage, die am öftesten getroffen wurde, ausgewählt werden.

Im obersten Abschnitt der Tabelle (Tabelle 9) wird das beste Modell (XGB) noch einmal angeführt. Im zweiten Abschnitt werden alle möglichen 2er-Kombinationen aus den besten vier Modellen gebildet. Im dritten Abschnitt werden vier 3er-Kombinationen und im letzten Abschnitt eine 4er-Kombination überprüft. Wie in Tabelle 7 werden neben der Genauigkeit auch F1-Scores, Standardabweichungen sowie Minimal- und Maximal-Werte berechnet. Die Kombination aus XGB und Bagging performt gemeinsam mit der 3er-Kombination aus XGB, Bagging und RandomTree am besten. Die verschiedenen Kombinationen zeigen, dass es hier sehr wohl deutliche Unterschiede (bis zu 2,6 %) gibt.

8.5.4 Attribute eliminieren

Für den letzten Schritt der Modellanalyse wurde die 3er-Kombination aus XGB, Bagging und RandomForest ausgewählt. Sie wurde gegenüber der 2er-Kombination bevorzugt, da man sich so noch unabhängiger von einzelnen Modellen macht. Im letzten Schritt wurden erneut die Attribute genauer analysiert. Tabelle X reiht die Attribute nach ihrer Bedeutung. In diesem letzten Schritt der Modellfindung wurden, beginnend mit den unwichtigsten Attributen (POI3 bis POI6), Attribute entfernt und die Genauigkeit und weitere Kennzahlen berechnet.

Tabelle 10 zeigt, dass das beste Ergebnis erzielt werden kann, wenn nur die 20 wichtigsten Attribute berücksichtigt werden. Generell variiert die Genauigkeit im oberen Teil der Tabelle nur geringfügig. Selbst wenn nur die sieben wichtigsten Attribute für das Training verwendet werden, kann eine Genauigkeit von 87,7 % erreicht werden. Dies ist eine wichtige Erkenntnis, da manche Attribute (z. B. Bildeigenschaften) wesentlich aufwendiger zu berechnen sind. Mit den drei wichtigsten Attributen (Umfang, X-Koordinate, Y-Koordinate) kann immerhin noch eine Genauigkeit von 80,9 % erzielt werden.

Tabelle 9: Genauigkeit der verschiedenen Modell-Kombinationen

							Genauigkeit				F1-Scores				Mean F1 pro Klasse				
Modell	n_ estimators	random_ state	n_jobs	class_ weight	Attribute	Splits	Ø	std	Max	Min	Ø	Std	Max	Min	AGR	ANC	COS	IND	RES
XGBClassifier	500	42	1	balanced	alle	5	86,9%	2,2	90,3%	84,6%	83,8%	4,8%	88,9%	75,2%	71,4%	91,7%	82,5%	83,4%	90,1%
XGBClassifier	500	42	1	balanced	alle	5	88,0%	2,2	91,1%	86,2%	86,2%	3,2%	89,8%	80,9%	78,9%	91,5%	84,3%	85,0%	91,3%
BaggingClassifier	150	42	1		alle	5													
XGBClassifier	500	42	1	balanced	alle	5	87,2%	1,8	90,3%	85,4%	84,5%	5,2%	91,2%	75,9%	74,3%	92,2%	83,1%	82,5%	90,4%
RandomForestClassifier	200	42	1	balanced	alle	5													
XGBClassifier	500	42	1	balanced	alle	5	85,6%	1,3	87,1%	83,7%	82,5%	3,5%	87,1%	78,8%	69,1%	91,9%	79,7%	83,7%	88,9%
LGBMClassifier	150	42	1	balanced	alle	5													
BaggingClassifier	150	42	1		alle	5	87,9%	1,5	89,5%	86,2%	85,9%	2,5%	88,1%	81,1%	78,9%	90,7%	85,1%	83,5%	91,3%
RandomForestClassifier	200	42	1	balanced	alle	5													
BaggingClassifier	150	42	1		alle	5	86,7%	1,1	88,7%	85,5%	84,4%	1,5%	86,1%	81,8%	75,1%	91,4%	81,9%	82,8%	90,9%
LGBMClassifier	150	42	1	balanced	alle	5													
RandomForestClassifier	200	42	1	balanced	alle	5	85,4%	1,3	87,1%	83,9%	82,5%	2,6%	87,1%	79,8%	69,1%	91,4%	79,7%	83,8%	88,4%
LGBMClassifier	150	42	1	balanced	alle	5													

							Genauigkeit				F1-Scores				Mean F1 pro Klasse				
Modell	n_estimators	random_state	n_jobs	class_weight	Attribute	Splits	Ø	Std	Max	Min	Ø	Std	Max	Min	AGR	ANC	COS	IND	RES
XGBClassifier	500	42	1	balanced	alle	5	88,0%	1,9	90,3%	86,2%	86,2%	3,0%	89,0%	80,9%	78,9%	91,3%	84,5%	84,8%	91,3%
BaggingClassifier	150	42	1		alle	5													
RandomForestClassifier	200	42	1	balanced	alle	5													
XGBClassifier	500	42	1	balanced	alle	5	87,1%	1,2	88,7%	85,5%	84,7%	2,4%	87,8%	81,0%	75,1%	92,0%	81,7%	84,2%	90,7%
BaggingClassifier	150	42	1		alle	5													
LGBMClassifier	150	42	1	balanced	alle	5													
XGBClassifier	500	42	1	balanced	alle	5	85,8%	1,5	87,9%	83,7%	82,7%	3,7%	87,1%	78,8%	69,1%	92,3%	79,7%	83,7%	88,6%
RandomForestClassifier	200	42	1	balanced	alle	5													
LGBMClassifier	150	42	1	balanced	alle	5													
BaggingClassifier	150	42	1		alle	5	87,1%	1,4	89,5%	85,5%	84,7%	1,7%	86,9%	81,8%	75,1%	91,7%	82,1%	83,9%	90,9%
RandomForestClassifier	200	42	1	balanced	alle	5													
LGBMClassifier	150	42	1	balanced	alle	5													
XGBClassifier	500	42	1	balanced	alle	5	87,1%	1,2	88,7%	85,5%	84,7%	2,4%	87,8%	81,0%	75,1%	92,0%	81,7%	84,2%	90,8%
BaggingClassifier	150	42	1		alle	5													
RandomForestClassifier	200	42	1	balanced	alle	5													
LGBMClassifier	150	42	1	balanced	alle	5													

Tabelle 10: Genauigkeit des Modells ohne bestimmte Attribute

Modell	n_ estimators	random_ state	n_jobs	class_ weight	Attribute	bis inkl Rg	Genauigkeit				F1-Scores				Mean F1 pro Klasse					
							Ø	Std.	Max	Min	Ø	Std	Max	Min	AGR	ANC	COS	IND	RES	
XGB	500	42	1	balanced	alle	30														
Bagging	150	42	1		alle															
RandomForest	200	42	1	balanced	alle		88,0	1,9	90,3	86,2	86,2	3,0	89,0	80,9	78,9	91,3	84,5	84,8	91,3	
same combination						29	87,7	1,8	90,3	86,2	85,8	2,8	88,9	80,9	78,9	91,0	84,3	83,7	91,3	- POI 3-6
same combination						28	87,9	2,0	90,3	86,2	86,0	3,0	89,0	80,9	78,9	91,0	84,5	84,3	91,3	- POI 2
same combination						27	88,0	1,9	90,3	86,2	86,2	3,0	89,0	80,9	78,9	91,3	84,5	84,8	91,3	- NUTZUNG
same combination						26	87,9	2,1	90,3	85,4	86,1	3,2	89,0	80,3	78,9	91,9	83,8	85,0	90,8	- SKZ
same combination						25	87,5	2,0	90,3	85,4	85,7	3,1	89,0	80,3	78,9	91,6	83,5	83,9	90,8	- HAUSNUMMER
same combination						24	87,5	2,0	90,3	85,4	85,7	3,1	89,0	80,3	78,9	91,9	83,2	83,9	90,8	- POI_1
same combination						23	88,2	2,1	91,1	86,2	86,3	3,1	89,6	80,9	78,9	92,2	84,4	85,1	91,0	- STRASSE
same combination						22	87,9	1,7	90,3	86,2	86,0	2,8	88,9	81,1	78,9	91,0	84,0	85,0	91,4	- band2_mean
same combination						21	87,7	2,0	91,1	85,4	85,9	3,2	89,8	80,3	78,9	91,3	83,7	84,4	91,0	- GEBUNTERSC
same combination						20	88,7	2,7	92,7	86,2	86,8	3,5	90,9	81,1	78,9	91,9	85,2	86,7	91,1	- band1_mean
same combination						19	88,2	1,9	91,1	86,2	86,4	3,0	89,4	81,1	78,9	91,0	84,5	86,7	91,0	- band3_dark
same combination						18	87,7	1,6	90,3	85,5	85,9	2,2	88,9	82,5	78,9	91,0	84,2	84,6	90,8	- PLZ
same combination						17	88,0	1,6	90,3	85,5	86,1	2,3	88,9	82,5	78,9	91,0	84,4	84,8	91,0	- band1_brig
same combination						16	87,2	0,9	87,9	87,5	85,4	1,7	87,0	82,5	78,9	91,0	83,1	83,0	91,1	- ADRHAUPTGB
same combination						15	87,5	1,0	88,7	86,3	85,7	2,2	87,9	81,7	78,9	91,0	83,6	83,8	91,4	- ADRCDD
same combination						14	87,2	1,1	88,7	85,5	85,4	2,1	87,6	81,7	78,9	91,0	82,9	82,8	91,4	- band2_dark
same combination						13	87,1	0,8	87,9	86,2	85,3	2,1	86,7	81,1	78,9	90,7	83,2	83,0	90,7	- band3_mean
same combination						12	87,1	0,5	87,9	86,3	85,7	2,4	89,1	81,7	81,7	90,7	82,9	81,4	91,6	- band2_brig
same combination						11	86,9	1,9	89,5	84,6	85,7	2,7	88,1	80,5	81,7	89,9	83,2	83,0	90,4	- band3_brig
same combination						10	87,0	1,7	88,7	84,6	86,3	3,2	90,0	80,4	84,6	89,7	83,4	82,5	91,2	- ADRID
same combination						9	87,4	1,5	89,5	85,4	85,8	3,0	88,0	80,0	80,3	89,3	84,9	83,5	91,2	- GIKID
same combination						8	87,5	1,9	90,2	85,4	85,7	1,2	87,2	84,3	78,9	89,3	85,2	83,4	91,5	- ZU_ADRID

same combination	7	87,7	1,3	88,7	85,4	84,9	2,3	87,5	80,6	73,3	89,5	85,9	85,2	90,6	- band1_dark
same combination	6	86,4	1,7	88,6	83,7	83,9	1,9	85,7	80,1	74,3	89,3	83,8	81,4	90,7	- WOHNZW
same combination	5	85,0	2,0	87,0	82,3	82,1	3,2	85,0	76,1	71,6	88,4	81,1	79,3	89,9	- area
same combination	4	79,9	3,3	83,7	75,6	75,2	9,5	83,4	67,7	52,9	86,6	76,9	83,1	76,7	- GDAEIGENS
same combination	3	80,9	0,8	82,3	79,8	76,7	3,2	82,7	72,9	55,2	87,8	77,1	86,6	77,0	- h
same combination	2	67,3	4,5	71,0	58,5	61,7	6,6	69,1	52,5	43,9	86,6	60,1	50,3	67,8	- centroid_x
same combination	1	45,5	5,3	52,0	38,7	45,4	6,2	57,0	39,7	52,0	42,7	52,2	34,1	45,9	- perimeter

Nach diesen vier Schritten zur Modellfindung stellte sich die folgende Modellkombination mit den folgenden Attributen als die geeignetste heraus.

Tabelle 11: Finale Modellkombination

Modell	n_estimators	random_state	n_jobs	class_weight
XGBClassifier	500	42	1	balanced
BaggingClassifier	150	42	1	
RandomForestClassifier	200	42	1	balanced

Tabelle 12: Finale Attributauswahl

Attribute				
Y-Koordinate	Umfang	X-Koordinate	Höhe	Gebäudeeigenschaft
Fläche	Wohnzweck	Band1_dark	ZU_ADRID	ADRID
GIKID	Band3_bright	Band2_bright	Band3_mean	Band2_dark
ADRCID	ADRHAUPTGB	Band1_bright	PLZ	Band3_dark

8.6 Bewertung der Ergebnisse

8.6.1 Confusion-Matrix

Um die Fehler des Modells besser analysieren zu können, wurde eine Confusion-Matrix erstellt (Tabelle 11). Diese liefert etwas andere Genauigkeitswerte als die F1-Scores pro Klasse, da bei der Confusion-Matrix kein Durchschnitt aus allen Folds der Cross-Validation berechnet wird, sondern die Ergebnisse der Folds gemeinsam betrachtet werden. Die Confusion-Matrix zeigt, dass landwirtschaftliche Gebäude am schlechtesten vom Algorithmus erkannt werden (72 %). Häufig werden sie mit Wohngebäuden (17 %) oder Handels-/Dienstleistungsgebäuden (11 %) verwechselt. Nebengebäude werden am besten erkannt (96 %). Eine weitere häufige Verwechslung besteht bei Industriegebäuden, die als Handels- und Dienstleistungsgebäude missinterpretiert werden. Im Gegensatz zu den relativen Werten (Tabelle 11), zeigt Tabelle 12 die absoluten Werte. Hier sieht man die geringe Anzahl an landwirtschaftlichen Gebäuden, die dazu führt, dass bereits bei wenigen, falsch klassifizierten Gebäuden die Genauigkeit stark abnimmt. Die einzige Fehlinterpretation, die öfter als zehnmal vorkommt, sind Industriegebäude, die als Handels- und Dienstleistungsgebäude erkannt werden.

Tabelle 13: Confusion Matrix

		PREDICTED CLASS				
		AGRICULTURE	ANCILLARY	COMMERCEandSERVICES	INDUSTRY	RESIDENTIAL
TRUE CLASS	AGRICULTURE	0.72	0.00	0.11	0.00	0.17
	ANCILLARY	0.00	0.96	0.04	0.00	0.01
	COMMERCEandSERVICES	0.00	0.05	0.86	0.05	0.04
	INDUSTRY	0.00	0.03	0.13	0.83	0.01
	RESIDENTIAL	0.01	0.05	0.04	0.00	0.90

Tabelle 14: Confusion Matrix (absolute Werte)

		PREDICTED CLASS				
		AGRICULTURE	ANCILLARY	COMMERCEandSERVICES	INDUSTRY	RESIDENTIAL
TRUE CLASS	AGRICULTURE	13	0	2	0	3
	ANCILLARY	0	153	6	0	1
	COMMERCEandSERVICES	0	8	149	9	7
	INDUSTRY	0	3	13	85	1
	RESIDENTIAL	2	9	6	0	148

8.6.2 Ergebnis für ganz Linz

Nachdem das Modell zur Vorhersage bestimmt wurde, wurden sämtliche Gebäude in Linz mithilfe des ML-Modells automatisch einer der fünf INSPIRE-Gebäudekategorien zugewiesen. Der Großteil der Rechenzeit bei diesem Schritt entfällt auf das Zuschneiden des Orthofotos auf die einzelnen Gebäude. Allerdings kann, wie man in Kapitel 8.5.4 sieht, auch ohne Bildeigenschaften ein gutes Ergebnis erzielt werden. Die Genauigkeit nimmt dann zwar etwas ab, aber im Gegenzug sinkt die Rechenzeit signifikant.

Die finalen Ergebnisse für ganz Linz können nicht mehr weiter validiert werden, da kein weiterer Validierungsdatensatz vorliegt. Deshalb wurde das Ergebnis für Linz stichprobenartig untersucht, um Muster in den Fehlern zu erkennen. Zwei Arten von falschen Klassifikationen stachen bei dieser Überprüfung heraus. Zum einen traten verhältnismäßig viele Fehler am Rand von Gebieten mit einer dominanten Klasse auf. In Linz trifft das vor allem auf die Umgebung des Industriegebiets im Osten der Stadt zu. In solchen Gebieten werden beispielsweise Supermärkte, die im restlichen Stadtgebiet meist korrekt als COMMERCEandSERVICE klassifiziert werden, als INDUSTRY kategorisiert. Ein anderes Beispiel für diese Fehlerart findet man im Norden des Stadtgebiets. Hier liegt der Großteil der selten vertretenen landwirtschaftlichen Gebäude. In der Nähe von Bereichen mit vielen landwirtschaftlichen Gebäuden werden manche Gebäude fälschlicherweise als AGRICULTURE klassifiziert. Die zweite auffällige Fehlerart findet man bei Gebäuden, die aus sehr vielen Gebäudeteilen bestehen. Bei solchen Gebäuden werden oft einzelne Teile falsch klassifiziert. Ein Beispiel ist das Neue Rathaus, das aus sehr vielen Teilen besteht, von denen der Großteil richtig als COMMERCEandSERVICE klassifiziert wird. Zwei Teile werden jedoch falsch zugewiesen, obwohl sie in einem Fall sogar komplett von anderen Gebäudeteilen desselben Gebäudes umschlossen sind.

Das Ergebnis für ganz Linz bietet vor allem einen guten Überblick über die Verteilung der Gebäudearten im Stadtgebiet. In der Karte (Abbildung 45) kann man sehr gut Industrie- und Wohngebiete erkennen. Auch Bereiche mit gemischter Nutzung können so schnell identifiziert werden.

