

Klassifikation von Netzwerktopologien mithilfe von Deep Learning in Bezug auf IIoT

Andreas Buss, Henrik Oppenländer, Jannis Lang, Javier Villalba-Diez

Zusammenfassung: Das schnelle Wachstum des Datenvolumens in IIoT-Netzwerken und der damit verbundenen komplexen Beziehungen, wie Maschineninteraktionen und Prozessautomatisierungen, wachsen zu immer größeren Herausforderungen heran. Sie bieten aber auch große Potenziale als Informationsträger, doch die manuelle Bearbeitung dieser umfangreichen Datensätze ist zu zeitaufwendig und anfällig für subjektive Fehler. Es werden effizientere Methoden zur Analyse und Unterscheidung verschiedener Netzwerktopologien benötigt. Das Hauptziel der Studie ist es, die Machbarkeit der Klassifizierung von Netzwerktopologien mit Hilfe von Deep Learning zu demonstrieren. Dafür werden aus zufällig generierten Netzwerken, Laplacian-Matrizen gebildet, welche für das Training des Deep Learning Modells benutzt werden. Die Ergebnisse zeigen, dass solche KI-Modelle mit einer hohen Genauigkeit effektiv Netzwerkstrukturen identifizieren können, was vielversprechend für eine Bandbreite von IIoT-Anwendungsfällen wie Supply Chain Management, Predictive Maintenance, Quality Control und weitere ist. Damit können die Effizienz und Qualität der Betriebsabläufe verbessert, sowie der Warenfluss optimiert werden und zur Entwicklung neuer Geschäftsmodelle beitragen.

Schlüsselwörter: IIoT, Netzwerkanalyse, Graph-Klassifikation, Neural Networks, Deep Learning

1 Einleitung

Die Analyse und Klassifizierung von Daten in einer Netzwerkstruktur wird im Bereich des Industrial Internet of Things (IIoT) immer bedeutender (vgl. Elhaloui et al., 2023). Mit dem rasanten Wachstum des Datenvolumens in IIoT-Netzwerken, das komplexe Beziehungen wie Maschineninteraktionen, Prozessautomatisierungen und Systemvernetzungen widerspiegelt, steigt die Notwendigkeit effizienter Methoden zur Analyse und Unterscheidung verschiedener Netzwerktopologien (vgl. gl. Borges et al., 2022). Die manuelle Bearbeitung dieser umfangreichen Datensätze im IIoT-Kontext ist zeitaufwändig und anfällig für Fehler (vgl. Martha Rodríguez et al., 2023). Hier kann künstliche Intelligenz (KI) einen wesentlichen Beitrag leisten, indem sie automatisierte, datengesteuerte Analyseprozesse ermöglicht. Um diesem Problem zu begegnen, widmen wir uns der Beantwortung folgender Forschungsfrage: „Wie können wir mit Hilfe von KI die Klassifizierung von Daten in einer Netzwerkstruktur ermöglichen?“. Das Hauptziel dieses Projekts besteht darin, die Machbarkeit der Klassifizierung von Netzwerktopologien mithilfe von KI zu untersuchen. Es soll anhand unterschiedlicher, zufällig generierter Netzwerkdatensätze die Leistungsfähigkeit von Deep Learning Modellen festgestellt werden. Als Trainingsbasis dient dafür eine Laplacian-Matrix, die aus den jeweiligen zufällig generierten Graphen hervor geht. Die erfolgreiche Umsetzung dieser Zielsetzungen trägt dazu bei, ein besseres Verständnis dafür zu entwickeln, wie KI strukturelle Unterschiede in Netzwerken erkennen kann, um damit Potenziale für Anwendungen in verschiedenen Bereichen wie Supply Chain Management, Predictive Maintenance, Quality Control und weiteren zu identifizieren. In Abbildung 1 wird ein Graphical Abstract gezeigt, welcher die komplexen Informationen zu dieser Forschungsarbeit zusätzlich visuell und kompakt zusammenfasst, um das Verständnis und die Zugänglichkeit des Forschungsgegenstandes zu verbessern.

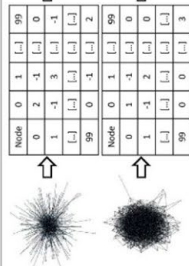
KLASSIFIKATION VON NETZWERKTOPOLOGIEN MIT HILFE VON DEEP LEARNING IM BEZUG AUF IIOT

1. Zielsetzung

Das Ziel dieses Papers ist die Klassifizierung von zwei Graphen anhand ihrer Unterschiedlichen Netzwerktopologien mit Deep Learning. Das erste Netzwerk ist ein Free-Scale Netzwerk, das zweite Netzwerk ist ein Erdős-Rényi-Netzwerk. Durch diesen Ansatz werden unterschiedliche Netzwerke dargestellt, wie sie im IIOT-Bereich durch verschiedene Akteure, beispielsweise Menschen, Autos und Sensoren generiert werden. Die praktische Anwendung dieser Arbeit ist dabei vielseitig und kann von der frühzeitigen Erkennung von Problemen in industriellen Anlagen, über die Optimierung von Produktionsprozessen bis hin zur Verbesserung der IT-Infrastruktur und -sicherheit führen.

3. Darstellung der Graphen

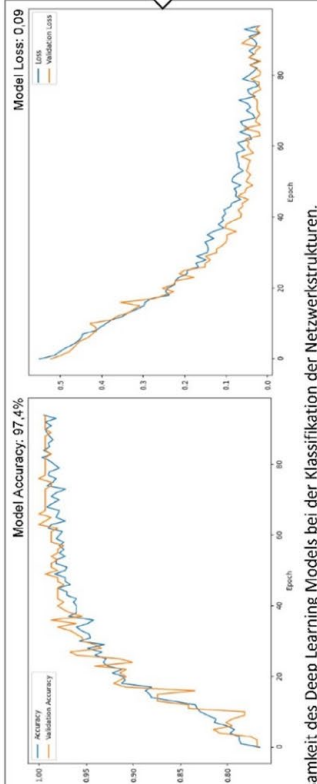
Die gesamten Graphen Informationen der beiden zufällig generierten Graphen werden durch Laplacian-Matrizen dargestellt. Auf den Kanten der Graphen werden anschließend Vektoren mit 10 Elementen generiert. Die daraus entstandenen Würfel sind die Grundlage für das Training des Machine-Learning Modells.



5. Ergebnisse

Der linke Graph zeigt die Accuracy des Modells (97.4%), die konsistent steigt, bis keine Verbesserung des Modells mehr stattfindet und eine Early-Stopping-Funktion greift.

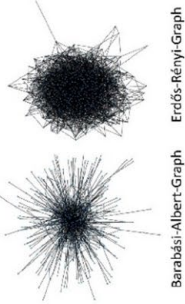
Der rechte Graph zeigt den Loss des Modells (0.09), mit dem eine Aussage über die Genauigkeit des Modells getroffen werden kann.



→ Beide Ergebnisse zeigen die gute Wirksamkeit des Deep Learning Modells bei der Klassifikation der Netzwerkstrukturen.

2. Generierung der Graphen

Es werden zwei Graphen mittels eines Erdős-Rényi- und eines Barabási-Albert-Algorithmus zufällig generiert. Beide haben jeweils 100 Knoten.



4. Training und Testing des Deep Learning Models

Für das Modell verwenden wir ein Feedforward-Neuronales Netzwerk (FNN) mit 4 Layern. Es eignet sich gut, um Daten zu verarbeiten, die in Form von Graphen vorliegen. Es kann nicht nur Merkmale einzelner Knoten berücksichtigen, sondern auch deren Verbindungen im Netzwerk. Dadurch können Muster und Anomalien im gesamten Netzwerks erkannt werden. Besonders im IIOT-Umfeld ist das wichtig. Der Input Layer arbeitet mit 256 Knoten, um eine große Merkmalsverarbeitung der komplexen IIOT-Daten sicher zu stellen. In den ersten drei Schichten wird LeakyReLU als Aktivierungsfunktion verwendet, da sie die Flexibilität von ReLU bietet und tote Neuronen vermieden werden. Im Output Layer wird Sigmoid verwendet, da sie sich gut für die binäre Klassifizierung eignet.

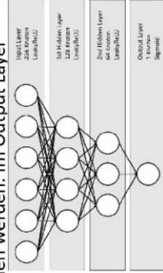


Abbildung 1: Graphical Abstract (eigene Darstellung)

2 Theoretischer Hintergrund

Industrial Internet of Things

Das Industrial Internet of Things (IIoT) erweitert das Internet of Things (IoT) auf industrielle Anwendungen, indem es Maschinen und Anlagen in vernetzte Systeme integriert. Diese Vernetzung ermöglicht die Optimierung und Automatisierung industrieller Prozesse (vgl. Petrik, 2022). Im Zentrum des IIoT stehen verschiedene Kernkonzepte. Erstens sind da die vernetzten Geräte: Maschinen und Sensoren, die miteinander vernetzt sind, ermöglichen den Datenaustausch und die Fernsteuerung. Dies ist ein grundlegender Aspekt für die Effizienzsteigerung in der Industrie (vgl. Oberländer et al., 2021). Zweitens spielt die Datenanalyse eine wesentliche Rolle. Mittels KI und maschinellem Lernen werden die generierten Daten analysiert, um zu gewinnen (vgl. Schmihing & Jochem, 2021). Drittens ist die Automatisierung ein entscheidender Faktor: IIoT-Systeme nutzen Daten in Echtzeit, um wertvolle Erkenntnisse für die Prozessoptimierung und vorbeugende Wartung einer Netzwerkstruktur, um Prozesse weiter zu optimieren und die Effizienz zu steigern (vgl. Schmihing & Hochem, 2021). KI ermöglicht es, Muster und Anomalien zu erkennen und unterstützt automatisierte Entscheidungsprozesse (vgl. Gauger et al., 2022). Dies führt zu einer gesteigerten Effizienz und Kostensenkung. Anwendung findet das IIoT unter anderem in der vorausschauenden Wartung, der Echtzeit-Überwachung von Produktionsprozessen und in intelligenten Versorgungsketten (vgl. Petrik, 2022).

Neuronale Netze

Neuronale Netze sind ein zentrales Konzept in der KI und bilden die Basis für viele fortschrittliche Anwendungen, einschließlich der Klassifizierung von Daten in einer Netzwerkstruktur. Sie sind inspiriert von den biologischen Prozessen im menschlichen Gehirn und bestehen aus einer Vielzahl von miteinander verbundenen Einheiten, den sogenannten Neuronen (Lanquillon & Schacht, 2023, S. 72). Die Struktur neuronaler Netze umfasst typischerweise drei Hauptkomponenten: Eingabeschicht, versteckte Schichten und Ausgabeschicht (vgl. Kreßel et al., 1991). Neuronen in diesen Schichten sind über Gewichte mit Neuronen in der nächsten Schicht verbunden. Die Eingabeschicht empfängt die Daten, während die versteckten Schichten die Datenverarbeitung übernehmen und die Ausgabeschicht das Endergebnis liefert (vgl. Kreßel et al., 1991). Neuronale Netze lernen durch einen Prozess, der als „Training“ bezeichnet wird. Dabei werden ihnen Beispieldaten zusammen mit den gewünschten Ergebnissen präsentiert. Das Netz passt seine Gewichte durch Algorithmen wie den Backpropagation-Algorithmus an, um die Differenz zwischen dem tatsächlichen Ausgang des Netzes und dem gewünschten Ausgang zu minimieren (vgl. Poddig, 2000). Eine weitere Schlüsselkomponente in neuronalen Netzen sind die Aktivierungsfunktionen. Sie bestimmen, wie die gewichteten Eingaben eines Neurons verarbeitet und an die nächste Schicht weitergegeben werden (vgl. Lanquillon & Schacht, 2023, S. 79). Bekannte Aktivierungsfunktionen sind die Sigmoid-, Tanh- und ReLU-Funktionen (vgl. Fuhrmann, 2019). Man kann neuronale Netze außerdem in verschiedene Arten unterscheiden, darunter konventionelle Feedforward-Netze, die für sequenzielle Daten geeignet sind, und Convolutional Neural Networks (CNNs), die häufig in der Bild- und Videoverarbeitung eingesetzt werden (vgl. Namvar et al., 2021). Convolutional Neural Networks sind gut geeignet zum Erlernen und Extrahieren von Merkmalen aus Bilddaten (vgl. Chen et al., 2023). Darüber hinaus kann das Modell aber auch zur Klassifikation von Daten in einer Netzwerkstruktur verwendet werden. Das Modell besteht aus mehreren Schichten und funktioniert dabei so, dass das neuronale Netzwerk die hierarchische Struktur beibehält, indem es interne Merkmalsdarstellungen lernt und die Merkmale verallgemeinert (vgl. Manaswi, 2018). Die Einsatzgebiete von Feed Forward Neural Networks sind sehr vielfältig. Der Aufbau des Modells ist jedoch immer gleich. Die Verbindungen zwischen den Neuronen gehen immer nur in eine Richtung und sind einseitig. Die Neuronen sind dabei angeordnet in parallelen Schichten, wobei die erste Schicht die Eingabeschicht und die letzte Schicht die Ausgabeschicht ist. Alle weiteren Schichten dazwischen werden als verborgene Schichten bezeichnet (vgl. Mirjalili, 2019). Ein Graph Neural Network ist ein neuronales Netzwerk, welches Graphdaten als Eingabe verwendet (vgl. Bishop & Bishop, 2024). Die Graphdaten werden dann