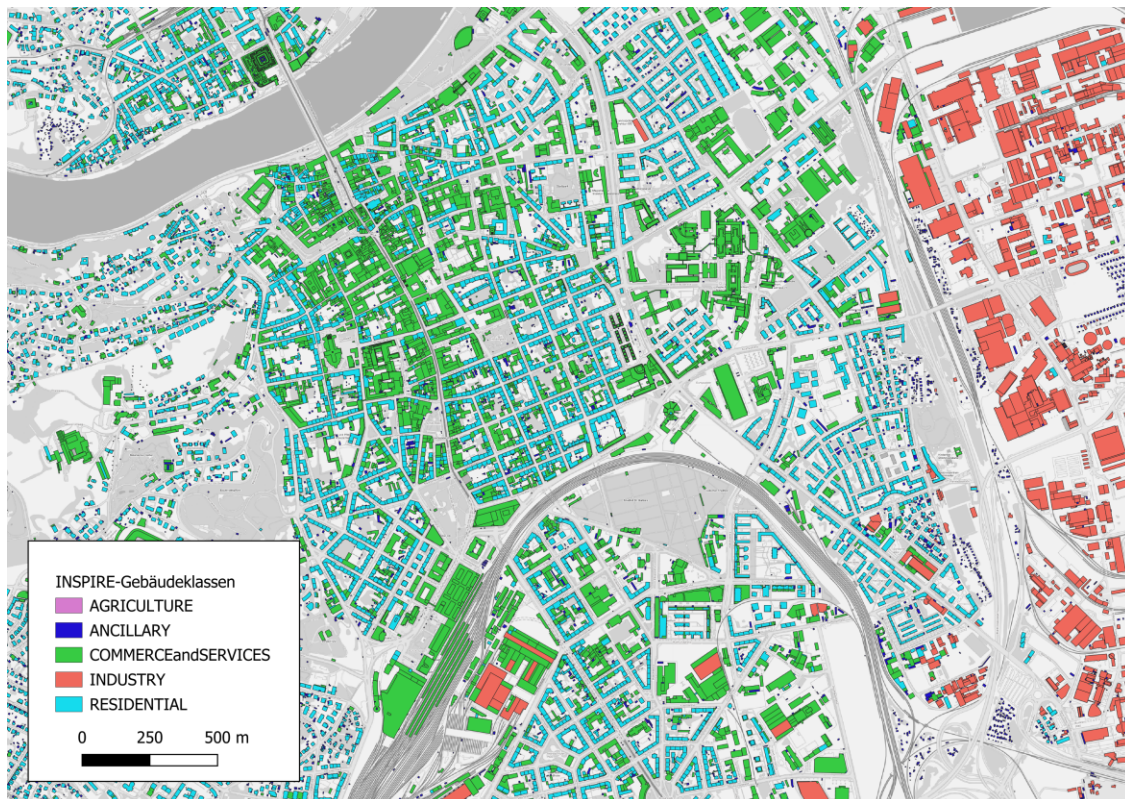


Abbildung 47: Gebäudeklassen für einen Ausschnitt der Stadt Linz



Abbildung 48: Gebäudeklassen für einen Ausschnitt der Stadt Linz (größerer Maßstab)

8.7 Umwandlung in CityGML

Das Training und die Anwendung des ML-Algorithmus fanden im Shapefile-Format statt. Um das Ergebnis in einen UDZ einbetten und dreidimensional visualisieren zu können, wurden die finalen Daten in das Format CityGML umgewandelt. Die größte Herausforderung hierbei liegt im Schema-Matching. Die Art und Weise, wie die Attribute heißen und gespeichert werden, unterscheidet sich signifikant zwischen Shapefile und CityGML. Tabelle 13 zeigt, wie die wichtigsten Attribute des Shapefiles in das CityGML-Format überführt werden können. Manche können in bestehende CityGML-Attribute umgewandelt werden. Für andere müssen neue Attribute erstellt werden. Abbildung 46 zeigt den XML-Code einer Adresse im CityGML-Format. Dies ist nur ein Beispiel und wurde nicht für alle Adressen im Zuge dieser Arbeit gemacht. Um alle Informationen aus dem AGWR in das CityGML-Modell zu überführen wäre dieser Schritt mit sämtlichen Attributen aus dem AGWR notwendig. Abbildung 47 zeigt den XML-Code eines Gebäudes. Auch hier können noch weitere Attribute ergänzt werden. Für die Umsetzung dieser Arbeit wurden nur die wesentlichsten Attribute für jedes Gebäude als CityGML-Attribut gespeichert. Die Informationen der Adressen werden über eine Verlinkung mit den einzelnen Gebäuden verknüpft. Dies geschieht durch "bldg:address xlink:href="ADRID". Nach dieser Umwandlung kann das Ergebnis in verschiedenen 3D-Viewern dargestellt werden (Abbildung 48). Die Ergebnisse dieser Arbeit wurden in Plexmap visualisiert (GEOPLEX 2026). Abbildung 49 zeigt die Attribute des ausgewählten Gebäudes (hellgrün). Hier könnten noch wesentlich mehr Attribute mitgegeben werden. Der Fokus wurde auf die, für die Analyse wichtigsten, Attribute gelegt. Die Einfärbung der Gebäude erfolgt, wie auch in der 2D-Darstellung nach der INSPIRE-Gebäudeklasse. Tabelle 14 zeigt die Farben der jeweiligen Gebäudekategorien. Diese Farben gelten sowohl für die 2D-, als auch die 3D-Darstellung.

Tabelle 15: Attributnamen in CityGML

Attributname im Shapefile	Attributname in CityGML
GIKID	building:building gml:id
h	buliding: measuredHeight
INSPIRE	building:class
area	building:floorArea
perimeter	neues Attribut
Centroid_x & centroid_y	neues Attribut
ADRID	address:address gml:id
ADRCDD	address:code
PLZ	address:postalCode
GDAEIGENSC	neues Attribut
ADRHAUPTGB	neues Attribut
ZU_ADRID	gibt es nicht

Tabelle 16: Farben in der 2D- und 3D-Darstellung

INSPIRE-Gebäudeklasse	Farbe
AGRICULTURE	
ANCILLARY	
COMMERCEandSERVICES	
INDUSTRY	
RESIDENTIAL	

```
<core:addressMember>
  <addr:Address gml:id="5261905_001">
    <addr:postalCode>4020</addr:postalCode>
    <addr:code>5261905</addr:code>
    <gen:stringAttribute name="gdaeigenschaft">Bürogebäude</gen:stringAttribute>
    <gen:stringAttribute name="isPrimaryBuilding">true</gen:stringAttribute>
  </addr:Address>
</core:addressMember>
```

Abbildung 49: XML-Code einer Adresse in CityGML

```
<bldg:Building gml:id="7ab02e5d-6e2f-4c96-b83f-b168e42dd6de">
  <bldg:measuredHeight>6.21</bldg:measuredHeight>
  <bldg:class>AGRICULTURE</bldg:class>
  <gen:stringAttribute name="area">102.50</gen:stringAttribute>
  <gen:stringAttribute name="perimeter">52.85</gen:stringAttribute>
  <gen:doubleAttribute name="centroid_x">68536.06</gen:doubleAttribute>
  <gen:doubleAttribute name="centroid_y">355151.1</gen:doubleAttribute>
  <bldg:address xlink:href="5261905_001"/>
</bldg:Building>
```

Abbildung 50: XML-Code eines Gebäudes in CityGML



Abbildung 51: Darstellung der CityGML-Datei in Plexmap

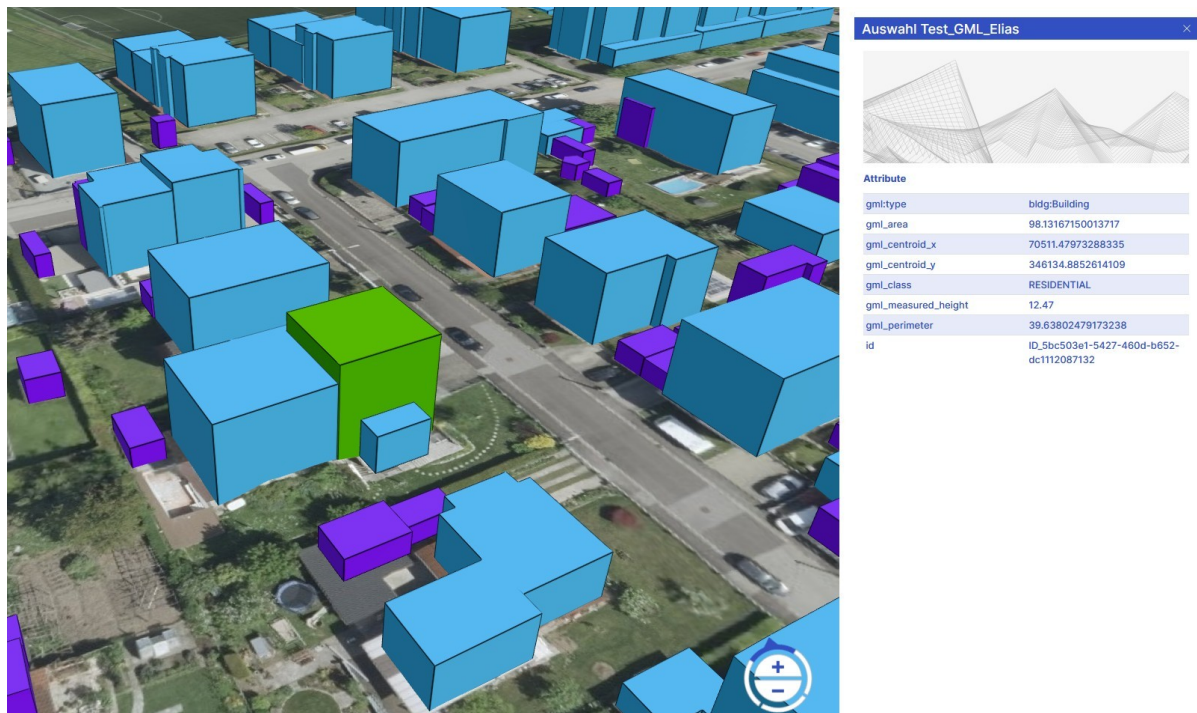


Abbildung 52: Attribute eines Gebäudes im CityGML-Modell

8.8 Fortführung & Umsetzung in anderen Städten

Ein zentraler Aspekt bei der Umsetzung des praktischen Teils dieser Arbeit ist die Übertragbarkeit des Workflows auf andere Städte, sowie die Fortführung einer regelmäßigen Klassifikation. In diesem Kapitel wird daher analysiert, welche Voraussetzungen erfüllt sein müssen, um die vorgestellte Methode erfolgreich anwenden zu können und wie eine Fortführung in Linz gelingen kann. Die Analyse basiert auf einem Gebäudedatensatz. Zwar erhöht eine hohe Qualität dieses Datensatzes die Genauigkeit der Analyse, jedoch ist die Methode grundsätzlich auch auf weniger hochwertige Gebäudedaten anwendbar.

Um die Gebäude mit zusätzlichen Attributen anzureichern, werden weitere Daten benötigt. Einige dieser Attribute lassen sich direkt aus den Geometriedaten der Gebäude ableiten, wie beispielsweise Umfang, Fläche oder Koordinaten. In österreichischen Städten können zahlreiche weitere Attribute durch eine Verknüpfung mit dem Adress-Gebäude-Wohnungsregister (AGWR) generiert werden. Es ist jedoch zu beachten, dass die eindeutige Zuordnung jedes Gebäudes zu einer Adresse, wie sie in Linz vorliegt, eher die Ausnahme darstellt. In der Regel erhalten nur jene Gebäude die Adressattribute, auf deren Fläche ein Adresspunkt liegt. Weitere Attribute, wie etwa die Gebäudehöhe, können auf unterschiedliche Weise ermittelt werden. Hierfür sind in der Regel verschiedene digitale Höhenmodelle erforderlich. Die Liste der POIs kann ebenfalls aus unterschiedlichen Datenquellen erstellt werden. Wie die Analyse des Modells zeigt, hat die POI-Liste jedoch nur einen geringen Einfluss auf die Genauigkeit der Vorhersage. Hinsichtlich der Bildeigenschaften ist die Verwendung eines True-Orthofotos notwendig, dessen Berechnung allerdings einen hohen Rechenaufwand erfordert. Auch diese Attribute sind für das Gesamtergebnis von untergeordneter Bedeutung.

Für eine zufriedenstellende Genauigkeit sind jedoch nur wenige Voraussetzungen notwendig: ein Gebäudedatensatz, manuell klassifizierte Gebäude und idealerweise einige zusätzliche Attribute. Damit ist die Methode grundsätzlich auch mit einem minimalen Datenbestand in anderen Städten umsetzbar.

Um die Klassifikation nicht nur einmalig, sondern laufend umsetzen zu können, sind nicht viele Schritte notwendig. Die händische Klassifizierung muss, sofern keine neuen Attribute hinzukommen, nicht erneut durchgeführt werden. Dasselbe gilt auch für das trainierte ML-Modell. Auch hier muss nur im Falle von anderen Attributen oder Parameter-Einstellungen ein erneutes Training durchgeführt werden. Bleiben die Rahmenbedingungen gleich, kann jederzeit auf Knopfdruck eine automatische Klassifizierung für ganz Linz durchgeführt werden. Dazu muss nur ein aktueller Stand der Gebäude-Informationen-Karte exportiert und dem ML-Modell zur Vorhersage übergeben werden. Die Überführung der Ergebnisse in das CityGML-Format, damit eine Visualisierung in einem dreidimensionalen Viewer stattfinden kann, kann mittels FME erfolgen. Diese Schritte (Export, Klassifikation, CityGML-Erzeugung) können mit Hilfe

eines kleinen Skripts auch automatisch (z. B. täglich) durchgeführt werden. Um die Daten dann automatisch im 3D-Viewer aktualisieren zu können, muss eine Schnittstelle zwischen Speicherort der CityGML-Dateien und 3D-Viewer erstellt werden. Diese ist in Linz aktuell noch nicht vorhanden. Der Import in den Viewer wird derzeit händisch angestoßen.

8.9 Fazit

Diese Arbeit zeigt, dass die KI-gestützte Klassifikation von Gebäuden basierend auf räumlichen Daten und Machine Learning ein vielversprechender Ansatz für die semantische Anreicherung von urbanen Datensätzen ist. Eine hohe Genauigkeit in der automatischen Zuweisung von Gebäudekategorien kann schon mit relativ geringem Aufwand erreicht werden. Die Qualität und Struktur der Eingabedaten haben einen großen Einfluss auf die Performance des Modells. Während hochwertige Gebäudedaten und eine gute Verlinkung zu adressbezogenen Informationen eine präzise Klassifizierung erleichtern, ist die Methode auch auf weniger detaillierte und umfangreiche Datensätze übertragbar. Die Analyse der Attributbedeutungen hat gezeigt, dass wenige, meist sehr einfach zu berechnende Merkmale (Fläche, Umfang, Lage etc.) einen Großteil der Modellleistung erklären. Aufwändig zu generierende Attribute, wie etwa POI-Listen und Bildeigenschaften, erhöhen die Genauigkeit hingegen nur marginal. Sie können im Austausch gegen geringe Abzüge bei der Genauigkeit vernachlässigt werden.

Ein großer zeitlicher Aufwand im Workflow entfällt auf die manuelle Erstellung eines qualitativ hochwertigen Trainingsdatensatzes. Dieser Schritt erfordert ein gutes Verständnis der Gebäudestrukturen und gewissenhaftes Arbeiten. Weitere methodische Hürden stellen die verschiedenen Gebäudenutzungen innerhalb einer Adresse und die eingeschränkte Übertragbarkeit von adressbasierten Informationen auf die Gebäude dar. Außerdem werden seltene Gebäudeklassen (AGRICULTURE) weniger zuverlässig erkannt. Dies spiegelt sich in den F1-Scores der einzelnen Klassen wider. Positiv hervorzuheben sind die Flexibilität und gute Skalierbarkeit des ML-Ansatzes. Die Robustheit der Vorhersage gegenüber Ausreißern konnte durch die Kombination mehrerer Modelle (Ensemble Learning) erhöht werden. Die methodische Trennung von Segmentierung und Klassifizierung ermöglicht zudem eine modulare Anpassung des Workflows an unterschiedliche städtische Kontexte. Die Überführung der Ergebnisse ins CityGML-Format ermöglicht eine vielfältige Visualisierung und erleichtert den Austausch mit anderen Städten. Allerdings verlangt dieser Schritt eine gute Übertragung der Attribute zwischen den Formaten. Um das Ergebnis dieser Arbeit zu optimieren, wird eine Übertragung sämtlicher Attribute ins CityGML-Format vorgeschlagen. Dafür müssen Adressen im CityGML-Datensatz angelegt und mit den Gebäuden verknüpft werden.

Die stichprobenartige Validierung der Ergebnisse für das gesamte Stadtgebiet bestätigt die Praxistauglichkeit der Methode, offenbart aber auch einige Fehlerquellen. Vor allem an den Rändern homogener Nutzungsgebiete und bei komplexen Gebäudestrukturen

treten häufig Fehler auf. Dies sind wertvolle Erkenntnisse, um das Modell in Zukunft gezielt zu verbessern.

Insgesamt belegt der praktische Teil, dass eine KI-gestützte Gebäudeklassifizierung mit überschaubarem Aufwand auf Basis allgemein verfügbarer Geodaten möglich ist und einen substanziellen Beitrag zur Anreicherung von Gebäudedaten leisten kann. Die präsentierte Methode ist grundsätzlich auf andere Städte übertragbar, sofern ein Mindestmaß an Gebäude- und Attributdaten vorliegt. Die Ergebnisse bieten eine solide Grundlage für weitere Anwendungen und Forschungen in diesem Bereich. Somit können UDZ in Zukunft noch effektiver semantisch angereichert werden.

9 Zusammenfassung

Die Analyse hat gezeigt, dass sowohl UDZ als auch GeoAI Themen sind, die sich rasant entwickeln und deren Synthese großes innovatives Potenzial hat. UDZ fungieren als digitale Repräsentationen von urbanen Räumen und bieten eine Plattform, um Simulationen und Analysen durchzuführen sowie Entscheidungshilfen zu schaffen. GeoAI umfasst viele Methoden, um heterogene, umfangreiche Datenmengen automatisiert zu verarbeiten, anzureichern und zu analysieren.

Die systematische Betrachtung entlang der vier zentralen Phasen eines UDZ hat verdeutlicht, dass GeoAI in allen Bereichen einen wesentlichen Beitrag zur Bewältigung bestehender Herausforderungen leisten kann. Insbesondere die Fähigkeit, Daten aus unterschiedlichsten Quellen zu harmonisieren und automatisiert anzureichern, eröffnet neue Möglichkeiten für die Weiterentwicklung von UDZ. GeoAI-Methoden sind zudem in der Lage, komplexe Muster in urbanen Prozessen zu erkennen, die mit klassischen Ansätzen häufig verborgen bleiben. Ihr größter Mehrwert liegt jedoch in der Skalierbarkeit und Automatisierbarkeit, die für die Echtzeitfähigkeit und den nachhaltigen Betrieb von UDZ essenziell sind. Trotz dieser Synergien bleibt der Zusammenschluss von GeoAI und UDZ in der Literatur stark unterrepräsentiert. Diese Arbeit hat diverse Forschungslücken identifiziert. Dazu zählt beispielsweise die Ausarbeitung von standardisierten Datenmodellen und Austauschformaten, die die Interoperabilität einzelner UDZ verbessern würden. Weiters sollten Methoden ausgearbeitet werden, um KI-gestützte Entscheidungen transparent und verständlich erklären zu können. Das ist vor allem im urbanen Kontext notwendig, in dem viele verschiedene Stakeholder den Auswirkungen der Entscheidungen ausgesetzt sind. Von besonderer Bedeutung sind auch soziale und ethische Aspekte, wie etwa Datenschutz, Partizipation und Fairness. Diese Aspekte werden zwar bereits in einigen Publikationen behandelt. Eine tiefergehende Analyse ist dennoch notwendig. Außerdem bleiben viele Ansätze in der konzeptionellen Phase. Projekte in größerem Format könnten den positiven Mehrwert von GeoAI im UDZ-Kontext besser demonstrieren.

Der praktische Teil dieser Arbeit zeigt eine konkrete Form, um mithilfe von GeoAI UDZ zu verbessern. Die Ergebnisse zeigen, dass bereits mit einer geringen Anzahl an Attributen eine gute Klassifikation von Gebäuden erfolgen kann (> 88 %). Die Qualität und Struktur der Eingangsdaten spielen eine große Rolle. Vor allem die Verknüpfung von geometrischen mit attributiven Datensätzen verspricht viele Vorteile (GIK-Gebäude mit AGWR). Die Analyse der Bedeutung der Attribute zeigt, dass wenige, leicht zu berechnende Attribute (Lage, Fläche, Umfang, Höhe) die größte Bedeutung beim Training des Modells haben. Aufwändig zu generierende Attribute verbessern die Genauigkeit hingegen nur marginal oder wirken sich zum Teil sogar negativ auf die Performance des Modells aus. Insgesamt unterstreicht diese Arbeit, dass eine solide Datenbasis und die

gezielte Auswahl von Attributen zum Training des Modells die Grundlage für eine erfolgreiche semantische Anreicherung von 3D-Stadtmodellen sind.

Im Fokus bei der Untersuchung des optimalen KI-Modells standen verschiedene ML-Algorithmen, darunter Random Forest, Bagging, XGBoost und LightGBM, die sich in zahlreichen Studien als leistungsstark für tabellarische Geodaten erwiesen haben. Ergänzend wurde ein Multi-Layer-Perceptron getestet, das allerdings nicht ansatzweise mit den getesteten ML-Modellen mithalten konnte. Die Ergebnisse zeigen, dass XGBoost und Bagging eine hohe Genauigkeit und Robustheit gegenüber Ausreißern aufweisen. Die Kombination mehrerer Modelle im Sinne des Ensemble-Learnings konnte die Genauigkeit der Ergebnisse weiter steigern. Außerdem erhöht dieser Ansatz die Stabilität der Ergebnisse und macht den Workflow robuster gegenüber Schwankungen in den Eingangsdaten. Die genaue Analyse verschiedener Modelle zeigt, dass moderne, baumbasierte ML-Modelle besonders gut geeignet sind, um Gebäude nach ihrer Nutzung zu klassifizieren.

Der ausgearbeitete Workflow ist prinzipiell auf andere Städte übertragbar, wobei ein Gebäudedatensatz und Attributdaten vorausgesetzt sind. Doch die Methode hat auch ihre Grenzen. Seltene Gebäudeklassen, komplexe Gebäudestrukturen und an homogene Gebiete angrenzende Gebäude stellen den Algorithmus vor Herausforderungen. Weiters ist die Überführung der Daten ins CityGML-Format herausfordernd und zeitaufwendig.

Die Ergebnisse dieser Arbeit zeigen, dass die Kombination von GeoAI und UDZ nicht nur konzeptionell vielversprechend, sondern auch praktisch wertvoll für urbane Geoinformation ist. Gleichzeitig verdeutlicht die Arbeit, dass noch viel Forschung und viele Projekte notwendig sind, um Methoden aus der GeoAI optimal mit den Ansprüchen von UDZ zu verknüpfen. Die Erkenntnisse dieser Arbeit bilden eine Basis für künftige Forschungen, um die Synthese der beiden Kernthemen voranzutreiben. Das Aufgreifen der hier ausgearbeiteten Forschungslücken kann einen bedeutenden Mehrwert für UDZ schaffen. Denn nur mit intensiver Forschung in allen Bereichen von UDZ kann das Potenzial von UDZ als Schlüsselinstrument für Smart Cities erreicht werden.

Abkürzungsverzeichnis

Wort	Abkürzung
Association for Computing Machinery	ACM
Auto-Encoders	AE
Architetur, Engineering and Consulting	AEC
AGRICULTURE	AGR
Adress-, Gebäude-, und Wohnungsregister	AGWR
Artificial Intelligence	AI
ANCILLARY	ANC
Bootstrap Aggregating	Bagging
Baunutzungsverordnung	BauNVO
Building-Information-Modell	BIM
Computer-aided design	CAD
City Information Model	CIM
Convolutional neural network	CNN
COMMERCEandSERVICES	COS
Digitales Geländemodell	DGM
Distributed Hash Table	DHT
Deep Learning	DL
Disjunctive Normal Form	DNF
Deep Neural Network	DNN
Digitales Oberflächenmodell	DOM
Differential Privacy	DP
Digitaler Zwilling	DZ
Elektronische Datenverarbeitung	EDV
Edge Machine Learning	eML
Extremely Randomized Trees	ExtraTrees
Fully Connected Layers	FC
Feature Manipulation Engine	FME
European General Data Protection Regulation	GDPR
Geospatial Artificial Intelligence	GeoAI
Gebäude-Informations-Karte	GIK
Geografisches-Informations-System	GIS
Geography Markup Language	GML
Global Navigation Satellite Systems	GNSS
Gradient-based one-side Sampling	GOSS
Gated Recurrent Units	GRU
Globally Unique Identifier	GUID
International Business Machines	IBM
Interface Digital Twin	IDT
Industry Foundation Classes	IFC

INDUSTRY	IND
Internet of Things	IoT
Information Technology	IT
JavaScript Object Notation	JSON
Künstliche Intelligenz	KI
Keyhole Markup Language	KML
Keyhole Markup Zipped	KMZ
Light Gradient Boosting Machine	LGBM
Light Detection and Ranging	LIDAR
Large Language Modells	LLM
Level of Accuracy	LoA
Level of Detail	LoD
Long Range Wide Area Network	LoRaWan
Long-Short-Term-Memory	LSTM
Machine Learning	ML
Multi-Layer Perceptron	MLP
Multi-Order Spatial Interaction Network	MSIN
National Aeronautics and Space Administration	NASA
neutralisiertes digitales Oberflächenmodell	nDOM
Natural Language Processing	NLP
Open Geospatial Consortium	OGC
Point of Interest	POI
Random Sample Consensus	RANSAC
RESIDENTIAL	RES
Random Forest	RF
Recurrent Neural Networks	RNN
Stacked Auto Encoder	SAE
Structured Query Language	SQL
Standardabweichung	Std
Simulation of Urban Mobility	SUMO
Urbane Datenplattform	UDP
Urbaner Digitaler Zwilling	UDZ
Virtuelle Realität	VR
Explainable Artificial Intelligence	XAI
eXtreme Gradient Boosting	XGB
Extensible Markup Language	XML

Tabellenverzeichnis

Tabelle 1: Stärken und Schwächen der verwandten Systeme	20
Tabelle 2: Anwendungsgebiete im Bereich Transport & Mobilität (Jafari et al. 2023).....	50
Tabelle 3: weitere Anwendungsgebiete von UDZ.....	54
Tabelle 4: verwendete Attribute aus dem AGWR.....	89
Tabelle 5: untersuchte Klassifizierungs-Standards	90
Tabelle 6: Aufteilung der händisch gelabelten Gebäude.....	91
Tabelle 7: Performance der verschiedenen Modelle	94
Tabelle 8: Bedeutung der einzelnen Attribute in Prozent.....	103
Tabelle 9: Genauigkeit der verschiedenen Modell-Kombinationen	106
Tabelle 10: Genauigkeit des Modells ohne bestimmte Attribute	108
Tabelle 11: Finale Modellkombination	110
Tabelle 12: Finale Attributauswahl	110
Tabelle 13: Confusion Matrix.....	110
Tabelle 14: Confusion Matrix (absolute Werte)	111
Tabelle 15: Attributnamen in CityGML	113
Tabelle 16: Farben in der 2D- und 3D-Darstellung.....	114

Abbildungsverzeichnis

Abbildung 1: Digital Twin Definitionen (Alvi et al. 2025)	12
Abbildung 2: Digital Twin Definitionen (Barricelli et al. 2019)	13
Abbildung 3: Digital Twin Definitionen (Barricelli et al. 2019)	13
Abbildung 4: Wissenschaftliche Artikel und Google Search Trends (2019-2025) (Alvi et al. 2025)	14
Abbildung 5: Datenfluss im Digitalen Modell und im Digitalen Schatten (Kritzinger et al. 2018)	15
Abbildung 6: Datenfluss im Digitalen Zwilling (Kritzinger et al. 2018)	15
Abbildung 7: Komponenten eines Gebäudezwillings (Khajavi et al. 2019)	17
Abbildung 8: Vor- und Nachteile der verschiedenen BIM-GIS Integrationen (Wang et al. 2019a).....	18
Abbildung 9: Publikationen zum Thema DZ im Zusammenhang mit "Smart Cities" (Wang et al. 2023b)	21
Abbildung 10: Hierarchischer Aufbau und Verbindungen verschiedener DZ (Lu et al. 2020)	22
Abbildung 11: Stufen/Formen von UDZ (Alvi et al. 2025)	24
Abbildung 12: Die 6 Phasen eines UDZ (Petrova-Antonova und Ilieva 2019)	24
Abbildung 13: Bausteine, Potenziale, Herausforderungen und Anwendungsgebiete von UDZ (Shahat et al. 2021)	25
Abbildung 14: Publikationen zu den Themen "Smart City", "IoT" und "Crowdsourcing" (Shahrour und Xie 2021)	26
Abbildung 15: Verarbeitung von Punktwolken mittels Machine-/Deep-Learning (Döllner 2020)	29
Abbildung 16: Speicherung von IoT-Daten mittels Blockchain und DHT (Li et al. 2019)	32
Abbildung 17: Cloud-, On-Premise & Edge-Computing als Teil eines UDZ (Weil et al. 2023)	34
Abbildung 18: 3Vs von Big Data (Rozony et al. 2024)	39
Abbildung 19: 3D-Gebäudemodell der Stadt Linz (www.3dlinz.at)	40
Abbildung 20: 3D-Mesh-Modell der Stadt Linz (www.3dlinz.at)	40
Abbildung 21: LODs für ein Wohngebäude (Biljecki et al. 2016)	41
Abbildung 22: Höhenreferenzen für ein LOD1 Gebäudemodell (Biljecki et al. 2016)	42
Abbildung 23: Ergebnis einer unüberwachten Gebäudesegmentierung (Nys et al. 2020)	43
Abbildung 24: Austauschformate für Vektordaten im UDZ Kontext (Weil et al. 2023) ...	46
Abbildung 25: Streaming-Formate im UDZ Kontext (Weil et al. 2023).....	47
Abbildung 26: Miteinbeziehen von Bürgern in den Planungsprozess mittels VR (Dembski et al. 2020)	48
Abbildung 27: UDZ für Kritische Infrastruktur (Zhu und Jin 2025)	51
Abbildung 28: UDZ für Kritische Infrastruktur (Zhu und Jin 2025)	52

Abbildung 29: Zusammenspiel von GeoAI und GIS (Boutayeb et al. 2024)	60
Abbildung 30: Publikationen zum Thema GeoAI (Lu et al. 2025)	61
Abbildung 31: Entwicklungsphasen von GeoAI (Lu et al. 2025)	62
Abbildung 32: CNN-Architektur (Boutayeb et al. 2024)	64
Abbildung 33: Auto-Encoder-Architektur (Boutayeb et al. 2024)	65
Abbildung 34: Attention-Mechanism	66
Abbildung 35: Einsatzgebiete von GeoAI (Boutayeb et al. 2024)	68
Abbildung 36: Publikationen zum Thema "Edge Computing" (Merenda et al. 2020)	73
Abbildung 37: Herausforderungen bei der Integration von Big Data (Rozony et al. 2024)	74
Abbildung 38: verschiedene Konstruktions-Möglichkeiten für ein LoD1-Gebäude (Lehner und Dorffner 2020)	77
Abbildung 39: Stadtmodell in der Videospielumgebung "Minecraft" (Schrotter und Hürzeler 2020)	80
Abbildung 40: Studien zur semantischen Segmentierung und Klassifizierung von Gebäudetypen (Hertrich 2024)	87
Abbildung 41: Workflow praktischer Teil	88
Abbildung 42: INSPIRE-/SIG3D-Gebäudekategorien	91
Abbildung 43: Händisches Labelling der Trainings- und Validierungsdaten.....	92
Abbildung 44: Generierung der verschiedenen Datensätze	92
Abbildung 45: Gebäudeeigenschaften im AGWR	100
Abbildung 46: Gebäudeklassen für einen Ausschnitt der Stadt Linz	112
Abbildung 47: Gebäudeklassen für einen Ausschnitt der Stadt Linz (kleiner Maßstab).....	112
Abbildung 48: XML-Code einer Adresse in CityGML	114
Abbildung 49: XML-Code eines Gebäudes in CityGML	114
Abbildung 50: Darstellung der CityGML-Datei in Plexmap	115
Abbildung 51: Attribute eines Gebäudes im CityGML-Modell.....	115

Hilfsmittelverzeichnis

KI-Hilfsmittel	Einsatz	Betroffene Teile der Arbeit	Bemerkungen
DeepL	Übersetzung der Kurzfassung	Abstract	
OpenAI ChatGPT-4.1	Erstellung von Code-Snippets für die Klassifizierung	Kap. 8.4.4 Kap. 8.5 Kap. 8.6	Der Python-Code ist im Anhang 1 dargestellt.
OpenAI ChatGPT-4.1	Vorschlag für die Gliederung der einzelnen Kapitel	Kap. 3	Die vorgeschlagene Gliederung wurde mehrfach überarbeitet.
OpenAI ChatGPT-4.1	Vorschlag für KI-Modelle	Kap. 8.5.1	Der Prompt ist im Kapitel 8.5.1 dargestellt.

Literaturverzeichnis

A Novel Digital Twin Framework for Adaptive Urban Weather Visualization Using IoT and Fuzzy Logic (2024). In: *IJIES* 17 (5), S. 494–507.

Aditya, Nova Chandra; Dewiyanti, Dhini; Natalia, Tri Widiyanti; Martana, Salmon Priaji; Abioso, Wanita Subadra (2025): Digital Twin Implementation in Building Lifecycle for Sustainable Urban Development. Department of Architecture Engineering, Universitas Komputer Indonesia.

Agapaki, Evangelia (2022): Airport Digital Twins for Resilient Disaster Management Response. In: Dimitris E. Simos, Varvara A. Rasskazova, Francesco Archetti, Ilias S. Kotsireas und Panos M. Pardalos (Hg.): *Learning and Intelligent Optimization*, Bd. 13621. Cham: Springer International Publishing (Lecture Notes in Computer Science), S. 467–486.

Ahleroff, Shohin; Xu, Xun; Zhong, Ray Y.; Lu, Yuqian (2021): Digital Twin as a Service (DTaaS) in Industry 4.0: An Architecture Reference Model. In: *Advanced Engineering Informatics* 47, S. 101225. DOI: 10.1016/j.aei.2020.101225.