in Zwischeneinbettungen umgewandelt und diese Einbettungen einer letzten Ebene zugeführt, die auf eine Vorhersageaufgabe ausgerichtet ist (vgl. Blumenfeld, 2023). Aufgrund der Fähigkeit von Graph Neural Networks, verschiedene Daten mit komplexen Strukturen zu modellieren, sind sie in vielen Anwendungen und Bereichen weit verbreitet, wie beispielsweise in der Verarbeitung natürlicher Sprache, in der Bioinformatik oder der Anomalieerkennung (vgl. Wu, L. et al., 2022). In manchen Bereichen wie dem stark ansteigenden Multimedia-Verkehr im Internet der Dinge haben diese Modelle jedoch immer noch Defizite (vgl. Wang et al., 2019). Daher wurde damit begonnen in diesem Bereich Klassifikationsprobleme mit Deep Learning Methoden zu lösen.

Graphen-/Netzwerktopologie

Im Rahmen der Klassifizierung von Daten in einer Netzwerkstruktur durch KI spielt das Verständnis der Graphen- und Netzwerktopologie eine wesentliche Rolle. Graphen repräsentieren die strukturelle Grundlage von Netzwerken und sind in der Netzwerktheorie eine Sammlung von Knoten (oder Ecken) und Kanten, die diese Knoten verbinden (vgl. Lanquillon & Schacht, 2023). Innerhalb der Netzwerktheorie werden verschiedene Arten von Graphen unterschieden. Zunächst gibt es ungerichtete Graphen, bei denen die Kanten bidirektional sind, was bedeutet, dass die Richtung der Verbindung zwischen den Knoten irrelevant ist (vgl. Berghammer, 2021). Beispiele für ungerichtete Graphen sind Erdős-Rényi-Graphen und Barabási-Albert-Graphen. Bei den Erdős-Rényi-Graphen handelt es sich um eine der einfachsten Arten von Zufallsgraphen (vgl. Mandjes et al., 2019). Sie werden durch zwei Parameter definiert: n , die Anzahl der Knoten im Graphen, und p , die Wahrscheinlichkeit, dass zwischen zwei Knoten eine Kante besteht (vgl. Bangachev & Bresler, 2023). Barabási-Albert-Graphen funktionieren so, dass neue Knoten die Verbindung zu bereits gut verbundenen Knoten bevorzugen. Das führt zu Netzwerken mit einer „scale-free“ Gradverteilung. Das bedeutet, es gibt einige wenige Knoten mit vielen Verbindungen und viele Knoten mit wenigen Verbindungen (vgl. Arizmendi & Dominguez, 2022). Die Gradverteilung in den Graphen folgt einem Power-Law, was bedeutet, dass die Wahrscheinlichkeit, dass ein Knoten k Verbindungen hat, proportional zu $k^{-\gamma}$ ist, wobei γ (Gamma) ein Parameter ist, der typischerweise zwischen 2 und 3 liegt (vgl. Ryabchenko & Samosvat, 2010). Dies steht im Gegensatz zu gerichteten Graphen, die gerichtete Kanten aufweisen. Diese zeigen eine bestimmte Orientierung von einem Knoten zum anderen an und sind daher für die Darstellung von asymmetrischen Beziehungen zwischen Knoten nützlich (vgl. Berghammer, 2021). Ein weiterer wichtiger Graphentyp sind die gewichteten Graphen. Hierbei haben die Kanten Gewichte, die verschiedene Eigenschaften wie die Bandbreite oder die Latenz der Verbindung repräsentieren können (vgl. Lanquillon & Schacht, 2023).

Daten in einer Netzwerkstruktur: Definition und Bedeutung

Daten in einer Netzwerkstruktur bekommen im IIoT eine besondere Bedeutung zuteil. Große Datensätze können dazu genutzt werden, gesamte Geschäftsprozesse zu transformieren, indem diese als Feedback genutzt und auf sie reagiert wird (vgl. Dagnino, 2021). Sie ermöglichen es, Einblicke in die Netzwerkleistung zu gewinnen, was für die Optimierung von Prozessen, die Sicherheit, die vorbeugende Wartung und die Unterstützung von Geschäftsentscheidungen unerlässlich ist. Das „intelligente“ Verhalten der verbundenen Objekte im Netzwerk äußert sich dadurch, dass diese die ständig anfallenden Daten auswerten, analysieren und auf diese angemessen reagieren (vgl. Boyes et al., 2018). IIoT kann somit sowohl die Produktivität als auch die Effizienz eines Geschäftsmodells erhöhen. Daten in einer Netzwerkstruktur helfen, Probleme in Maschinen und Anlagen frühzeitig zu erkennen, was wiederum Ausfallzeiten reduziert und die Effizienz steigert. Allerdings bringt der Umgang mit Daten in einer Netzwerkstruktur auch Probleme und Herausforderungen mit sich. Das große Volumen und die Komplexität der Daten erfordern effiziente Methoden für Speicherung, Verarbeitung und Analyse (vgl. Wu, Y. et al., 2022). Zudem sind Sicherheits- und Datenschutzaspekte von großer Bedeutung, insbesondere in sensiblen Betriebsbereichen. Die KI ist gut geeignet zur Analyse von Daten in einer Netzwerkstruktur. Sie ermöglicht es, in großen Datenmengen Muster und Fehler zu erkennen, die für den Menschen schwer erkennbar sind. KI-Methoden wie das maschinelle Lernen können zur Automatisierung und Effizienzsteigerung in Netzwerken eingesetzt

werden (vgl. Latif et al., 2021). Sie tragen zur Optimierung des Datenflusses bei, ermöglichen die Vorhersage von Netzwerkausfällen und erhöhen die Netzwerksicherheit durch die Erkennung und Prävention von Sicherheitsbedrohungen (vgl. Wang & Zang, 2019, S. 139). Ein weiterer entscheidender Vorteil ist die Echtzeit-Datenanalyse, die eine schnelle und fundierte Entscheidungsfindung ermöglicht.