Ahn, Chaewon; Lotfi-Jam, Farzin; Graham, Connor; Bunnell, Tim; Marvin, Simon (2025): Critical urban informatics for urban digital twin models. In: *Nat Cities* 2 (2), S. 114–116. DOI: 10.1038/s44284-024-00171-0.

Alastal, Abdelkhalek I.; Shaqfa, Ashraf Hassan (2022): GeoAI Technologies and Their Application Areas in Urban Planning and Development: Concepts, Opportunities and Challenges in Smart City (Kuwait, Study Case). In: *JDAIP* 10 (02), S. 110–126. DOI: 10.4236/jdaip.2022.102007.

Alem, Abebaw; Kumar, Shailender (2022): Deep Learning Models Performance Evaluations for Remote Sensed Image Classification. In: *IEEE Access* 10, S. 111784–111793. DOI: 10.1109/ACCESS.2022.3215264.

AlexandraMicu, E.; Asgari, Fereshteh; Khouadjia, Mostepha; Connolly, Lorcan; Zavitsas, Kostas (2025): A Probabilistic Digital Twin Model for Inland Waterway Transportation Systems Using Bayesian Networks. In: Eftihia G. Nathanail, Nikolaos Gavanis und Evangelos Adamos (Hg.): *Climate Crisis and Resilient Transportation Systems*. Cham: Springer Nature Switzerland (Lecture Notes in Intelligent Transportation and Infrastructure), S. 855–864.

Ali, Zeeshan; Diakite, Mama; Traore, Mamadou Kaba (2025): Developing Urban Digital Twin For Mobility. Hg. v. IEEE. Annual Modeling and Simulation Conference (ANNSIM).

Alibrandi, Umberto (2022): Risk-Informed Digital Twin of Buildings and Infrastructures for Sustainable and Resilient Urban Communities. In: *ASCE-ASME J. Risk Uncertainty Eng. Syst., Part A: Civ. Eng.* 8 (3), Artikel 04022032. DOI: 10.1061/AJRUA6.0001238.

Aloqaily, Moayad; Otoum, Safa; Ridhawi, Ismaeel Al; Jararweh, Yaser (2019): An intrusion detection system for connected vehicles in smart cities. In: *Ad Hoc Networks* 90, S. 101842. DOI: 10.1016/j.adhoc.2019.02.001.

Alvi, Maira; Dutta, Hrishikesh; Minerva, Roberto; Crespi, Noel; Raza, Syed Mohsan; Herath, Manoj (2025): Global perspectives on digital twin smart cities: Innovations, challenges, and pathways to a sustainable urban future. In: *Sustainable Cities and Society* 126, S. 106356. DOI: 10.1016/j.scs.2025.106356.

Austin, Mark; Delgoshaei, Parastoo; Coelho, Maria; Heidarinejad, Mohammad (2020): Architecting Smart City Digital Twins: Combined Semantic Model and Machine Learning Approach. In: *J. Manage. Eng.* 36 (4), Artikel 04020026. DOI: 10.1061/(ASCE)ME.1943-5479.0000774.

Awan, Faraz Malik; Minerva, Roberto; Crespi, Noel (2021): Using Noise Pollution Data for Traffic Prediction in Smart Cities: Experiments Based on LSTM Recurrent Neural Networks. In: *IEEE Sensors J.* 21 (18), S. 20722–20729. DOI: 10.1109/JSEN.2021.3100324.

Azhar, Salman (2011): Building Information Modeling (BIM): Trends, Benefits, Risks, and Challenges for the AEC Industry. In: *Leadership Manage. Eng.* 11 (3), S. 241–252. DOI: 10.1061/(ASCE)LM.1943-5630.0000127.

Babanagar, Nandeesh; Sheil, Brian; Ninić, Jelena; Zhang, Qianbing; Hardy, Stuart (2025): Digital twins for urban underground space. In: *Tunnelling and Underground Space Technology* 155, S. 106140. DOI: 10.1016/j.tust.2024.106140.

Badawi, Hawazin Faiz; Laamarti, Fedwa; El Saddik, Abdulmotaleb (2021): Devising Digital Twins DNA Paradigm for Modeling ISO-Based City Services. In: *Sensors (Basel, Switzerland)* 21 (4). DOI: 10.3390/s21041047.

Ballouch, Zouhair; Hajji, Rafika; Poux, Florent; Kharroubi, Abderrazzaq; Billen, Roland (2022): A Prior Level Fusion Approach for the Semantic Segmentation of 3D Point Clouds Using Deep Learning. In: *Remote Sensing* 14 (14), S. 3415. DOI: 10.3390/rs14143415.

Barricelli, Barbara Rita; Casiraghi, Elena; Fogli, Daniela (2019): A Survey on Digital Twin: Definitions, Characteristics, Applications, and Design Implications. In: *IEEE Access* 7, S. 167653–167671. DOI: 10.1109/ACCESS.2019.2953499.

Bartos, Matthew; Kerkez, Branko: Pipedream: An interactive digital twin model for natural and urban drainage systems. In: *Environmental Modelling & Software* 2021 (144).

Basha, Syed Muzamil; Taherdoost, Hamed; Zanchettin, Cleber (Hg.) (2024): Innovations in Cybersecurity and Data Science. Proceedings of ICICDS 2024. 1st ed. 2024.

Singapore: Springer Nature Singapore; Imprint Springer (Algorithms for Intelligent Systems).

Batty, Michael (2018): Digital twins. In: *Environment and Planning B: Urban Analytics and City Science* 45 (5), S. 817–820. DOI: 10.1177/2399808318796416.

Batty, Michael (2024): Digital twins in city planning. In: *Nature computational science* 4 (3), S. 192–199. DOI: 10.1038/s43588-024-00606-7.

Biljecki, Filip; Ledoux, Hugo; Stoter, Jantien (2016): An improved LOD specification for 3D building models. In: *Computers, Environment and Urban Systems* 59, S. 25–37. DOI: 10.1016/j.compenvurbsys.2016.04.005.

Biljecki, Filip; Stoter, Jantien; Ledoux, Hugo; Zlatanova, Sisi; Çöltekin, Arzu (2015): Applications of 3D City Models: State of the Art Review. In: *IJGI* 4 (4), S. 2842–2889. DOI: 10.3390/ijgi4042842.

Blucher Design Proceedings (2019 - 2019). 37 Education and Research in Computer Aided Architectural Design in Europe and XXIII Iberoamerican Society of Digital Graphics, Joint Conference (N. 1). Porto, Portugal, 11.09.2019 - 13.09.2019. São Paulo: Editora Blucher.

Boutayeb, Anasse; Lahsen-cherif, Iyad; Khadimi, Ahmed El (2024): A comprehensive GeoAI review: Progress, Challenges and Outlooks.

Braga, David M.; Rawal, Bharat S. (2024): Quantum and AI-Enhanced Digital Twin Systems. In: 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT). 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT). Kamand, India, 24.06.2024 - 28.06.2024: IEEE, S. 1–8.

Braik, Abdullah M.; Koliou, Maria (2023): A novel digital twin framework of electric power infrastructure systems subjected to hurricanes. In: *International Journal of Disaster Risk Reduction* 97, S. 104020. DOI: 10.1016/j.ijdrr.2023.104020.

Brasil, José Artur Teixeira; Macedo, Marina Batalini de; Oliveira, Thalita Raquel Pereira de; Ghiglieno, Filippo Giovanni; Souza, Vladimir Caramori Borges de; Marinho e Silva, Gabriel et al. (2022): Can we scale Digital Twins of Nature-based Solutions for stormwater and transboundary water security projects? In: *Journal of Hydroinformatics* 24 (4), S. 749–764. DOI: 10.2166/hydro.2022.142.

buildingSMART International (Hg.) (2024): Industry Foundation Classes (IFC) - An Introduction. Online verfügbar unter <https://technical.buildingsmart.org/standards/ifc/>.

Bundesministerium für Justiz (Hg.) (2023): Verordnung über die bauliche Nutzung der Grundstücke. Online verfügbar unter <https://www.gesetze-im-internet.de/baunvo/>.

Caldarelli, G.; Arcaute, E.; Barthelemy, M.; Batty, M.; Gershenson, C.; Helbing, D. et al. (2023): The role of complexity for digital twins of cities. In: *Nature computational science* 3 (5), S. 374–381. DOI: 10.1038/s43588-023-00431-4.

Cao, Rui; Qiu, Guoping (2018): Urban Land Use Classification Based on Aerial and Ground Images. In: 2018 International Conference on Content-Based Multimedia Indexing (CBMI). 2018 International Conference on Content-Based Multimedia Indexing (CBMI). La Rochelle, 04.09.2018 - 06.09.2018: IEEE, S. 1–6.

Caporusso, N.; Elleman, N.; Cho, S. S. (2020): Interface Digital Twins: Rendering Physical Devices Accessible to People who are Blind. In: 2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO). 2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO). Opatija, Croatia, 28.09.2020 - 02.10.2020: IEEE, S. 982–986.

Cárdenas-León, Iván; Koeva, Mila; Nourian, Pirouz; Davey, Calayde (2024): Urban digital twin-based solution using geospatial information for solid waste management. In: *Sustainable Cities and Society* 115, S. 105798. DOI: 10.1016/j.scs.2024.105798.

Chen, Changjie; Han, Yu; Galinski, Andrea; Calle, Christian; Carney, Jeffrey; Ye, Xinyue; van Westen, Cees (2025): Integrating Urban Digital Twin with Cloud-Based Geospatial Dashboard for Coastal Resilience Planning: A Case Study in Florida. In: *Journal of Planning Education and Research* 45 (4), S. 743–759. DOI: 10.1177/0739456X251316185.

Chiến Phạm (2014): The Business Value of BIM for Construction in Major Global Markets. How Contractors Around the World Are Driving Innovation With Building Information Modeling. Hg. v. SmartMarket Report. McGraw Hill Construction. Bedford, MA.

Coluccino, Delia Maria; Balzola, Beatrice; Vagnoni, Simone; Ciutat, Lucas Ramon; Ferraris, Andrea Filippo (2026): Invisible Citizens, Visible Futures: Rethinking Inclusivity in Urban Digital Twins. In: José Valente de Oliveira, João Leite, João Rodrigues, João Dias und Pedro Cardoso (Hg.): Progress in Artificial Intelligence, Bd. 16121. Cham: Springer Nature Switzerland (Lecture Notes in Computer Science), S. 492–504.

Cowell, Nicole Hannah; Chapman, Lee; Topping, David; James, Philip; Bell, Daniel; Bannan, Thomas et al. (2025): Moving from monitoring to real-time interventions for air quality: are low-cost sensor networks ready to support urban digital twins? In: *Front. Sustain. Cities* 6, Artikel 1500516. DOI: 10.3389/frsc.2024.1500516.

Damjanovic-Behrendt, Violeta (2018): A Digital Twin-based Privacy Enhancement Mechanism for the Automotive Industry. In: 2018 International Conference on Intelligent Systems (IS). 2018 International Conference on Intelligent Systems (IS). Funchal - Madeira, Portugal, 25.09.2018 - 27.09.2018: IEEE, S. 272–279.

- Dasgupta, Sagar; Rahman, Mizanur; Lidbe, Abhay D.; Jones, Steven (2021): A Transportation Digital-Twin Approach for Adaptive Traffic Control Systems.
- Dembski, Fabian; Wössner, Uwe; Letzgus, Mike (2019 - 2019): The Digital Twin Tackling Urban Challenges with Models, Spatial Analysis and Numerical Simulations in Immersive Virtual Environments. In: Blucher Design Proceedings. 37 Education and Research in Computer Aided Architectural Design in Europe and XXIII Iberoamerican Society of Digital Graphics, Joint Conference (N. 1). Porto, Portugal, 11.09.2019 - 13.09.2019. São Paulo: Editora Blucher, S. 795–804.
- Dembski, Fabian; Wössner, Uwe; Letzgus, Mike; Ruddat, Michael; Yamu, Claudia (2020): Urban Digital Twins for Smart Cities and Citizens: The Case Study of Herrenberg, Germany. In: *Sustainability* 12 (6), S. 2307. DOI: 10.3390/su12062307.
- Dimassi, Mohammad; Samhat, Abed Ellatif; Zaraket, Mohammad; Haidar, Jamal; Shukor, Mustafa; Ghandour, Ali J. (2021): Buildings Classification using Very High Resolution Satellite Imagery.
- DIN Media GmbH (Hg.) (2024): DIN SPEC 91607:2024-11 Digitale Zwillinge für Städte und Kommunen.
- Döllner, Jürgen (2020): Geospatial Artificial Intelligence: Potentials of Machine Learning for 3D Point Clouds and Geospatial Digital Twins. In: *PFG* 88 (1), S. 15–24. DOI: 10.1007/s41064-020-00102-3.
- Dou, S. Q.; Zhang, H. H.; Zhao, Y. Q.; Wang, A. M.; Xiong, Y. T.; Zuo, J. M. (2020): RESEARCH ON CONSTRUCTION OF SPATIO-TEMPORAL DATA VISUALIZATION PLATFORM FOR GIS AND BIM FUSION. In: *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XLII-3/W10, S. 555–563. DOI: 10.5194/isprs-archives-XLII-3-W10-555-2020.
- Droin, Ariane; Wurm, Michael; Sulzer, Wolfgang (2020): Semantic labelling of building types. A comparison of two approaches using Random Forest and Deep Learning. Hg. v. DGPF.
- Dwivedi, Rudresh; Dave, Devam; Naik, Het; Singhal, Smriti; Omer, Rana; Patel, Pankesh et al. (2023): Explainable AI (XAI): Core Ideas, Techniques, and Solutions. In: *ACM Comput. Surv.* 55 (9), S. 1–33. DOI: 10.1145/3561048.
- Eicker, Ursula; Weiler, Verena; Schumacher, Jürgen; Braun, Reiner (2020): On the design of an urban data and modeling platform and its application to urban district analyses. In: *Energy and Buildings* 217, S. 109954. DOI: 10.1016/j.enbuild.2020.109954.
- European Environment Agency (Hg.) (2021): Copernicus Land Monitoring Service - User Manual. CORINE Land Cover. Online verfügbar unter <https://land.copernicus.eu/en/technical-library/clc-product-user-manual/@@download/file>.

Fan, Chao (2022): Integrating Human Mobility and Infrastructure Design in Digital Twin to Improve Equity and Resilience of Cities. In: 2022 IEEE 2nd International Conference on Digital Twins and Parallel Intelligence (DTPI). 2022 IEEE 2nd International Conference on Digital Twins and Parallel Intelligence (DTPI). Boston, MA, USA, 24.10.2022 - 28.10.2022: IEEE, S. 1–2.

Fan, Chao; Jiang, Yucheng; Mostafavi, Ali (2020): Social Sensing in Disaster City Digital Twin: Integrated Textual–Visual–Geo Framework for Situational Awareness during Built Environment Disruptions. In: *J. Manage. Eng.* 36 (3), Artikel 04020002. DOI: 10.1061/(ASCE)ME.1943-5479.0000745.

Fauzi, Muhammad Ashraf (2023): Social media in disaster management: review of the literature and future trends through bibliometric analysis. In: *Nat Hazards* 118 (2), S. 953–975. DOI: 10.1007/s11069-023-06079-7.

Fras, Mojca Kosmatin; Sustersic, Katja; Kezul, Aleksandar Sasic (2021): True Orthophoto in urban areas.

Gadde, Hemanth (2023): Leveraging AI for Scalable Query Processing in Big Data Environments. *International Journal of Advanced Engineering Technologies and Innovations*.

Ge, Chenyu; Qin, Shengfeng (2025): Urban flooding digital twin system framework. In: *Systems Science & Control Engineering* 13 (1), Artikel 2460432. DOI: 10.1080/21642583.2025.2460432.

GEOPLEX (Hg.) (2026): PlexMap 3D. Online verfügbar unter <https://www.geoplex.de/apps/3d-stadtmodelle/>.

Ghaith, Maysara; Yosri, Ahmed; El-Dakhakhni, Wael (2022a): Digital Twin: A City-Scale Flood Imitation Framework. In: Scott Walbridge, Mazdak Nik-Bakht, Kelvin Tsun Wai Ng, Manas Shome, M. Shahria Alam, Ashraf El Damatty und Gordon Lovegrove (Hg.): *Proceedings of the Canadian Society of Civil Engineering Annual Conference 2021*, Bd. 250. Singapore: Springer Nature Singapore (Lecture Notes in Civil Engineering), S. 577–588.

Ghaith, Maysara; Yosri, Ahmed; El-Dakhakhni, Wael (2022b): Synchronization-Enhanced Deep Learning Early Flood Risk Predictions: The Core of Data-Driven City Digital Twins for Climate Resilience Planning. In: *Water* 14 (22), S. 3619. DOI: 10.3390/w14223619.

Gholami, Mansoureh; Torreggiani, Daniele; Tassinari, Patrizia; Barbaresi, Alberto (2022): Developing a 3D City Digital Twin: Enhancing Walkability through a Green Pedestrian Network (GPN) in the City of Imola, Italy. In: *Land* 11 (11), S. 1917. DOI: 10.3390/land11111917.

Grieves, Michael (2015): Digital Twin: Manufacturing Excellence through Virtual Factory Replication. University of Central Florida.

Grieves, Michael; Vickers, John (2017): Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. In: Franz-Josef Kahlen, Shannon Flumerfelt und Anabela Alves (Hg.): Transdisciplinary Perspectives on Complex Systems. Cham: Springer International Publishing, S. 85–113.