3 Klassifizierungsprozess der Graphen mit Hilfe von Deep Learning

3.1 Beschreibung des Datensatzes und der Datenquellen

Die Grundlage dieses Papers bilden zwei künstlich generierte Datensätze, die unterschiedliche Netzwerktopologien im IIoT-Kontext simulieren. Der erste Datensatz basiert auf einem Scale-Free Netzwerk mit einem Exponenten Gamma von 2,3, während der zweite Datensatz ein Erdős-Rényi-Netzwerk darstellt mit einer Wahrscheinlichkeit von 0,15 für die Erstellung von Kanten. Jedes dieser Netzwerke besteht aus 100 Knoten, von denen jeder 10 Datensätze, in Form eines Arrays zugewiesen bekam. Diese Datensätze wurden generiert, um Strukturen von IIoT-Netzwerken zu repräsentieren und dienen als Basis für die anschließende Klassifizierung durch KI.

3.2 Auswahl und Begründung der KI-Methoden

Für die Analyse und Klassifizierung der Netzwerke wurde sich für einen KI-basierten Ansatz mit einem Graph-Neural-Network (GNN) entschieden. Das GNN arbeitet direkt mit einer Graphen basierten Struktur, weshalb es sich besonders für die vorliegenden Netzwerktopologien eignet. Es zeigt außerdem gute Ergebnisse bei verschiedenen Anwendungen, unter anderem der Knoten- und Graph Klassifikation (vgl. Chen et al., 2021). GNNs werden bereits in verschiedenen Webanwendungen wie Sozialen Netzwerken, bei der Betrugserkennung und Empfehlungssystemen eingesetzt (vgl. Song et al., 2021).

3.3 Beschreibung des experimentellen Aufbaus

Der experimentelle Aufbau unseres Projekts umfasste mehrere Schritte: Zunächst wurden die Datensätze unter Verwendung der NetworkX-Bibliothek mit Python generiert. Aus diesen Datensätzen wurden Laplacian Matrizen erzeugt. Anschließend wurden Vektoren mit 10 zufälligen Werten den Netzwerkknoten zugewiesen. Diese simulieren Daten, die im IIoT-Kontext generiert werden. Es folgte die Kombination von Labeln und Daten und die anschließende Aufteilung der Daten in Trainings- und Testsets, um eine umfassende Bewertung der Modellleistung zu ermöglichen. Das Training des GNN fand dann mit den Trainingsdaten statt, die Validierungsdaten wurden zur Überwachung der Leistung und dem Erkennen von Overfitting genutzt.

4 Implementierung

4.1 Datenverarbeitung und Feature-Engineering

Um die Daten für das Modell verarbeitbar zu machen, wurden sie in eine geeignete Form gebracht. Dazu wurden nach der Erstellung der beiden Graphen die Laplacian-Matrizen berechnet (vgl. Buss et al., 2023). Lapla-

cian-Matrizen sind in der Graphentheorie weit verbreitet, da sie wichtige Informationen über die Verbindungen zwischen den Knoten erfassen und somit sehr übersichtlich Einblick in die Struktur der Graphen bieten (vgl. Babarinsa & Kamarulhaili, 2017). In Abbildung 2 ist die Laplacian-Matrix eines Erdős-Rényi-Graphen und eines Barabási-Albert-Graphen dargestellt mit zufälligen Vektoren auf den Knoten der Kanten.

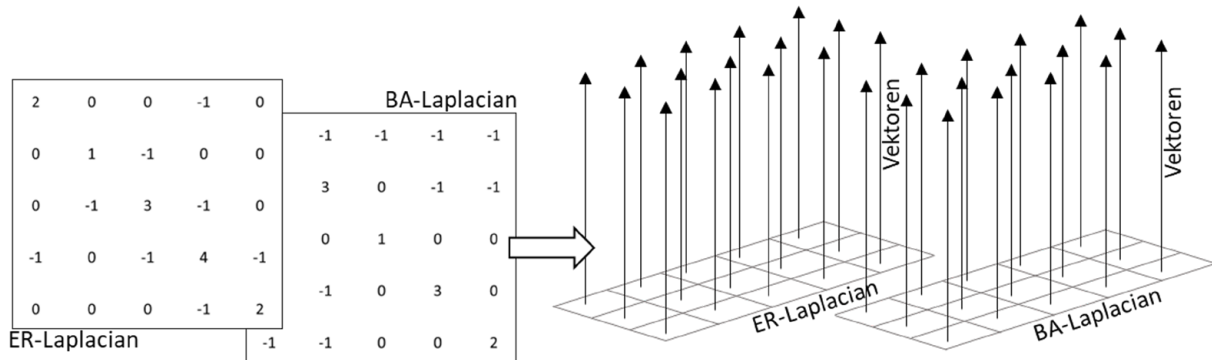


Abbildung 2: Laplacian-Matrizen eines Erdős-Rényi-Graphen und eines Barabási-Albert-Graphen mit aufgesetzten Vektoren (eigene Darstellung)

Allen Kanten des Barabási-Albert-Graphen wird das Label „1“ zugewiesen, während den Kanten des Erdős-Rényi-Graphen das Label „0“ zugeordnet wird. Als letzter Schritt bei der Aufbereitung der Daten werden zwei Arrays für die beiden Graphen erstellt. In diesen Arrays werden die Laplacian-Werte der Kanten mit dem zufälligen Vektor des ersten Knotens der Kante verbunden (vgl. Buss et al., 2023). Um die Daten für TensorFlow nutzbar zu machen, wurden sie in TensorFlow-Tensoren mit dem Datentyp „tf.float32“ konvertiert. Die beiden Arrays sowie die Labels der Kanten beider Graphen werden jeweils in einem eigenen Array gespeichert. Anschließend kann das Modell die Daten des ersten Arrays verarbeiten, da die Eingangsschicht des Modells die entsprechende Form definiert (vgl. Buss et al., 2023). Die Daten durchlaufen dann mehrere Dense-Schichten mit zugehörigen Aktivierungsfunktionen und Dropout-Schichten. Diese Schichten führen Transformationen auf den Eingabedaten aus und lernen dabei verschiedene Merkmale und Muster der Daten, um eine Vorhersage für die binäre Klassifikation zu treffen.

4.2 Entwicklung des Deep Neural Networks

Bei unserer Entwicklung des Modells haben wir mit der Keras-Bibliothek in Python gearbeitet. Diese half uns dabei ein sequenzielles Modell zu erstellen mit mehreren Schichten (vgl. Buss et al., 2023). Wie im Abschnitt 2.3 bereits beschrieben wurde, sind Deep Neural Networks dafür bekannt, dass sie meist sehr viele Schichten haben (vgl. Wang et al., 2019). In unserem Fall haben wir vier Schichten eingesetzt, um die Graphen zu klassifizieren. Als erste Schicht haben wir die Eingabeschicht, welche gleichzeitig auch die erste versteckte Schicht ist (vgl. Buss et al., 2023). Sie ist eine dicht verbundene Schicht, also eine Fully Connected Layer mit 256 Neuronen. Als Eingabeschicht definiert sie aber auch die Eingangsgrößen. Zu der Schicht gehört dann auch die Leaky ReLu Aktivierungsfunktion, welche keine negativen Eingaben erlaubt und die Dropout-Schicht, welche zufällig 30 % der Neuronen aus der vorherigen Schicht während des Trainings ausschaltet. Dies hilft gegen das Overfitting. Die nächsten zwei Schichten sind genauso aufgebaut bis auf, dass sie nur 128 und 64 Neuronen jeweils haben und keine Eingaben definieren. Als letzte Schicht des neuronalen Netzwerks steht dann die Ausgangsschicht, welche aus einem einzigen Neuron besteht mit einer Sigmoid-Aktivierungsfunktion (vgl. Buss et al., 2023). Die Sigmoid-Aktivierungsfunktion macht die binäre Klassifikation möglich.

4.3 Training und Validierung des Modells

Zum Modelltraining und -test wurden die Daten in Trainings- und Testdaten aufgeteilt. Dabei wurden 20 % der Daten als Testdaten verwendet und der Rest für das Training genutzt (vgl. Buss et al., 2023). Beim Modelltraining werden sowohl das Array mit den Laplacian-Werten der Kanten und dem zufälligen Vektor des ersten Knotens der Kante als auch das Array mit den Kanten-Labels dem Modell übergeben. Das Training erfolgt über 100 Epochen mit einer Batch-Größe von 32. Es werden Callback-Funktionen verwendet, nämlich „Early-Stopping“ und „ModelCheckpoint“ (vgl. Buss et al., 2023). Diese dienen dazu, das Training zu beenden, wenn keine Verbesserung mehr auf den Validierungsdaten zu erkennen ist, oder um die Modelldaten zu speichern, wenn sich die Leistung verbessert. Um dies zu ermöglichen, wurden 20 % der Trainingsdaten in Validierungsdaten umgewandelt. Das Modell wird mit den zuerst erstellten Testdaten getestet. Dabei werden die Metriken F1-Score, Loss, Accuracy, Recall und Precision verwendet (vgl. Buss et al., 2023).

5 Ergebnisse

5.1 Darstellung der Klassifizierungsergebnisse

Die Darstellung und Interpretation von Klassifizierungsergebnissen eines maschinellen Lernmodells, wie des beschriebenen neuronalen Netzwerks, ist ein wichtiger Schritt, um zu verstehen, wie gut das Modell funktioniert. Bei der Auswertung der Ergebnisse unseres KI-Modells haben wir uns für die Metriken Test-Accuracy, Test-Loss, Precision, Recall und F1-Score entschieden, da sie zusammen ein umfassendes Bild der Leistung des Modells wiedergeben. Die Test-Accuracy, die den Prozentsatz der korrekten Vorhersagen im Testset angibt, ist eine grundlegende Metrik, die die allgemeine Effektivität des Modells aufzeigt. Der Test-Loss ist eine Ergänzung, die die durchschnittliche Abweichung zwischen den Vorhersagen und den tatsächlichen Werten misst – niedrigere Werte stehen hier für eine höhere Präzision (vgl. Chen et al., 2022). Precision ist besonders aussagekräftig in Situationen, in denen die Konsequenzen falsch positiver Ergebnisse schwerwiegend sind, da sie den Anteil der korrekten positiven Vorhersagen an allen positiven Vorhersagen des Modells anzeigt. Der Recall hingegen misst, wie gut das Modell alle positiven Fälle erkennt. Das Übersehen positiver Fälle kann kontextabhängig von sehr großer Bedeutung sein (vgl. Muntean & Militaru, 2023). Der F1-Score präsentiert passend dazu die Balance zwischen Precision und Recall und ist somit eine wichtige Metrik zur Beurteilung der Gesamtleistung des Modells, insbesondere in Szenarien, in denen ein Ausgleich zwischen beiden wichtig ist (vgl. Yacouby & Axman, 2020).

- Accuracy: 97,4 % – das Modell hat die meisten Testdaten korrekt klassifiziert.
- Loss: 0,09 – ein niedriger Wert, der auf geringe Fehler der Vorhersagen hinweist.
- Precision: 93,9 % – ein hoher Anteil der positiven Vorhersagen des Modells war korrekt.
- Recall: 88,6 % – das Modell war effektiv darin, die tatsächlichen positiven Fälle zu identifizieren.
- F1-Score: 0,912 – zeigt eine starke Balance zwischen Precision und Recall.