Habib, Ahed; Habib, Maan; Bashir, Bashar; Bachir, Hussein (2025): Exploring the Sustainability Benefits of Digital Twin Technology in Achieving Resilient Smart Cities During Strong Earthquake Events. In: *Arab J Sci Eng* 50 (20), S. 16869–16883. DOI: 10.1007/s13369-025-10017-z.

Hai, Rihan; Koutras, Christos; Quix, Christoph; Jarke, Matthias (2023): Data Lakes: A Survey of Functions and Systems. In: *IEEE Trans. Knowl. Data Eng.* 35 (12), S. 12571–12590. DOI: 10.1109/TKDE.2023.3270101.

Ham, Youngjib; Kim, Jaeyoon (2020): Participatory Sensing and Digital Twin City: Updating Virtual City Models for Enhanced Risk-Informed Decision-Making. In: *J. Manage. Eng.* 36 (3), Artikel 04020005. DOI: 10.1061/(ASCE)ME.1943-5479.0000748.

Hämäläinen, Mervi (2020): Smart city development with digital twin technology. In: 33rd Bled eConference – Enabling Technology for a Sustainable Society: June 28 – 29, 2020, Online Conference Proceedings. Enabling Technology for a Sustainable Society, June 28 – 29, 2020: University of Maribor Press, S. 291–303.

Hertrich, Moritz (2024): CNN-basierte semantische Segmentierung von Gebäudetypen mittels hochauflösender Luftbilder und normalisierten digitalen Oberflächenmodellen. Masterarbeit. Katholische Universität, Eichstätt-Ingolstadt. Mathematisch-Geographische Fakultät.

Hlal, Mohammed; Baraka Munyaka, Jean-Claude; Chenal, Jérôme; Azmi, Rida; Diop, El Bachir; Bounabi, Mariem et al. (2025): Digital Twin Technology for Urban Flood Risk Management: A Systematic Review of Remote Sensing Applications and Early Warning Systems. In: *Remote Sensing* 17 (17), S. 3104. DOI: 10.3390/rs17173104.

Hoffmann, Eike Jens; Ali, Mohsin; Zhu, Xiao Xiang (2021): Zooming into Uncertainties: Towards Fusing Multi Zoom Level Imagery for Urban Land Use Segmentation. In: 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS. IGARSS 2021 - 2021 IEEE International Geoscience and Remote Sensing Symposium. Brussels, Belgium, 11.07.2021 - 16.07.2021: IEEE, S. 2090–2093.

Hoffmann, Eike Jens; Wang, Yuanyuan; Werner, Martin; Kang, Jian; Zhu, Xiao Xiang (2019): Model Fusion for Building Type Classification from Aerial and Street View Images. In: *Remote Sensing* 11 (11), S. 1259. DOI: 10.3390/rs11111259.

Hu, Wanyang; Wang, Shuhong; Zhai, Wei (2023): Human-centric vs. technology-centric approaches in a top-down smart city development regime: Evidence from 341 Chinese cities. In: *Cities* 137, S. 104271. DOI: 10.1016/j.cities.2023.104271.

Hu, Zhihui; Yao, Yao; Zhu, Qia; Guo, Zijin; Li, Guicheng; Li, Peiran et al. (2026): From human mobility to building functions: A deep learning approach for urban building classification in Megacity Tokyo. In: *Cities* 169, S. 106557. DOI: 10.1016/j.cities.2025.106557.

Huang, Xin; Li, Shuang; Li, Jiayi; Jia, Xiuping; Li, Jun; Zhu, Xiao Xiang; Benediktsson, Jon Atli (2021): A Multispectral and Multiangle 3-D Convolutional Neural Network for the Classification of ZY-3 Satellite Images Over Urban Areas. In: *IEEE Trans. Geosci. Remote Sensing* 59 (12), S. 10266–10285. DOI: 10.1109/TGRS.2020.3037211.

Hussain, Ejaz; Shan, Jie (2016): Urban building extraction through object-based image classification assisted by digital surface model and zoning map. In: *International Journal of Image and Data Fusion* 7 (1), S. 63–82. DOI: 10.1080/19479832.2015.1119206.

INSPIRE Maintenance and Implementation Group (MIG) (Hg.) (2024): D2.8.III.2 Data Specification on Buildings – Technical Guidelines. Online verfügbar unter https://inspire-mif.github.io/technical-guidelines/data/bu/dataspecification_bu.pdf.

Jafari, Mina; Kavousi-Fard, Abdollah; Chen, Tao; Karimi, Mazaher (2023): A Review on Digital Twin Technology in Smart Grid, Transportation System and Smart City: Challenges and Future. In: *IEEE Access* 11, S. 17471–17484. DOI: 10.1109/ACCESS.2023.3241588.

Janowicz, Krzysztof; Gao, Song; McKenzie, Grant; Hu, Yingjie; Bhaduri, Budhendra (2020): GeoAI: spatially explicit artificial intelligence techniques for geographic knowledge discovery and beyond. In: *International Journal of Geographical Information Science* 34 (4), S. 625–636. DOI: 10.1080/13658816.2019.1684500.

Kaewunruen, Sakdirat; AbdelHadi, Mohannad; Kongpuang, Manwika; Pansuk, Withit; Remennikov, Alex M. (2022): Digital Twins for Managing Railway Bridge Maintenance, Resilience, and Climate Change Adaptation. In: *Sensors (Basel, Switzerland)* 23 (1). DOI: 10.3390/s23010252.

Kaginalkar, Akshara; Kumar, Shamita; Gargava, Prashant; Niyogi, Dev (2021): Review of urban computing in air quality management as smart city service: An integrated IoT, AI, and cloud technology perspective. In: *Urban Climate* 39, S. 100972. DOI: 10.1016/j.uclim.2021.100972.

Kang, Ji-Soo; Chung, Kyungyong; Hong, Ellen J. (2021): Multimedia knowledge-based bridge health monitoring using digital twin. In: *Multimed Tools Appl* 80 (26-27), S. 34609–34624. DOI: 10.1007/s11042-021-10649-x.

Karabulut, Erkan; Pileggi, Salvatore F.; Groth, Paul; Degeler, Victoria (2024): Ontologies in digital twins: A systematic literature review. In: *Future Generation Computer Systems* 153, S. 442–456. DOI: 10.1016/j.future.2023.12.013.

Karatvuo, Helena; Linde, Michael; Dolatshah, Azam; Mortensen, Simon (2022): Improved Climate Change Adaptation in Port of Brisbane Using a Digital Twin Cloud-Based Modelling Approach. In: Volume 1: Offshore Technology. ASME 2022 41st International Conference on Ocean, Offshore and Arctic Engineering. Hamburg, Germany, 05.06.2022 - 10.06.2022: American Society of Mechanical Engineers.

Ketzler, Bernd; Naserentin, Vasilis; Latino, Fabio; Zangelidis, Christopher; Thuvander, Liane; Logg, Anders (2020): Digital Twins for Cities: A State of the Art Review. In: *built environ* 46 (4), S. 547–573. DOI: 10.2148/benv.46.4.547.

Khajavi, Siavash H.; Motlagh, Naser Hossein; Jaribion, Alireza; Werner, Liss C.; Holmstrom, Jan (2019): Digital Twin: Vision, Benefits, Boundaries, and Creation for Buildings. In: *IEEE Access* 7, S. 147406–147419. DOI: 10.1109/ACCESS.2019.2946515.

Kim, Jooho; Hatzis, Joshua J.; Klockow, Kim; Campbell, Patrick A. (2022): Building Classification Using Random Forest to Develop a Geodatabase for Probabilistic Hazard Information. In: *Nat. Hazards Rev.* 23 (3), Artikel 04022014. DOI: 10.1061/(ASCE)NH.1527-6996.0000561.

Kim, Yeji; Oh, Jeil; Bartos, Matthew (2025): Stormwater digital twin with online quality control detects urban flood hazards under uncertainty. In: *Sustainable Cities and Society* 118, S. 105982. DOI: 10.1016/j.scs.2024.105982.

Kocaman, Sultan; Akca, Devrim; Poli, Daniela; Remondino, Fabio (Hg.) (2024): 3D/4D city modelling. From sensors to applications. Dunbeath: Whittles Publishing.

Kritzinger, Werner; Karner, Matthias; Traar, Georg; Henjes, Jan; Sihn, Wilfried (2018): Digital Twin in manufacturing: A categorical literature review and classification. In: *IFAC-PapersOnLine* 51 (11), S. 1016–1022. DOI: 10.1016/j.ifacol.2018.08.474.

Kusic, Kresimir; Schumann, Rene; Ivanjko, Edouard (2023): A digital twin in transportation. Real-time synergy of traffic data streams and simulation for virtualizing motorway dynamics. Hg. v. Advanced Engineering Informatics.

Land Monitoring Service (Hg.) (2026): Copernicus Land Monitoring Service – Priority Area Monitoring – Urban Atlas 2021 and 2024. URBAN ATLAS 2021 - PRODUCT USER MANUAL (PUM). Online verfügbar unter <https://land.copernicus.eu/en/technical-library/product-user-manual-urban-atlas-2021/@@download/file>.

Lang, Kevin J.; Waibel, Alex H.; Hinton, Geoffrey E. (1990): A time-delay neural network architecture for isolated word recognition. In: *Neural Networks* 3 (1), S. 23–43. DOI: 10.1016/0893-6080(90)90044-L.

Le Yang; Tian, Miao; Xin, Duan; Cheng, Qishuo; Zheng, Jiajian (2024): AI-Driven Anonymization: Protecting Personal Data Privacy While Leveraging Machine Learning.

Lee, Lik-Hang; Braud, Tristan; Zhou, Peng Yuan; Wang, Lin; Xu, Dianlei; Lin, Zijun et al. (2024): All One Needs to Know about Metaverse: A Complete Survey on Technological Singularity, Virtual Ecosystem, and Research Agenda. In: *FNT in Human-Computer Interaction* 18 (2–3), S. 100–337. DOI: 10.1561/11000000095.

Lehner, Hubert; Dorffner, Lionel (2020): Digital geoTwin Vienna: Towards a Digital Twin City as Geodata Hub. In: *PFG* 88 (1), S. 63–75. DOI: 10.1007/s41064-020-00101-4.

Leplat, Léo; López-Alfaro, Claudia; Styve, Arne; Da Torres, Ricardo Silva (2025): NorDark-DT: A digital twin for urban lighting infrastructure planning and analysis. In: *Environment and Planning B: Urban Analytics and City Science* 52 (3), S. 667–688. DOI: 10.1177/23998083241272099.

Lewis, Ryan Hardesty; Jiao, Junfeng; Seong, Kijin; Farahi, Arya; Navrátil, Paul; Casebeer, Nate; Niyogi, Dev (2024): Fire and smoke digital twin – A computational framework for modeling fire incident outcomes. In: *Computers, Environment and Urban Systems* 110, S. 102093. DOI: 10.1016/j.compenvurbsys.2024.102093.

Li, Jie; Gao, Jun; Zhang, Zhonghao; Fu, Jing; Shao, Guofan; Zhao, Zhenyu; Yang, Panpan (2024a): Insights into citizens' experiences of cultural ecosystem services in urban green spaces based on social media analytics. In: *Landscape and Urban Planning* 244, S. 104999. DOI: 10.1016/j.landurbplan.2023.104999.

Li, Pengzhi; Pei, Yan; Li, Jianqiang (2023): A comprehensive survey on design and application of autoencoder in deep learning. In: *Applied Soft Computing* 138, S. 110176. DOI: 10.1016/j.asoc.2023.110176.

Li, Ruinian; Song, Tianyi; Mei, Bo; Li, Hong; Cheng, Xiuzhen; Sun, Limin (2019): Blockchain for Large-Scale Internet of Things Data Storage and Protection. In: *IEEE Trans. Serv. Comput.* 12 (5), S. 762–771. DOI: 10.1109/TSC.2018.2853167.

Li, Wenwen; Arundel, Samantha; Gao, Song; Goodchild, Michael; Hu, Yingjie; Wang, Shaowen; Zipf, Alexander (2024b): GeoAI for Science and the Science of GeoAI. In: *JOSIS* (29), S. 1–17. DOI: 10.5311/JOSIS.2024.29.349.

Li, Xianzhi; Li, Ruihui; Zhu, Lei; Fu, Chi-Wing; Heng, Pheng-Ann (2021): DNF-Net: A Deep Normal Filtering Network for Mesh Denoising. In: *IEEE transactions on visualization and computer graphics* 27 (10), S. 4060–4072. DOI: 10.1109/TVCG.2020.3001681.

Lieberman, Joshua; Leidner, Alan; Percivall, George; Ronsdorf, Carsten (2017): Using big data analytics and IoT principles to keep an eye on underground infrastructure. In: 2017 IEEE International Conference on Big Data (Big Data). 2017 IEEE International Conference on Big Data (Big Data). Boston, MA, 11.12.2017 - 14.12.2017: IEEE, S. 4592–4601.

Liu, Weiquan; Zang, Yu; Xiong, Zhangyue; Bian, Xuesheng; Wen, Chenglu; Lu, Xiaolei et al. (2023): 3D building model generation from MLS point cloud and 3D mesh using multi-source data fusion. In: *International Journal of Applied Earth Observation and Geoinformation* 116, S. 103171. DOI: 10.1016/j.jag.2022.103171.

Liu, Yongkang; Wang, Ziran; Han, Kyungtae; Shou, Zhenyu; Tiwari, Prashant; L. Hansen, John H. (2020): Sensor Fusion of Camera and Cloud Digital Twin Information for Intelligent Vehicles. In: 2020 IEEE Intelligent Vehicles Symposium (IV). 2020 IEEE Intelligent Vehicles Symposium (IV). Las Vegas, NV, USA, 19.10.2020 - 13.11.2020: IEEE, S. 182–187.

Lu, Feng; Cheng, Shifen; Wang, Peixiao (2025): GeoAI enabled urban computing: status and challenges. In: *Annals of GIS* 31 (4), S. 537–555. DOI: 10.1080/19475683.2025.2552152.

Lu, Qiuchen; Parlikad, Ajith Kumar; Woodall, Philip; Don Ranasinghe, Gishan; Xie, Xiang; Liang, Zhenglin et al. (2020): Developing a Digital Twin at Building and City Levels: Case Study of West Cambridge Campus. In: *J. Manage. Eng.* 36 (3), Artikel 05020004. DOI: 10.1061/(ASCE)ME.1943-5479.0000763.

Lucas, Maxime; Gallo, Luca; Ghavasieh, Arsham; Battiston, Federico; Domenico, Manlio de (2026): Reducibility of higher-order networks from dynamics. In: *Nature communications* 17 (1), S. 1551. DOI: 10.1038/s41467-025-68273-4.

Madakam, Somayya; Ramaswamy, R.; Tripathi, Siddharth (2015): Internet of Things (IoT): A Literature Review. In: *JCC* 03 (05), S. 164–173. DOI: 10.4236/jcc.2015.35021.

Mai, Gengchen; Hu, Yingjie; Gao, Song; Cai, Ling; Martins, Bruno; Scholz, Johannes et al. (2022): Symbolic and subsymbolic GeoAI : Geospatial knowledge graphs and spatially explicit machine learning. In: *Transactions in GIS* 26 (8), S. 3118–3124. DOI: 10.1111/tgis.13012.

Major, Pierre; Li, Guoyuan; Hildre, Hans Petter; Zhang, Houxiang (2021): The Use of a Data-Driven Digital Twin of a Smart City: A Case Study of Ålesund, Norway. In: *IEEE Instrum. Meas. Mag.* 24 (7), S. 39–49. DOI: 10.1109/MIM.2021.9549127.

Maurer, Tom (2017): What is a digital twin. Siemens Product Lifecycle Management Software Inc.

Mavrokapnidis, Dimitri; Mohammadi, Neda; Taylor, John (2021): Community Dynamics in Smart City Digital Twins: A Computer Vision-based Approach for Monitoring and Forecasting Collective Urban Hazard Exposure. In: Tung Bui (Hg.): Proceedings of the 54th Hawaii International Conference on System Sciences. Hawaii International Conference on System Sciences: Hawaii International Conference on System Sciences (Proceedings of the Annual Hawaii International Conference on System Sciences).