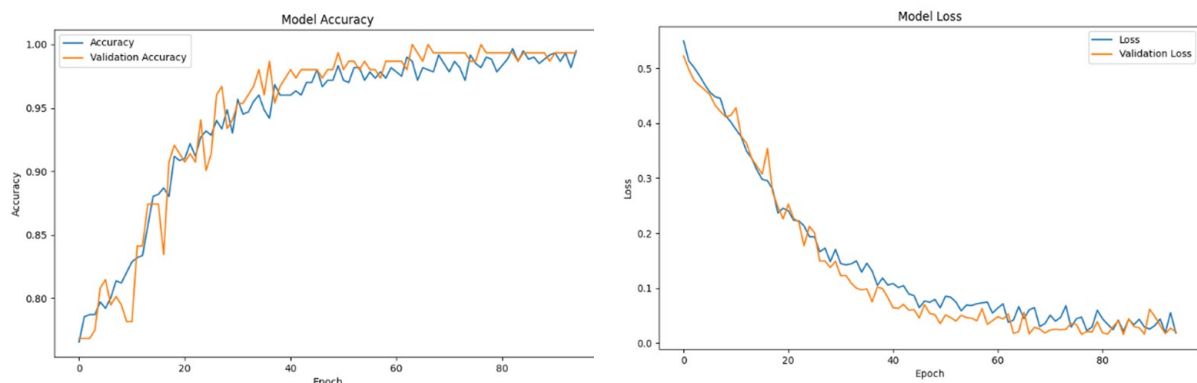


Abbildung 3: Graphen der Trainings- und Testergebnisse des Deep-Learning-Modells (eigene Darstellung)

5.2 Interpretation der Ergebnisse

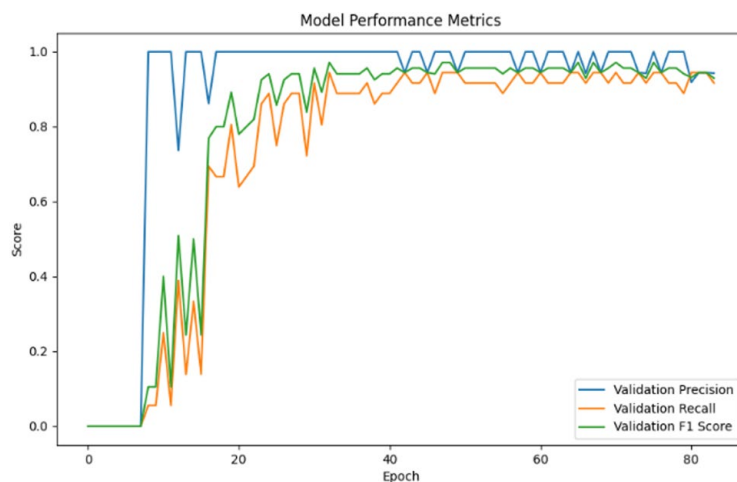


Abbildung 4: Graphen der Testergebnisse des Deep-Learning-Modells zu Precision, Recall und F1-Score (eigene Darstellung)

Das Ergebnis des Deep-Learning-Modells zeigt, dass mit dessen hoher Test-Accuracy von 0.974 und niedrigem Test-Loss von 0.09 eine gute Generalisierung und zuverlässige Vorhersagen erzielt wurde. Die Kurven für die Trainings- und Validierungs-Accuracy liegen dicht beieinander und zeigen keine großen Abweichungen. Das deutet auf ein nicht überangepasstes Modell hin (vgl. Schneider & Vlachos, 2021). Der Validierungs-Loss lässt eine ähnliche Schlussfolgerung zu. Die drei Metriken Precision, Recall und F1-Score stabilisieren sich nach etwa 20 Epochen und weisen ebenfalls gute Werte von knapp über 0.9 im Durchschnitt an, was auf eine gute Kalibrierung hinweist. Es gibt einige Schwankungen in den Metriken, was normal ist, da sich das Modell auf verschiedene Daten während des Trainings einstellt. Aufgrund der „EarlyStopping“-Funktion mit einer „Patience = 10“, hat das Deep Learning Modell nicht die vollen 100 eingestellten Epochen zum Trainieren benutzt, sondern nach 92 Epochen aufgehört. Die Ergebnisse zeigen, dass das Modell in der Lage ist, mit hoher Wahrscheinlichkeit genaue und relevante Ergebnisse zu liefern. Diese Ergebnisse sind vielversprechend für die Anwendung in Bereichen, in denen Genauigkeit von entscheidender Bedeutung ist, und bieten eine solide Grundlage für weitere Optimierungen und Anpassungen des Modells.

6 Schlussfolgerungen und Ausblick

Die Integration von KI-Modellen in IIoT-Logistiksysteme kann die Effizienz und Effektivität der Betriebsabläufe erheblich verbessern (vgl. Doshi et al., 2023). KI-Modelle können komplexe Netzwerkstrukturen analysieren und in Echtzeit Muster und Anomalien identifizieren, was eine schnellere und fundiertere Entscheidungsfindung ermöglicht (vgl. Flores-García et al., 20229). Dies führt zur Optimierung des Warenflusses und zur prädiktiven Wartung (vgl. Gautam et al., 2023), was nicht nur die Ausfallzeiten reduziert, sondern auch die Kosten senkt. Darüber hinaus kann die KI-basierte Analyse von IIoT-Daten dazu beitragen, neue Geschäftsmodelle und Dienstleistungen zu entwickeln, die einen Wettbewerbsvorteil darstellen. Die Verwendung von KI in der IIoT-Logistik steht somit für eine fortgeschrittene Automatisierung und Personalisierung, welche die Skalierbarkeit unterstützt und gleichzeitig nachhaltige Betriebspraktiken fördert (vgl. Allahloh et al., 2023). Insgesamt kann die Kombination von IIoT und KI transformative Veränderungen in industriellen Betriebsmodellen ermöglichen, was zu einer gesteigerten Effizienz, Kosteneinsparungen und einer höheren Kundenzufriedenheit führt. In der zukünftigen Forschung zur Klassifikation von Netzwerktopologien im IIoT-Kontext könnten sich einige Kernbereiche als besonders interessant erweisen. Die Entwicklung fortschrittlicher Lernalgorithmen, einschließlich Graph Neural Networks und Reinforcement Learning, könnte zu präziseren Analysemethoden führen (vgl. Balega et al., 2022). Die Fähigkeit zur Echtzeit-Datenverarbeitung ist entscheidend, um auf die dynamischen Anforderungen von IIoT-Systemen reagieren zu können (vgl. Huang et al., 2020). Anomalie-Erkennung steht im Mittelpunkt, um Sicherheitsverletzungen oder Systemausfälle frühzeitig zu identifizieren (vgl. Yan et al. 2020). Darüber hinaus könnte die Forschung zur Verbesserung der Interoperabilität zwischen verschiedenen IIoT-Geräten und -Plattformen durch intelligente Klassifizierungssysteme die Zusammenarbeit und Effizienz in der Industrie erheblich verbessern (vgl. Chen et al., 2019). Diese Forschungsbereiche könnten die Grundlage für die Schaffung fortschrittlicher, sicherer und effizienter IIoT-Systeme bilden, die mit den Anforderungen der modernen Industrie mithalten können.

Literatur

- Allahloh, A. S., Sarfraz, M., Ghaleb, A. M., Al-Shamma'a, A. A., Hussein Farh, H. M. & Al-Shaalan, A. M. (2023) „Revolutionizing IC Genset Operations with IIoT and AI: A Study on Fuel Savings and Predictive Maintenance“, *Sustainability*, Vol. 15, No. 11, S. 8808.
- Arizmendi, O. & Dominguez, E. (2022) „Barabasi – Albert trees are hypoenergetic“, *Boletín de la Sociedad Matemática Mexicana*, Vol. 28, No. 3.
- Babarinsa, O. & Kamarulhaili, H. (2017) „On Determinant of Laplacian Matrix and Signless Laplacian Matrix of a Simple Graph“, in Arumugam, S., Bagga, J., Beineke, L. W. & Panda, B. S. (Hg.) *Theoretical Computer Science and Discrete Mathematics*, Cham, Springer International Publishing, S. 212–217.
- Balega, M., Farag, W., Ezekiel, S., Wu, X.-W., Deak, A. & Good, Z. (2022) „IoT Anomaly Detection Using a Multitude of Machine Learning Algorithms“, *2022 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, S. 1–7.
- Bangachev, K. & Bresler, G. (2025), *Random Algebraic Graphs and Their Convergence to Erdős-Rényi*. *Random Struct Alg*, 66: e21276. <https://doi.org/10.1002/rsa.21276>
- Berghammer, R. (2021) *Mathematik für die Informatik: Grundlegende Begriffe, Strukturen und Anwendungen*, Springer Professional „Wirtschaft + Technik“, Springer Fachmedien Wiesbaden.
- Bishop, C. M. & Bishop, H. (2024) „Graph Neural Networks“, in Bishop, C. M. & Bishop, H. (Hg.) *Deep Learning*, Cham, Springer International Publishing, S. 407–427.
- Blumenfeld, Z. (2023) „Graph Machine Learning: An Overview“, *Towards Data Science* [Online]. Verfügbar unter <https://towardsdatascience.com/graph-machine-learning-an-overview-c996e53fab90?gi=537cf60163ad> (Abgerufen am 4 Dezember 2023).

- Borges, J. B., Ramos, H. S. & Loureiro, A. A. F. (2022) „A Classification Strategy for Internet of Things Data Based on the Class Separability Analysis of Time Series Dynamics“, *ACM Transactions on Internet of Things*, Vol. 3, No. 3, S. 1–30.
- Boyes, H., Hallaq, B., Cunningham, J. & Watson, T. (2018) „The industrial internet of things (IIoT): An analysis framework“, *Computers in Industry*, Vol. 101, S. 1–12.
- Buss, A., Lang, J., Oppenländer, H. & Villalba-Diez, J. (2023) IIOT Deep Neural Network [Online], Heilbronn. Verfügbar unter <https://github.com/Jannis2002/IIOT-Deep-Neural-Network> (Abgerufen am 20 Dezember 2023).
- Chen, D., Lu, Y. & Hsu, C.-Y. (2022) „Measurement Invariance Investigation for Performance of Deep Learning Architectures“, *IEEE Access*, Vol. 10, S. 78070–78087.
- Chen, J., Lin, X., Xiong, H., Wu, Y., Zheng, H. & Xuan, Q. (2021) „Smoothing Adversarial Training for GNN“, *IEEE Transactions on Computational Social Systems*, S. 618–629.
- Chen, L., Chen, W., Wang, L., Zhai, C., Hu, X., Sun, L., Tian, Y., Huang, X. & Jiang, L. (2023) „Convolutional neural networks (CNNs)-based multi-category damage detection and recognition of high-speed rail (HSR) reinforced concrete (RC) bridges using test images“, *Engineering Structures*, Vol. 276, S. 115306.
- Chen, N., Qiu, T., Zhou, X., Li, K. & Atiquzzaman, M. (2019) „An Intelligent Robust Networking Mechanism for the Internet of Things“, *IEEE Communications Magazine*, S. 91–95.
- Dagnino, A. (2021) *Data Analytics in the Era of the Industrial Internet of Things*, Cham, Springer International Publishing; Imprint Springer.
- Doshi, S., Ghatage, P. & Borole, R. (2023) „IIoT for Factory Optimization and Logistics Management using PTC Thingworx and Kepware“, 2 (ICETET – SIP), *IEEE*, S. 1–5.
- Elhaloui, L., El Filali, S., Benlahmer, E. H., Tabaa, M., Tace, Y. & Rida, N. (2023) „Machine learning for internet of things classification using network traffic parameters“, *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 13, No. 3, S. 3449.
- Flores-García, E., Gopalakrishnan, M., Jeong, Y. & Wiktorsson, M. (2022) „IIoT-Enabled Digital Services for Maintenance Planning in Smart Production Logistics Using Maintenance Opportunity Window“, IOS PRESS, INC.
- Fuhrmann, J. (2019) „Implementierung einer Tensor Processing Unit mit dem Fokus auf Embedded Systems und das Internet of Things“, in Unger, H. (Hg.) *Echtzeit 2019: Autonome Systeme – 50 Jahre PEARL*, Fachtagung des gemeinsamen Fachausschusses Echtzeitsysteme von Gesellschaft für Informatik e. V. (GI), VDI, Wiesbaden, Springer Vieweg, S. 61–70.
- Gauger, I., Nagel, T. & Huber, M. (2022) „Hybrides Maschinelles Lernen im Kontext der Produktion“, in Hartmann, E. A. (Hg.) *Digitalisierung souverän Gestalten: Handlungsspielräume in digitalen*, [S.l.], Morgan Kaufmann, S. 64–79.
- Gautam, S., Noureddine, R. & Solvang, W. D. (2023) „Machine Learning and IIoT Application for Predictive Maintenance“, in Wang, Y. (Hg.) *Advanced Manufacturing and Automation XII*, Singapore, Springer Verlag, S. 257–265.
- Huang, H., Ding, S., Zhao, L., Huang, H., Chen, L., Gao, H. & Ahmed, S. H. (2020) „Real-Time Fault Detection for IIoT Facilities Using GBRBM-Based DNN“, *IEEE Internet of Things Journal*, Vol. 7, No. 7, S. 5713–5722.
- Kreßel, U., Schürmann, J. & Franke, J. (1991) „Neuronale Netze für die Musterklassifikation“, in Radig, B. & DAGM-Symposium 13, 1. M. (Hg.) *Mustererkennung 1991: 13. DAGM-Symposium München, Oktober 1991*; Proceedings, Berlin [u. a.], Springer, S. 1–18.
- Lanquillon, C. & Schacht, S. (2023) *Knowledge Science – Grundlagen: Methoden der künstlichen Intelligenz für die Wissensextraktion aus Texten*, [S.l.], Springer Vieweg.
- Latif, S., Driss, M., Boulila, W., Huma, Z. E., Jamal, S. S., Idrees, Z. & Ahmad, J. (2021) „Deep Learning for the Industrial Internet of Things (IIoT): A Comprehensive Survey of Techniques, Implementation Frameworks, Potential Applications, and Future Directions“, *Sensors (Basel, Switzerland)*, Vol. 21, No. 22.
- Manaswi, N. K. (2018) „Convolutional Neural Networks“, in Manaswi, N. K. (Hg.) *Deep Learning with Applications Using Python*, Berkeley, CA, Apress, S. 91–96.
- Mandjes, M., Starreveld, N., Bekker, R. & Spreij, P. (2019) „Dynamic Erdős-Rényi Graphs“, in Steffen, B. & Woeginger, G. (Hg.) *Computing and Software Science*, Cham, Springer International Publishing, S. 123–140.