- Merenda, Massimo; Porcaro, Carlo; Iero, Demetrio (2020): Edge Machine Learning for AI-Enabled IoT Devices: A Review. In: *Sensors (Basel, Switzerland)* 20 (9). DOI: 10.3390/s20092533.
- Miller, Harvey J.; Goodchild, Michael F. (2015): Data-driven geography. In: *GeoJournal* 80 (4), S. 449–461. DOI: 10.1007/s10708-014-9602-6.
- Mortaheb, Reza; Jankowski, Piotr (2023): Smart city re-imagined: City planning and GeoAI in the age of big data. In: *Journal of Urban Management* 12 (1), S. 4–15. DOI: 10.1016/j.jum.2022.08.001.
- Nys, G.-A.; Billen, R.; Poux, F. (2020): AUTOMATIC 3D BUILDINGS COMPACT RECONSTRUCTION FROM LIDAR POINT CLOUDS. In: *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XLIII-B2-2020, S. 473–478. DOI: 10.5194/isprs-archives-XLIII-B2-2020-473-2020.
- Nystuen, John D. (1984): Comment On “Artificial Intelligence And Its Applicability To Geographical Problem Solving”. In: *The Professional Geographer* 36 (3), S. 358–359. DOI: 10.1111/j.0033-0124.1984.00358.x.
- Ohori, Ken; Ledoux, Hugo; Biljecki, Filip; Stoter, Jantien (2015): Modeling a 3D City Model and Its Levels of Detail as a True 4D Model. In: *IJGI* 4 (3), S. 1055–1075. DOI: 10.3390/ijgi4031055.
- Omar, Osama (2025): Digital Twins for Climate-Responsive Urban Development: Integrating Zero-Energy Buildings into Smart City Strategies. In: *Sustainability* 17 (15), S. 6670. DOI: 10.3390/su17156670.
- Omrany, Hossein; Al-Obaidi, Karam M. (2025): Application of digital twin technology for Urban Heat Island mitigation: review and conceptual framework. In: *Smart and Sustainable Built Environment* 14 (7), S. 2145–2175. DOI: 10.1108/SASBE-05-2024-0189.
- Open Geospatial Consortium (Hg.) (2021): OGC City Geography Markup Language (CityGML) Part 1: Conceptual Model Standard. Online verfügbar unter <https://docs.ogc.org/is/20-010/20-010.html#toc45>.
- Openshaw, Stan (1992): Some suggestions concerning the development of artificial intelligence tools for spatial modelling and analysis in GIS. Hg. v. The Annals of Regional Science. Newcastle upon Tyne.
- Openshaw, Stan; Openshaw, Christine (1997): Artificial intelligence in geography. Chichester, West Sussex: Wiley. Online verfügbar unter <http://www.loc.gov/catdir/description/wiley037/96045100.html>.
- Paden, Ivan; Peters, Ravi; Garcia-Sanchez, Clara; Ledoux, Hugo (2024): Automatic high-detailed building reconstruction workflow for urban microscale simulations. Hg. v. Building and Environment.

Pan, Zhuokun; Xu, Jiashu; Guo, Yubin; Hu, Yueming; Wang, Guangxing (2020): Deep Learning Segmentation and Classification for Urban Village Using a Worldview Satellite Image Based on U-Net. In: *Remote Sensing* 12 (10), S. 1574. DOI: 10.3390/rs12101574.

Petrova-Antonova, Dessislava; Ilieva, Sylvia (2019): Methodological Framework for Digital Transition and Performance Assessment of Smart Cities. In: 2019 4th International Conference on Smart and Sustainable Technologies (SpliTech). 2019 4th International Conference on Smart and Sustainable Technologies (SpliTech). Split, Croatia, 18.06.2019 - 21.06.2019: IEEE, S. 1–6.

Prasath, C.Arun; Vishnupriya, T. (2025): AI-Enabled Digital Twin Framework for Predictive Maintenance in Smart Urban Infrastructure. In: *Journal of Smart Infrastructure and Environmental Sustainability* 2025, 10.02.2025 (Vol 2).

Rahman, Md. Abdur; Rashid, Md. Mamunur; Hossain, M. Shamim; Hassanain, Elham; Alhamid, Mohammed F.; Guizani, Mohsen (2019): Blockchain and IoT-Based Cognitive Edge Framework for Sharing Economy Services in a Smart City. In: *IEEE Access* 7, S. 18611–18621. DOI: 10.1109/ACCESS.2019.2896065.

Roudbari, Naghmeh Shafiee; Puneekar, Shubham Rajeev; Patterson, Zachary; Eicker, Ursula; Poullis, Charalambos (2024): From data to action in flood forecasting leveraging graph neural networks and digital twin visualization. In: *Scientific reports* 14 (1), S. 18571. DOI: 10.1038/s41598-024-68857-y.

Roy, Santanu; Bhunia, Gouri Sankar; Shit, Pravat Kumar (2021): Spatial prediction of COVID-19 epidemic using ARIMA techniques in India. In: *Modeling earth systems and environment* 7 (2), S. 1385–1391. DOI: 10.1007/s40808-020-00890-y.

Rozony, Farhana Zaman; Aktar, Mst Nahida Aktar; Ashrafuzzaman, Md; Islam, Ashraful (2024): A SYSTEMATIC REVIEW OF BIG DATA INTEGRATION CHALLENGES AND SOLUTIONS FOR HETEROGENEOUS DATA SOURCES. In: *AJBAS* 4 (4), S. 1–18. DOI: 10.69593/ajbais.v4i04.111.

Ruohomaki, Timo; Airaksinen, Enni; Huuska, Petteri; Kesaniemi, Outi; Martikka, Mikko; Suomisto, Jarmo (2018): Smart City Platform Enabling Digital Twin. In: 2018 International Conference on Intelligent Systems (IS). 2018 International Conference on Intelligent Systems (IS). Funchal - Madeira, Portugal, 25.09.2018 - 27.09.2018: IEEE, S. 155–161.

Sabbata, Stefano de; Ballatore, Andrea; Miller, Harvey J.; Sieber, Renée; Tyukin, Ivan; Yeboah, Godwin (2023): GeoAI in urban analytics. In: *International Journal of Geographical Information Science* 37 (12), S. 2455–2463. DOI: 10.1080/13658816.2023.2279978.

Salem, Fathi M. (Hg.) (2022): Recurrent Neural Networks. Cham: Springer International Publishing.

- Santosh, K. C.; Das, Nibaran; Ghosh, Swarnendu (2021): Deep Learning Models for Medical Imaging: Academic Press.
- Sarbout, Ilias; Ounissi, Mehdi; Racoceanu, Daniel; Milea, Dan (2024): AI-powered autonomous mobility system assisting blind digital twin. Hg. v. EUNOS 2024 - Congress of the European Neuro-ophthalmology Society. Rotterdam.
- Savage, Thomas; Akroyd, Jethro; Mosbach, Sebastian; Krdzavac, Nenad; Hillman, Michael; Kraft, Markus (2022): Universal Digital Twin: Integration of national-scale energy systems and climate data. In: *DCE* 3. DOI: 10.1017/dce.2022.22.
- Scheider, Simon; Richter, Kai-Florian (2023): GeoAI. In: *Künstl Intell* 37 (1), S. 5–9. DOI: 10.1007/s13218-022-00797-z.
- Schrotter, Gerhard; Hürzeler, Christian (2020): The Digital Twin of the City of Zurich for Urban Planning. In: *PFG* 88 (1), S. 99–112. DOI: 10.1007/s41064-020-00092-2.
- Senyurdusev, Guclu; Ulugtekin, Necla; Dogru, Ahmet Ozgur (2020): Exploring the opportunities of open source data use in creation 3D procedural city models. Hg. v. Conference: 8th International Conference on Cartography and GIS. Nessebar.
- Shahat, Ehab; Hyun, Chang T.; Yeom, Chunho (2021): City Digital Twin Potentials: A Review and Research Agenda. In: *Sustainability* 13 (6), S. 3386. DOI: 10.3390/su13063386.
- Shahrour, Isam; Xie, Xiongyao (2021): Role of Internet of Things (IoT) and Crowdsourcing in Smart City Projects. In: *Smart Cities* 4 (4), S. 1276–1292. DOI: 10.3390/smartcities4040068.
- Sherstinsky, Alex (2020): Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. In: *Physica D: Nonlinear Phenomena* 404, S. 132306. DOI: 10.1016/j.physd.2019.132306.
- SIG3D AG Qualität (Hg.) (2014): Handbuch für die Modellierung von 3D Objekten. Teil 2: Modellierung Gebäude (LoD1, LoD2 und LoD 3). Online verfügbar unter https://www.sig3d.org/files/media/downloads/SIG3D/AG-Qualitaet/201311_SIG3D_Modellierungshandbuch_Teil_2.pdf.
- Simic, Mirko; Peric, Miroslav; Popadic, Ilija; Peric, Dragana; Pavlovic, Milos; Vucetic, Miljan; Stankovic, Milos (2020): Big Data and development of smart city: System architecture and practical public safety example. In: *Serb J Electr Eng* 17 (3), S. 337–355. DOI: 10.2298/SJEE2003337S.
- Singh, Sumit; Shehab, Essam; Higgins, Nigel; Fowler, Kevin; Reynolds, Dylan; Erkoyuncu, John A.; Gadd, Peter (2021): Data management for developing digital twin ontology model. In: *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 235 (14), S. 2323–2337. DOI: 10.1177/0954405420978117.

Smith, Terence R. (1984): ARTIFICIAL INTELLIGENCE AND ITS APPLICABILITY TO GEOGRAPHICAL PROBLEM SOLVING. In: *The Professional Geographer* 36 (2), S. 147–158. DOI: 10.1111/j.0033-0124.1984.00147.x.

Statistik Austria (Hg.) (2022a): Adress-, Gebäude und Wohnungsregister. Bundesanstalt Statistik Österreich. Wien.

Statistik Austria (Hg.) (2022b): Adress-, Gebäude und Wohnungsregister. Überblick. Online verfügbar unter https://www.statistik.at/fileadmin/pages/489/AGWR_Ueberblick_2024_09.pdf.

Strauch, Christof: NoSQL Databases. Hochschule der Medien, Stuttgart.

Streltsov, Artem; Malof, Jordan M.; Huang, Bohao; Bradbury, Kyle (2020): Estimating residential building energy consumption using overhead imagery. In: *Applied Energy* 280, S. 116018. DOI: 10.1016/j.apenergy.2020.116018.

Syamimi, Athirah; Gong, Yiwei; Liew, Ryan (2020): VR industrial applications—A singapore perspective. In: *Virtual Reality & Intelligent Hardware* 2 (5), S. 409–420. DOI: 10.1016/j.vrih.2020.06.001.

Syed, Aamir; Purushotham, Keerthana; Shidaganti, Ganeshayya (2020): Cloud Storage Security Risks, Practices and Measures: A Review. In: 2020 IEEE International Conference for Innovation in Technology (INOCON). 2020 IEEE International Conference for Innovation in Technology (INOCON). Bangluru, India, 06.11.2020 - 08.11.2020: IEEE, S. 1–4.

Tama, Bayu Adhi; Rhee, Kyung Hyune (2017): Attack Classification Analysis of IoT Network via Deep Learning Approach. Research Briefs on Information & Communication Technology Evolution. Hg. v. Department of IT Convergence and Application Engineering Pukyong National University.

Tegtmeyer, Sascha; Schubbe, Nicole; Gras, Pierre; Krause, Kai-Uwe (2022): Digitale Zwillinge und Datenplattformen - Vernetzung in alle Richtungen am Beispiel der Stadt Hamburg. Urban Data Plattform Hamburg. Hamburg.

The OpenFOAM Foundation (Hg.) (2026). Online verfügbar unter <https://openfoam.org/version/13/>.

Varlamis, Iraklis; Sardianos, Christos; Chronis, Christos; Dimitrakopoulos, George; Himeur, Yassine; Alsalemi, Abdullah et al. (2023): Using big data and federated learning for generating energy efficiency recommendations. In: *Int J Data Sci Anal* 16 (3), S. 353–369. DOI: 10.1007/s41060-022-00331-2.

Villani, Lorenzo; Gugliermetti, Luca; Barucco, Maria Antonia; Cinquepalmi, Federico (2025): A Digital Twin Framework to Improve Urban Sustainability and Resiliency: The Case Study of Venice. In: *Land* 14 (1), S. 83. DOI: 10.3390/land14010083.

- Wang, Boyu; Wang, Qian; Cheng, Jack C.P.; Yin, Chao (2022a): Object verification based on deep learning point feature comparison for scan-to-BIM. In: *Automation in Construction* 142, S. 104515. DOI: 10.1016/j.autcon.2022.104515.
- Wang, Chaoqun; He, Jie; Pan, Weijiang; Xu, Xin (2025): Application of Geographic Information Artificial Intelligence in Spatio-Temporal Semantic Extraction. In: 2025 32nd International Conference on Geoinformatics. 2025 32nd International Conference on Geoinformatics. Jiaozuo, China, 16.06.2025 - 18.06.2025: IEEE, S. 1–17.
- Wang, Dong; Xie, Wenjun; Cai, Youcheng; Li, Xinjie; Liu, Xiaoping (2023a): Multi-order spatial interaction network for human pose estimation. In: *Digital Signal Processing* 142, S. 104219. DOI: 10.1016/j.dsp.2023.104219.
- Wang, Ge; Badal, Andreu; Jia, Xun; Maltz, Jonathan S.; Mueller, Klaus; Myers, Kyle J. et al. (2022b): Development of metaverse for intelligent healthcare. In: *Nature machine intelligence* 4 (11), S. 922–929. DOI: 10.1038/s42256-022-00549-6.
- Wang, Hao; Chen, Xiaowei; Jia, Fu; Cheng, Xiaojuan (2023b): Digital twin-supported smart city: Status, challenges and future research directions. In: *Expert Systems with Applications* 217, S. 119531. DOI: 10.1016/j.eswa.2023.119531.
- Wang, Hao; Pan, Yisha; Luo, Xiaochun (2019a): Integration of BIM and GIS in sustainable built environment: A review and bibliometric analysis. In: *Automation in Construction* 103, S. 41–52. DOI: 10.1016/j.autcon.2019.03.005.
- Wang, Huiyue; Chen, Dongjie; Duan, Huabo; Yin, Fengfu; Niu, Yongning (2019b): Characterizing urban building metabolism with a 4D-GIS model: A case study in China. In: *Journal of Cleaner Production* 228, S. 1446–1454. DOI: 10.1016/j.jclepro.2019.04.341.
- Watanabe, Alice; Schelling, Kathrin; Go, Stefanie; Schmohl, Tobias (2022): Die KI-Frage. Hg. v. DELFI Workshops 2022.
- Weil, Charlotte; Bibri, Simon Elias; Longchamp, Régis; Golay, François; Alahi, Alexandre (2023): Urban Digital Twin Challenges: A Systematic Review and Perspectives for Sustainable Smart Cities. In: *Sustainable Cities and Society* 99, S. 104862. DOI: 10.1016/j.scs.2023.104862.
- White, Gary; Zink, Anna; Codecá, Lara; Clarke, Siobhán (2021): A digital twin smart city for citizen feedback. In: *Cities* 110, S. 103064. DOI: 10.1016/j.cities.2020.103064.
- Xia, Haishan; Liu, Zishuo; Efremochkina, Maria; Liu, Xiaotong; Lin, Chunxiang (2022): Study on city digital twin technologies for sustainable smart city design: A review and bibliometric analysis of geographic information system and building information modeling integration. In: *Sustainable Cities and Society* 84, S. 104009. DOI: 10.1016/j.scs.2022.104009.

Xu, Haowen; Berres, Andy; Yoginath, Srikanth B.; Sorensen, Harry; Nugent, Phil J.; Severino, Joseph et al. (2023): Smart Mobility in the Cloud: Enabling Real-Time Situational Awareness and Cyber-Physical Control Through a Digital Twin for Traffic. In: *IEEE Trans. Intell. Transport. Syst.* 24 (3), S. 3145–3156. DOI: 10.1109/TITS.2022.3226746.

Xu, Haowen; Shao, Yunli; Chen, Jianfei; Wang, Chieh; Berres, Andy (2024): Semi-Automatic Geographic Information System Framework for Creating Photo-Realistic Digital Twin Cities to Support Autonomous Driving Research. In: *Transportation Research Record: Journal of the Transportation Research Board* 2678 (6), S. 1068–1084. DOI: 10.1177/03611981231205884.

Xu, Haowen; Sun, Yulin; Tupayachi, Jose; Omitaomu, Olufemi; Zlatanova, Sisi; Li, Xueping (2025): Towards the Autonomous Optimization of Urban Logistics: Training Generative AI with Scientific Tools via Agentic Digital Twins and Model Context Protocol.

Xu, Zhen; Qi, Mingzhu; Wu, Yingying; Hao, Xintian; Yang, Yajun (2021): City Information Modeling: State of the Art. In: *Applied Sciences* 11 (19), S. 9333. DOI: 10.3390/app11199333.

Yan, Mingyu; Gan, Wei; Zhou, Yue; Wen, Jianfeng; Yao, Wei (2022): Projection method for blockchain-enabled non-iterative decentralized management in integrated natural gas-electric systems and its application in digital twin modelling. In: *Applied Energy* 311, S. 118645. DOI: 10.1016/j.apenergy.2022.118645.

Ye, Nanqi; Wang, Bowen; Kita, Michihiro; Xie, Ming; Cai, Wenyue (2019): Urban Commerce Distribution Analysis Based on Street View and Deep Learning. In: *IEEE Access* 7, S. 162841–162849. DOI: 10.1109/ACCESS.2019.2951294.

Yi, Kun; Zhang, Qi; He, Hui; Shi, Kaize; Hu, Liang; An, Ning; Niu, Zhendong (2024): Deep Coupling Network for Multivariate Time Series Forecasting. In: *ACM Trans. Inf. Syst.* 42 (5), S. 1–28. DOI: 10.1145/3653447.

Ying, Yue; Koeva, Mila; Kuffer, Monika; Zevenbergen, Jaap (2023): Toward 3D Property Valuation—A Review of Urban 3D Modelling Methods for Digital Twin Creation. In: *IJGI* 12 (1), S. 2. DOI: 10.3390/ijgi12010002.

Yu, Xiaohui; Merritt, Jeff (2022): Digital Twin Cities: Framework and Global Practices. Hg. v. World Economic Forum.

Zaker, Reza; Coloma, Eloi (2018): Virtual reality-integrated workflow in BIM-enabled projects collaboration and design review: a case study. In: *Vis. in Eng.* 6 (1). DOI: 10.1186/s40327-018-0065-6.