- Martha Rodríguez, Diana P. Tobón & Danny Múnera (2023) „Anomaly classification in industrial Internet of things: A review“, *Intelligent Systems with Applications*, Vol. 18, S. 200232.
- Mirjalili, S. (2019) *Evolutionary Algorithms and Neural Networks*, Cham, Springer International Publishing.
- Muntean, M., Militaru, F.D. (2023). Metrics for Evaluating Classification Algorithms. In: Ciurea, C., Pocatilu, P., Filip, F.G. (eds) *Education, Research and Business Technologies. Smart Innovation, Systems and Technologies*, vol 321, S. 307-317. Springer, Singapore. https://doi.org/10.1007/978-981-19-6755-9_24.
- Namvar, A., Thapa, C., Kanhere, S. S. & Camtepe, S. (2021) „Evaluating the Security of Machine Learning Based IoT Device Identification Systems Against Adversarial Examples“, in Hacid, H., Kao, O., Mecella, M., Moha, N. & Paik, H. (Hg.) *Service-Oriented Computing: 19th international conference, icsoc 2021, dubai, [S.l.]*, Springer, S. 800–810.
- Oberländer et al. (2021) „IIoT-basierte Geschäftsmodellinnovation im Industrie-Kontext – Archetypen und praktische Einblicke“, Wiesbaden, Springer Fachmedien Wiesbaden, S. 23–35.
- Petrik, D. (2022) „IIoT, digitale Plattformen und Plattformökosysteme“, in *Management der Zufriedenheit der Wertschöpfungspartner auf Basis der Boundary Resources im IIoT*, Wiesbaden, Springer Fachmedien Wiesbaden, S. 25–62.
- Poddig, T. (2000) „Klassische neuronale Netze und ihre Anwendung“, in Müller, W. A., Bol, G., Nakhaeizadeh, G. & Vollmer, K.-H. (Hg.) *Datamining und Computational Finance*, Heidelberg, Physica-Verlag HD, S. 143–170.
- Ryabchenko, A. A. & Samosvat, E. A. (2010) „On the number of subgraphs of a random graph in the Barabasi-Albert model“, *Doklady Mathematics*, Vol. 82, No. 3, S. 946–949.
- Schmihing, F. & Jochem, R. (2021) „Steigerung von Daten- und Prozessqualität mit Hilfe von Visual Analytics in der Produktion“, in Schmitt, R. H. (Hg.) *Datengetriebenes Qualitätsmanagement: Bericht zur gqw-jahrestagung 2019*, S. 1–19.
- Schneider, J. & Vlachos, M. (2021) „Personalization of Deep Learning“, in Haber, P., Lampoltshammer, T., Mayr, M. & Plankensteiner, K. (Hg.) *Data Science – Analytics And Applications: Proceedings of the 3rd international*, [S.l.], Springer, S. 89–96.
- Song, X., Ma, R., Li, J., Zhang, M. & Wipf, D. P. (2021) „Network In Graph Neural Network“, *arXiv.org*.
- Wang, X. & Zhang, T. (2019) „Reinforcement Learning Based Resource Allocation for Network Slicing in 5G C-RAN“, *2019 Computing, Communications and IoT Applications (ComComAp)*, Shenzhen, China, 2019, pp. 106–111, doi: 10.1109/ComComAp46287.2019.9018774.
- Wang, Z., Mao, S. & Yang, W. (2019) „Deep learning approach to multimedia traffic classification based on QoS characteristics“, *IET Networks*, Vol. 8, No. 3, S. 145–154.
- Wu, L., Cui, P., Pei, J. & Zhao, L. (Hg.) (2022) *Graph Neural Networks: Foundations, Frontiers, and Applications*, Singapore, Springer Nature Singapore.
- Wu, Y., Dai, H.-N. & Tang, H. (2022) „Graph Neural Networks for Anomaly Detection in Industrial Internet of Things“, *IEEE Internet of Things Journal*, Vol. 9, No. 12, S. 9214–9231.
- Yacoub, R. & Axman, D. (2020) „Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models“, *Online: Association for Computational Linguistics*, S. 79–91. Verfügbar unter: <https://aclanthology.org/2020.eval4nlp-1.9/>.
- Yan, X., Xu, Y., Xing, X., Cui, B., Guo, Z. & Guo, T. (2020) „Trustworthy Network Anomaly Detection Based on an Adaptive Learning Rate and Momentum in IIoT“, *IEEE Transactions on Industrial Informatics*, Vol. 16, No. 9, S. 6182–6192.