Zhang, Jing; Tao, Dacheng (2021): Empowering Things With Intelligence: A Survey of the Progress, Challenges, and Opportunities in Artificial Intelligence of Things. In: *IEEE Internet Things J.* 8 (10), S. 7789–7817. DOI: 10.1109/JIOT.2020.3039359.

Zhang, Xiang (2024): City Information Modeling and Its Applications: A Review. In: Ali Cheshmehzangi, Michael Batty, Zaheer Allam und David S. Jones (Hg.): City Information Modelling. Singapore: Springer Nature Singapore (Urban Sustainability), S. 63–74.

Zhao, Dongmiao; Li, Xuefei; Wang, Xingtian; Shen, Xiang; Gao, Weijun (2022): Applying Digital Twins to Research the Relationship Between Urban Expansion and Vegetation Coverage: A Case Study of Natural Preserve. In: *Frontiers in plant science* 13, S. 840471. DOI: 10.3389/fpls.2022.840471.

Zhou, Zhi-Hua (2021): Machine learning. Singapore: Springer Singapore; Springer (Springer eBook Collection).

Zhu, Mingyu; Jin, Jiayi (2025): Data-Driven Urban Digital Twins and Critical Infrastructure Under Climate Change: A Review of Frameworks and Applications. In: *UP* 10, Artikel 10109. DOI: 10.17645/up.10109.

Anhang

Python-Code: THESIS.data_processing	i
Python-Code: THESIS.load_orthofoto	ii
Python-Code: THESIS.trainingTESTING	iii
Python-Code: THESIS.trainingFINAL	v
Python-Code: THESIS.evaluation	vii
Python-Code: THESIS.prediction	viii
Python-Code: SCRIPTS.enrich_data	ix
Python-Code: SCRIPTS.train_model	x
Python-Code: SCRIPTS.multiple_models	xii
Python-Code: SCRIPTS.feature_importance	xv
Python-Code: SCRIPTS.cross_validation	xvii
Python-Code: SCRIPTS.predict_all	xix

```

##THESIS.data_processing
import geopandas as gpd
import Pandas as pd
import osmnx as ox
from sklearn.model_selection import train_test_split
from sklearn.model_selection import StratifiedKFold

def load_buildings(path: str) -> gpd.GeoDataFrame:
    """Lädt das Shapefile mit Gebäude-Geometrien."""
    gdf = gpd.read_file(path)
    return gdf

def enrich_buildings(gdf: gpd.GeoDataFrame) -> gpd.GeoDataFrame:
    """
    Reichert Gebäude-Polygone mit zusätzlichen Attributen an.
    """
    # Fläche und Umfang
    gdf["area"] = gdf.geometry.area
    gdf["perimeter"] = gdf.geometry.length

    # Beispiel: Schwerpunktkoordinaten (Centroid)
    gdf["centroid_x"] = gdf.geometry.centroid.x
    gdf["centroid_y"] = gdf.geometry.centroid.y

    return gdf

def filter_labeled(gdf: gpd.GeoDataFrame, label_col: str = "INSPIRE") -> gpd.GeoDataFrame:
    """Filtert nur Gebäude, die ein Label besitzen (z. B. für Training)."""
    return gdf[gdf[label_col].notna()].copy()

def split_train_validation(gdf: gpd.GeoDataFrame, test_size=0.2, random_state=42):
    """Teilt die gelabelten Gebäude in Trainings- und Validierungsdaten."""
    train_gdf, val_gdf = train_test_split(
        gdf,
        test_size=test_size,
        random_state=random_state,
        stratify=gdf["INSPIRE"]
    )
    return train_gdf, val_gdf

##Formel für Cross-Validation
def cross_validate_gdf(gdf, n_splits=5, random_state=42):
    """Erzeugt Trainings- und Validierungs-Splits für Cross-Validation."""
    skf = StratifiedKFold(n_splits=n_splits, shuffle=True, random_state=random_state)
    folds = []
    X = gdf
    y = gdf["INSPIRE"]
    for train_idx, val_idx in skf.split(X, y):
        train_gdf = gdf.iloc[train_idx].copy()
        val_gdf = gdf.iloc[val_idx].copy()
        folds.append((train_gdf, val_gdf))
    return folds

def remove_attributes(gdf: gpd.GeoDataFrame, att_remove):
    # Prüfen, welche Attribute vorhanden sind
    existing_attributes = [attr for attr in att_remove if attr in gdf.columns]

    # Attribute entfernen
    gdf = gdf.drop(columns=existing_attributes)

    return gdf

```

```

##THESIS.load_orthofoto
import rasterio
from rasterio.mask import mask
import numpy as np

def calc_imagestats(gdf, raster_path, band_stats=['mean','std','min','max','bright_ratio','dark_ratio']):
    """
    Berechnet für jedes Polygon im GeoDataFrame verschiedene Bildstatistiken pro Band
    und speichert sie als neue Spalten im GeoDataFrame.
    """
    with rasterio.open(raster_path) as raster:
        # CRS abgleichen
        if gdf.crs != raster.crs:
            gdf = gdf.to_crs(raster.crs)
        band_count = raster.count
        # Spalten anlegen
        for b in range(1, band_count + 1):
            for stat in band_stats:
                col = f'band{b}_{stat}'
                gdf[col] = np.nan
        # Iteration über alle Polygone
        for idx, row in gdf.iterrows():
            geom = [row['geometry']]
            try:
                out_image, _ = mask(raster, geom, crop=True, filled=True, nodata=0)
                valid_mask = out_image[0] != 0 # Annahme: 0 = Hintergrund/Nodata
                for b in range(band_count):
                    band_data = out_image[b][valid_mask]
                    if band_data.size > 0:
                        # Mittelwert
                        if 'mean' in band_stats:
                            gdf.at[idx, f'band{b + 1}_mean'] = band_data.mean()
                        # Standardabweichung
                        if 'std' in band_stats:
                            gdf.at[idx, f'band{b + 1}_std'] = band_data.std()
                        # Minimum
                        if 'min' in band_stats:
                            gdf.at[idx, f'band{b + 1}_min'] = band_data.min()
                        # Maximum
                        if 'max' in band_stats:
                            gdf.at[idx, f'band{b + 1}_max'] = band_data.max()
                        # Anteil heller Pixel (>180)
                        if 'bright_ratio' in band_stats:
                            gdf.at[idx, f'band{b + 1}_bright_ratio'] = np.sum(band_data > 180) / band_data.size
                        # Anteil dunkler Pixel (<50)
                        if 'dark_ratio' in band_stats:
                            gdf.at[idx, f'band{b + 1}_dark_ratio'] = np.sum(band_data < 50) / band_data.size
            except Exception as e:
                print(f"Polygon {idx} konnte nicht verarbeitet werden: {e}")
    return gdf

```

```

##THESIS.trainingTESTING
import geopandas as gpd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier, ExtraTreesClassifier, BaggingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report
import joblib
import pandas as pd
from typing import Tuple, Dict
import os
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier

def train_model(gdf: gpd.GeoDataFrame, label_col: str = "INSPIRE") -> Tuple[RandomForestClassifier, LabelEncoder, Dict]:
    """Trainiert ein ML-Modell auf gelabelten Gebäudedaten."""

    # Features vorbereiten
    X = prepare_features(gdf, label_col=label_col)
    #X = X.reindex(columns=features_train, fill_value=0)

    # Labels encoden
    y = gdf[label_col]
    label_encoder = LabelEncoder()
    y_encoded = label_encoder.fit_transform(y)

    # Modell trainieren
    #print(f"Trainiere RandomForest auf {len(X)} Gebäuden ...")
    #clf = RandomForestClassifier(
    #    n_estimators=200,
    #    random_state=42,
    #    n_jobs=-1,
    #    class_weight="balanced"
    #)

    #ALTERNATIVE EXTRA TREES
    #print(f"Trainiere ExtraTrees auf {len(X)} Gebäuden ...")
    #clf = ExtraTreesClassifier(
    #    n_estimators=500,
    #    random_state=42,
    #    n_jobs=-1,
    #    class_weight="balanced"
    #)

    #ALTERNATIVE BAGGING
    #print(f"Trainiere Bagging auf {len(X)} Gebäuden ...")
    #clf = BaggingClassifier(
    #    n_estimators=150,
    #    random_state=42,
    #    n_jobs=-1,
    #)

    ##XGBClassifier
    #print(f"Trainiere XGBoost auf {len(X)} Gebäuden ...")
    #clf = XGBClassifier(
    #    n_estimators=500,
    #    random_state=42,
    #    n_jobs=-1,
    #    scale_pos_weight=1 # entspricht grob class_weight="balanced"
    #)

```

```

# LGBMC
print(f"Trainiere LightGBM auf {len(X)} Gebäuden ...")
clf = LGBMClassifier(
    n_estimators=150,
    random_state=42,
    n_jobs=-1,
    class_weight="balanced"
)

#ALTERNATIVE MULTI-LAYER_PERCEPTRON(28%)
#print(f"Trainiere MLP auf {len(X)} Gebäuden ...")
#clf = MLPClassifier(
#    hidden_layer_sizes=(100, 50),
#    random_state=42,
#    max_iter=5000)

clf.fit(X, y_encoded)

# Metadaten zurückgeben
metadata = {
    "label_mapping": {int(i): label for i, label in enumerate(label_encoder.classes_)},
    "features": list(X.columns),
    "metrics": {"info": "Evaluation erfolgt in validate_model.py"},
}

return clf, label_encoder, metadata

def save_model(model, path: str):
    """Speichert ein trainiertes Modell mit joblib."""
    joblib.dump(model, path)

def load_model(path: str):
    """Lädt ein trainiertes Modell."""
    return joblib.load(path)

def prepare_features(gdf, label_col="INSPIRE"):
    """Bereitet Features für Training oder Prediction vor."""
    drop_cols = [label_col, "geometry"]
    if "id" in gdf.columns:
        drop_cols.append("id")

    X = gdf.drop(columns=drop_cols, errors="ignore")

    # Numerische Spalten
    X_num = X.select_dtypes(include=["number"])

    # Nicht-numerische Spalten in Codes umwandeln
    X_cat = X.select_dtypes(include=["object", "category"]).copy()
    for col in X_cat.columns:
        X_cat[col] = pd.Categorical(X_cat[col]).codes

    # Zusammenführen
    X_processed = pd.concat([X_num, X_cat], axis=1)
    return X_processed

```

```

##THESIS.trainingFINAL
import geopandas as gpd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier, ExtraTreesClassifier, BaggingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier
import joblib
import pandas as pd
from typing import Tuple, Dict
import os

def train_ALL(gdf: gpd.GeoDataFrame, label_col: str = "INSPIRE") -> Tuple[RandomForestClassifier, LabelEncoder, Dict]:
    """Trainiert ein ML-Modell auf gelabelten Gebäudedaten."""

    # Features vorbereiten
    X = prepare_features(gdf, label_col=label_col)

    # Labels encoden
    y = gdf[label_col]
    label_encoder = LabelEncoder()
    y_encoded = label_encoder.fit_transform(y)

    # Modell trainieren
    ##RandomForest
    print(f"Trainiere RandomForest auf {len(X)} Gebäuden ...")
    clf3 = RandomForestClassifier(
        n_estimators=200,
        random_state=42,
        n_jobs=-1,
        class_weight="balanced"
    )

    ##XGBClassifier
    print(f"Trainiere XGBoost auf {len(X)} Gebäuden ...")
    clf2 = XGBClassifier(
        n_estimators=500,
        random_state=42,
        n_jobs=-1,
        scale_pos_weight=1 # entspricht grob class_weight="balanced"
    )

    ##BAGGING
    print(f"Trainiere Bagging auf {len(X)} Gebäuden ...")
    clf1 = BaggingClassifier(
        n_estimators=150,
        random_state=42,
        n_jobs=-1,
    )

    clf1.fit(X, y_encoded)
    clf2.fit(X, y_encoded)
    clf3.fit(X, y_encoded)

    # Metadaten zurückgeben
    metadata = {
        "label_mapping": {int(i): label for i, label in enumerate(label_encoder.classes_)},
        "features": list(X.columns),
    }

```

```

    "metrics": {"info": "Evaluation erfolgt in validate_model.py"},
}

return (clf1,
        clf2,
        clf3,
        label_encoder, metadata)

def save_model(model, path: str):
    """Speichert ein trainiertes Modell mit joblib."""
    joblib.dump(model, path)

def load_model(path: str):
    """Lädt ein trainiertes Modell."""
    return joblib.load(path)

def prepare_features(gdf, label_col="INSPIRE"):
    """Bereitet Features für Training oder Prediction vor."""
    drop_cols = [label_col, "geometry"]
    if "id" in gdf.columns:
        drop_cols.append("id")

    X = gdf.drop(columns=drop_cols, errors="ignore")

    # Numerische Spalten
    X_num = X.select_dtypes(include=["number"])

    # Nicht-numerische Spalten in Codes umwandeln
    X_cat = X.select_dtypes(include=["object", "category"]).copy()
    for col in X_cat.columns:
        X_cat[col] = pd.Categorical(X_cat[col]).codes

    # Zusammenführen
    X_processed = pd.concat([X_num, X_cat], axis=1)
    return X_processed

```

```

##THESIS.evaluation
from sklearn.metrics import classification_report, confusion_matrix
from thesis.training import prepare_features
import numpy as np

def evaluate_model(gdf, model, metadata, label_col="INSPIRE"):
    """Bewertet ein trainiertes Modell anhand von Validierungsdaten."""

    features_train = metadata["features"]
    label_mapping = metadata["label_mapping"]

    # Mapping der Labels
    inv_label_mapping = {v: int(k) for k, v in label_mapping.items()}

    # Features vorbereiten + Reindex auf Trainingsfeatures
    X = prepare_features(gdf, label_col=label_col) # numerisch + kategorisch codieren

    # Labels encodieren
    y_true = gdf[label_col].map(inv_label_mapping)

    print("Spalten Y:", X.columns.tolist())

    # Vorhersage
    y_pred = model.predict(X)

    #Wahrscheinlichkeit für die vorhergesagte Klasse
    probas = model.predict_proba(X)
    max_probas = probas.max(axis=1) #höchste Warhscheinlichkeit pro Feature

    # Tabelle erstellen
    import pandas as pd
    result_df = pd.DataFrame({
        'Tatsächliche Klasse': y_true,
        'Vorhergesagte Klasse': y_pred,
        'Sicherheit': max_probas
    })
    print(result_df.head(200))
    # Optional: Tabelle als CSV speichern
    result_df.to_csv("P:/Elias/Masterarbeit/Praxisteil/results/MULTIPLEmodels.csv", index=False)
    # Metriken
    report = classification_report(y_true, y_pred, output_dict=True)
    cm = confusion_matrix(y_true, y_pred, labels=list(inv_label_mapping.values()))

    ### optional
    results = {
        "metrics": {
            "accuracy": report["accuracy"],
            "macro_avg": report["macro avg"],
            "weighted_avg": report["weighted avg"],
        },
        "class_report": report,
        "confusion_matrix": cm.tolist(),
        "n_samples": len(gdf)
    }

    print(f"🇩🇪 Evaluation abgeschlossen: {len(gdf)} Samples")
    print(f"Accuracy: {report['accuracy']:.3f}")

    ##Variante für Cross-Validation
    results = report["accuracy"]
    return results

```

```
##THESIS.prediction
def predict_all(gdf, model, metadata):
    """Wendet ein trainiertes Modell auf alle Gebäude an."""

    features = metadata["features"]
    label_mapping = metadata["label_mapping"]

    X = gdf[features]
    y_pred = model.predict(X)

    inv_label_mapping = {int(k): v for k, v in label_mapping.items()}
    gdf["predicted_label"] = [inv_label_mapping[i] for i in y_pred]

    return gdf
```

```

##SCRIPTS.enrich_data
import os
import geopandas as gpd
from thesis.data_processing import load_buildings, enrich_buildings, filter_labeled, split_train_validation, remove_attributes
from thesis.load_ortho import calc_imagestats

def main():
    # Eingabedatei
    input_path = "P:/Elias/Masterarbeit/Praxisteil/Data/FULL2.shp"
    # Ausgabedatei (angereichert)
    output_full = "P:/Elias/Masterarbeit/Praxisteil/Data/FULL3.shp"
    # Trainingsdatensatz (nur gelabelte Gebäude)
    output_train="P:/Elias/Masterarbeit/Praxisteil/Data/TRAIN3.shp"
    # Validierte Daten
    output_val = "P:/Elias/Masterarbeit/Praxisteil/Data/VAL3.shp"
    #Daten für Cross-Validation
    output_crossval = "P:/Elias/Masterarbeit/Praxisteil/Data/CROSSVAL3.shp"
    # Orthofoto
    ortho = "W:/kart/KI/Exporte/DOP_A3_800_geo.tif"

    print(f"[INFO] Lade Shapefile: {input_path}")
    gdf = load_buildings(input_path)

    print("[INFO] Reichere Gebäude mit Attributen an ...")
    gdf_enriched = enrich_buildings(gdf)

    #print("[INFO] Orthofot zuschneiden und Bildstatistiken berechnen ...")
    gdf_enriched = calc_imagestats(gdf_enriched,ortho)

    print(f"[INFO] Speichere Ergebnis nach: {output_full}")
    gdf_enriched.to_file(output_full)

    print(f"[INFO] Speichere gelabelte Daten für Cross-Validation nach: {output_crossval}")
    gdf_labeled = filter_labeled(gdf_enriched)
    gdf_labeled.to_file(output_crossval)

    print(f"[INFO] Trainings- und Validierungsdaten speichern ...")
    train_gdf, val_gdf = split_train_validation(gdf_labeled)
    train_gdf.to_file(output_train)
    val_gdf.to_file(output_val)
    print("Spalten nach SAVE:", gdf_labeled.columns.tolist())

    print("[INFO] Fertig!")

if __name__ == "__main__":
    main()

```

```

##SCRIPTS.train_model
import os
import joblib
import geopandas as gpd
from thesis.training import train_model, save_model
from thesis.data_processing import remove_attributes

def main():
    input_path = "P:/Elias/Masterarbeit/Praxisteil/Data/TRAIN2.shp"
    model_path = "P:/Elias/Masterarbeit/Praxisteil/Models/building_classifier.pkl"
    metadata_path = "P:/Elias/Masterarbeit/Praxisteil/Models/model_metadata.json"
    model_dir = "P:/Elias/Masterarbeit/Praxisteil/Models"
    os.makedirs(model_dir, exist_ok=True)
    encoder_path = os.path.join(model_dir, "label_encoder.pkl")
    features_path = os.path.join(model_dir, "feature_names.pkl")

    print(f"[INFO] Lade gelabelte Daten: {input_path}")
    gdf_train = gpd.read_file(input_path)

    #!!!!!!!!!!!!nur fürs Attribute finden!!!!!!!!!!!!
    # Liste mit zu entfernenden Attributen erstellen
    att_remove = ["POI_1",
                  #"POI_2",
                  #"POI_3",
                  #"POI_4",
                  #"POI_5",
                  #"POI_6",
                  #"NUTZUNG",
                  #"GEBUNTERSC",
                  #"HAUSNUMMER",
                  #"STRASSE",
                  #"SKZ",
                  #"PLZ",
                  #"centroid_x",
                  #"centroid_y",
                  #"perimeter",
                  #"area",
                  #"h",
                  #"GDAEIGENSC",
                  #"WOHNZWECKE",
                  #"ZU_ADRID",
                  #"band1_dark",
                  #"GIKID",
                  #"ADRHAUPTGB",
                  #"ADRID",
                  #"band3_brig",
                  #"band1_min",
                  #"band2_dark",
                  #"band2_min",
                  #"band2_brig",
                  #"band3_dark",
                  #"band3_min",
                  #"ADRCD",
                  #"band1_brig"
                  ]
    gdf_train = remove_attributes(gdf_train, att_remove)

    print("[INFO] Trainiere Modell ...")
    clf, label_encoder, meta = train_model(gdf_train, label_col="INSPIRE")

    print("[INFO] Speichere Modell ...")

```

```
# Modell speichern
save_model(clf, model_path)
save_model(label_encoder, encoder_path)
joblib.dump(meta["features"], features_path)

print("[INFO] Training abgeschlossen.")
print(f"[INFO] Modell gespeichert unter: {model_path}")
print("[INFO] Wichtigste Metriken:")

if __name__ == "__main__":
    main()
```

```

##SCRIPTS.multiple_models
from thesis.data_processing import load_buildings, cross_validate_gdf, remove_attributes
from thesis.training import prepare_features
from thesis.trainingALL import train_ALL
import numpy as np
from sklearn.metrics import f1_score, confusion_matrix
import pandas as pd

#Daten für Cross-Validation
input_path = "P:/Elias/Masterarbeit/Praxisteil/Data/CROSSVAL.shp"

print(f"[INFO] Lade Shapefile: {input_path}")
gdf = load_buildings(input_path)

# !!!!!!!!!!!!!nur fürs Attribute finden!!!!!!!!!!!!!!
# Liste mit zu entfernenden Attributen erstellen
att_remove = [
    #"POI_1",
    #"POI_2",
    #"POI_3",
    #"POI_4",
    #"POI_5",
    #"POI_6",
    #"NUTZUNG",
    #"GEBUNTERSC",
    #"HAUSNUMMER",
    #"STRASSE",
    #"SKZ",
    #"PLZ",
    #"centroid_x",
    #"centroid_y",
    #"perimeter",
    #"area",
    #"h",
    #"GDAEIGENSC",
    #"WOHNZWECKE",
    #"ZU_ADRID",
    #"band1_dark",
    #"GIKID",
    #"ADRHAUPTGB",
    #"ADRID",
    #"band3_brig",
    #"band1_meann",
    #"band2_dark",
    #"band2_mean",
    #"band2_brig",
    #"band3_dark",
    #"band3_mean",
    #"ADRCD",
    #"band1_brig"
]
gdf = remove_attributes(gdf, att_remove)

folds = cross_validate_gdf(gdf, n_splits=5)
results = []
f1_scores = []
f1_scores_per_class = []
all_y_true = []
all_y_pred = []

for i, (train_gdf, val_gdf) in enumerate(folds):

```

```

# Beispiel: Modelltraining
## Trainiere Modell
(clf1, clf2,
 clf3,
 label_encoder, metadata) = train_ALL(train_gdf, label_col="INSPIRE")
label_col = "INSPIRE"
label_mapping = metadata["label_mapping"]

# Mapping der Labels
inv_label_mapping = {v: int(k) for k, v in label_mapping.items()}

# Features vorbereiten + Reindex auf Trainingsfeatures
X = prepare_features(val_gdf, label_col=label_col) # numerisch + kategorisch codieren

# Labels encodieren
y_true = val_gdf[label_col].map(inv_label_mapping)

## print("Spalten Y:", X.columns.tolist())

# Vorhersage
y_pred1 = clf1.predict(X)
y_pred2 = clf2.predict(X)
y_pred3 = clf3.predict(X)

# Wahrscheinlichkeit für die vorhergesagte Klasse
probas1 = clf1.predict_proba(X)
probas2 = clf2.predict_proba(X)
probas3 = clf3.predict_proba(X)

max_probas1 = probas1.max(axis=1) # höchste Wahrscheinlichkeit pro Feature
max_probas2 = probas2.max(axis=1) # höchste Wahrscheinlichkeit pro Feature
max_probas3 = probas3.max(axis=1) # höchste Wahrscheinlichkeit pro Feature

# Zuerst alle Vorhersagen und Sicherheiten in Arrays packen
vorhersagen = np.column_stack([y_pred1, y_pred2,
                               y_pred3,
                               ])
sicherheiten = np.column_stack([max_probas1, max_probas2,
                                max_probas3,
                                ])

# Index der höchsten Sicherheit je Zeile bestimmen
max_indices = np.argmax(sicherheiten, axis=1)
# Die entsprechende Vorhersageklasse je Zeile auswählen
beste_vorhersage = [row[idx] for row, idx in zip(vorhersagen, max_indices)]
# Vergleich mit tatsächlicher Klasse
correct = beste_vorhersage == y_true
# Prozentueller Anteil
percent_correct = np.mean(correct)
results.append(percent_correct)
print(percent_correct)

f1_per_class = f1_score(y_true, beste_vorhersage, average=None)
f1_scores_per_class.append(f1_per_class)

f1 = f1_score(y_true, beste_vorhersage, average="macro")
f1_scores.append(f1)

all_y_true.extend(y_true)
all_y_pred.extend(beste_vorhersage)

# Tabelle erstellen

```

```

import pandas as pd

result_df = pd.DataFrame({
    'Tatsächliche Klasse': y_true,
    'Vorhergesagte Klasse 1': y_pred1,
    'Sicherheit 1': max_prob1,
    'Vorhergesagte Klasse 2': y_pred2,
    'Sicherheit 2': max_prob2,
    'Vorhergesagte Klasse 3': y_pred3,
    'Sicherheit 3': max_prob3,
    'Beste Vorhersage': beste_vorhersage,
})
print(result_df)
result_df.to_csv("P:/Elias/Masterarbeit/Praxisteil/results/MULTIPLEmodels.csv", index=False)

results = np.array(results)
average = results.mean()
std = results.std()
maximum = results.max()
minimum = results.min()
print(f"Durchschnitt: {average}")
print(f"Std: {std}")
print(f"Maximum: {maximum}")
print(f"Minimum: {minimum}")

# F1-Score Statistik
# F1-Scores pro Klasse über alle Folds als Array
f1_scores_per_class = np.array(f1_scores_per_class) # shape: (n_folds, n_classes)
print("Statistik pro Klasse über alle Folds:")
for idx, class_name in enumerate(label_encoder.classes_):
    scores = f1_scores_per_class[:, idx]
    print(f"Klasse '{class_name}':")
    print(f"    Mittelwert: {np.mean(scores):.4f}")

print(f"Mean F1-Score: {np.mean(f1_scores):.4f}")
print(f"Standardabweichung F1: {np.std(f1_scores):.4f}")
print(f"Max F1: {np.max(f1_scores):.4f}")
print(f"Min F1: {np.min(f1_scores):.4f}")

#Confusion Matrix
cm = confusion_matrix(all_y_true, all_y_pred)
# Optional: Als DataFrame für bessere Übersicht
class_names = label_encoder.classes_
cm_df = pd.DataFrame(cm, index=class_names, columns=class_names)
print("Confusion Matrix:")
print(cm_df)
cm_df.to_csv("P:/Elias/Masterarbeit/Praxisteil/results/confusion_matrix.csv")

```

```

##SCRIPTS.feature_importance
import geopandas as gpd
import pandas as pd
import json
from thesis.evaluation import evaluate_model
from thesis.training import load_model, prepare_features
from thesis.data_processing import remove_attributes

def main():
    model_path = "P:/Elias/Masterarbeit/Praxisteil/Models/building_classifier.pkl"
    metadata_path = "P:/Elias/Masterarbeit/Praxisteil/models/model_metadata.json"
    validation_path = "P:/Elias/Masterarbeit/Praxisteil/Data/VAL.shp"
    report_path = "P:/Elias/Masterarbeit/Praxisteil/results/validation_report.json"

    print(f"[INFO] Lade Validierungsdaten: {validation_path}")
    gdf_val = gpd.read_file(validation_path)
    # !!!!!!!!!!!!!!!nur fürs Attribute finden!!!!!!!!!!!!!!!!!!!!!!
    # Liste mit zu entfernenden Attributen erstellen
    att_remove = ["POI_1",
                  "POI_2",
                  "POI_3",
                  "POI_4",
                  "POI_5",
                  "POI_6",
                  "NUTZUNG",
                  "GEBUNTERSC",
                  "HAUSNUMMER",
                  "STRASSE",
                  "SKZ",
                  "PLZ",
                  "centroid_x",
                  "centroid_y",
                  "perimeter",
                  "area",
                  "h",
                  "GDAEIGENSC",
                  "WOHNZWECKE",
                  "ZU_ADRID",
                  "band1_dark",
                  "GIKID",
                  "ADRHAUPTGB",
                  "ADRID",
                  "band3_brig",
                  "band1_min",
                  "band2_dark",
                  "band2_min",
                  "band2_brig",
                  "band3_dark",
                  "band3_min",
                  "ADRCD",
                  "band1_brig"
                 ]
    gdf_val = remove_attributes(gdf_val, att_remove)

    print("[INFO] Lade Modell und Metadaten ...")
    model = load_model(model_path)
    with open(metadata_path, "r", encoding="utf-8") as f:
        metadata = json.load(f)
        metadata["label_mapping"] = {int(k): v for k, v in metadata["label_mapping"].items()}

```

```

# Vorhersage & Evaluation

print("[INFO] Starte Modellvalidierung ...")
results = evaluate_model(gdf_val, model, metadata, label_col="INSPIRE")

print(f"[INFO] Speichere Ergebnisse nach: {report_path}")
with open(report_path, "w", encoding="utf-8") as f:
    json.dump(results, f, indent=2)

print(f"[INFO] Accuracy: {results['metrics']['accuracy']:.3f}")
print("[INFO] Validierung abgeschlossen!")

# Feature Importance berechnen
feature_importance = pd.DataFrame({
    "Feature": model.feature_names_in_,
    "Importance": model.feature_importances_
}).sort_values(by="Importance", ascending=False)

print("\n 🇮🇹 Wichtigste Features:")
print(feature_importance.head(40))

# Optional speichern
#feature_importance.to_csv("P:/Elias/Masterarbeit/Praxisteil/results/feature_importance.csv", index=False)
#print("[INFO] Feature-Importance gespeichert!")

if __name__ == "__main__":
    main()

```

```

##SCRIPTS.cross_validation
from thesis.data_processing import cross_validate_gdf
from thesis.data_processing import load_buildings, remove_attributes
from thesis.training import train_model, prepare_features
from thesis.evaluation import evaluate_model
import numpy as np
from sklearn.metrics import f1_score
import pandas as pd

#Daten für Cross-Validation
input_path = "P:/Elias/Masterarbeit/Praxisteil/Data/CROSSVAL2.shp"

print(f"[INFO] Lade Shapefile: {input_path}")
gdf = load_buildings(input_path)

# !!!!!!!!!!!!!!!nur fürs Attribute finden!!!!!!!!!!!!!!!!!!!!
# Liste mit zu entfernenden Attributen erstellen
att_remove = [
    "POI_1",
    "POI_2",
    "POI_3",
    "POI_4",
    "POI_5",
    "POI_6",
    "NUTZUNG",
    "GEBUNTERSC",
    "HAUSNUMMER",
    "STRASSE",
    "SKZ",
    "PLZ",
    # "centroid_x",
    # "centroid_y",
    # "perimeter",
    # "area",
    # "h",
    # "GDAEIGENSC",
    "WOHNZWECKE",
    # "ZU_ADRID",
    "band1_dark",
    # "GIKID",
    "ADRHAUPTGB",
    # "ADRID",
    "band3_brig",
    "band1_mean",
    "band2_dark",
    "band2_mean",
    "band2_brig",
    "band3_dark",
    # "band3_mean",
    "ADRCD",
    "band1_brig",
    "band1_max",
    "band2_max",
    "band3_max",
    "band1_std",
    "band2_std",
    "band3_std"
]

gdf_train = remove_attributes(gdf, att_remove)

folds = cross_validate_gdf(gdf, n_splits=5)

```

```

results = []
f1_scores = []
f1_scores_per_class = []
all_importances = []

for i, (train_gdf, val_gdf) in enumerate(folds):
    # Beispiel: Modelltraining
    clf, label_encoder, meta = train_model(train_gdf, label_col="INSPIRE") # Deine Trainingsfunktion
    score = evaluate_model(val_gdf, clf, meta, label_col="INSPIRE") # Deine Bewertungsfunktion
    results.append(score)
    print(f"Fold {i+1}: Validation Score = {score}")

    X_val = prepare_features(val_gdf, label_col="INSPIRE")
    y_val = val_gdf["INSPIRE"]
    y_val_encoded = label_encoder.transform(y_val)
    y_pred = clf.predict(X_val)

    # Feature Importance berechnen
    feature_importance = pd.DataFrame({
        "Feature": clf.feature_names_in_,
        "Importance": clf.feature_importances_
    }).sort_values(by="Importance", ascending=False)
    all_importances.append(feature_importance.set_index("Feature")["Importance"])

    f1_per_class = f1_score(y_val_encoded, y_pred, average=None)
    f1_scores_per_class.append(f1_per_class)

    f1 = f1_score(y_val_encoded, y_pred, average="macro")
    f1_scores.append(f1)
    print(f"Fold {i + 1}: Validation Score = {score}, F1-Score = {f1:.4f}")
# Durchschnittliche Performance über alle Folds
print(f"Mean Validation Score: {np.mean(results)}")
#Standardabweichung
print(f"Standardabweichung: {np.std(results)}")
#Min und Max
print(f"Max: {np.max(results)}")
print(f"Min: {np.min(results)}")

# F1-Score Statistik
# F1-Scores pro Klasse über alle Folds als Array
f1_scores_per_class = np.array(f1_scores_per_class) # shape: (n_folds, n_classes)
print("Statistik pro Klasse über alle Folds:")
for idx, class_name in enumerate(label_encoder.classes_):
    scores = f1_scores_per_class[:, idx]
    print(f"Klasse '{class_name}':")
    print(f"    Mittelwert: {np.mean(scores):.4f}")

print(f"Mean F1-Score: {np.mean(f1_scores):.4f}")
print(f"Standardabweichung F1: {np.std(f1_scores):.4f}")
print(f"Max F1: {np.max(f1_scores):.4f}")
print(f"Min F1: {np.min(f1_scores):.4f}")

#Feature-Importance
mean_importance = pd.concat(all_importances, axis=1).mean(axis=1).sort_values(ascending=False)
print(mean_importance)

```

```

##SCRIPTS.predict_all
import geopandas as gpd
import json
from thesis.training import load_model
from thesis.prediction import predict_all

def main():
    input_path = "data/processed/buildings_enriched.shp"
    output_path = "data/results/buildings_classified.shp"
    model_path = "data/models/building_classifier.pkl"
    metadata_path = "data/models/model_metadata.json"

    print(f"[INFO] Lade Gebäudedaten: {input_path}")
    gdf = gpd.read_file(input_path)

    print("[INFO] Lade Modell und Metadaten ...")
    model = load_model(model_path)
    with open(metadata_path, "r", encoding="utf-8") as f:
        metadata = json.load(f)

    print("[INFO] Starte Klassifikation ...")
    gdf_pred = predict_all(gdf, model, metadata)

    print(f"[INFO] Speichere Ergebnisse nach: {output_path}")
    gdf_pred.to_file(output_path)

    print(f"[INFO] {len(gdf_pred)} Gebäude klassifiziert.")
    print("[INFO] Fertig!")

if __name__ == "__main__":
    main()

```