Optimierung der Pünktlichkeit: Ein Vergleich von Random Forest und Neuronalen Netzen im öffentlichen Verkehr

Im Rahmen des Moduls "Einführung Künstliche Intelligenz"

an der Digital Business University of Applied Sciences DBU Berlin

Autor:	Simon Sahli
Studiengang:	Data Science und Management (M.Sc.)
Immatrikulation:	190252
Adresse:	CH-3065 Bern
E-Mail:	simon.sahli@student.dbuas.de
Datum der Einreichung:	25.06.2024

Inhaltsverzeichnis

1 Einleitung

Im öffentlichen Verkehr spielt die Pünktlichkeit eine wichtige Rolle und ist entscheidend für die Kundenzufriedenheit[9]. Um einen pünktlichen öffentlichen Verkehr zu ermöglichen, bedarf es eines gesicherten Substanzerhalts und einer ausreichenden Finanzierung der Bahninfrastruktur. An seiner Sitzung vom 15. Mai 2024 hat der Bundesrat in der Schweiz die Botschaft verabschiedet, für die Jahre 2025-2028 einen Zahlungsrahmen von 16.4 Milliarden CHF für den Betrieb und die Erneuerung des bestehenden Schienennetzes, der Bahnanlagen und der Bahnhöfe bereitzustellen[2].

Im Rahmen seiner Botschaft informiert der Bundesrat auch über den Zustand der Anlagen sowie die Belastung und Auslastung der Bahninfrastruktur. Insgesamt befindet sich diese, bei hoher Verkehrsbelastung, in einem ausreichenden bis guten Zustand.
Einzelne Anlagen, Anbieter und Strecken, die möglicherweise aufgrund unzureichender Investitionen in den Substanzerhalt der Infrastruktur oder durch nicht optimale Fahrplanplanung im Vergleich zu anderen Strecken öfter Probleme haben, sollten sich auch in der Pünktlichkeit der Fahrten widerspiegeln. So lautet die grundlegende Prämisse dieser Arbeit.

Diese Arbeit hat zum Ziel, Ist-Daten des öffentlichen Verkehrs des Monats Mai 2024 im Hinblick auf Verspätungen zu analysieren. Dabei handelt es sich um Ist-Daten der konzessionierten Transportunternehmen der Schweiz. Gemäss den Erläuterungen der Kolleginnen und Kollegen von opentransportdata.swiss [3] sind die tatsächlich gefahrenen Fahrten aller erfassten Transportunternehmen in Relation zu den geplanten Fahrten (Fahrplan) abgebildet. Neben Bus und Tram insbesondere auch alle Zugfahrten in der Schweiz. Die Daten sollen mithilfe von supervised und deep learning Modellen analysiert werden. Dabei werden zwei verschiedene Architekturen auf den Datensatz angewendet: Einerseits wird mit der Python Library scikit-learn [11] ein Random Forest Modell erstellt, andererseits mit TensorFlow und Keras [13] ein sequentielles neuronales Netzwerk aufgegleist.
Somit ergibt sich für die Studienarbeit folgende Hypothese: Durch Modellarchitekturen im Bereich von supervised und deep learning können Ansätze im Bereich der künstlichen Intelligenz gefunden werden, um Verspätungen im öffentlichen Verkehrssystem zu einem gewissen Grad antizipieren zu können.

Eine Übersicht über dieses Projekt, ein Screencast so wie auch die Datengrundlage ist unter folgendem Link einsehbar:
`https://www.simonsahli.ch/project_verspaetungen.html`

2 Daten und eine explorative Analyse

Für die explorative Datenanalyse (EDA) ist der erste Schritt das Einlesen der Ist-Daten des Monats Mai. Ist-Daten sind die tatsächlichen Daten, die im Vergleich zu den geplanten Fahrten (dem Fahrplan) aufgezeichnet wurden. Ist-Daten sind die tatsächlich gefahrenen Fahrten aus Sicht der Kundeninformation und sind erst nach Erbringung der Leistung verfügbar, also erst im Augenblick oder danach, so die Erläuterungen der Kolleginnen und Kollegen von opentransportdata.swiss[3].

Die Daten, auf welche diese Arbeit basiert[15], umfassen 21 Variablen je Datenzeile. Jede Datenzeile beschreibt eine Fahrt eines bestimmten Transportmittels (Zug, Bus, Tram, Zahnradbahn) zu bzw. ab einer Haltestelle gemäss Fahrplan. Dies sind ca. 2 Millionen Datenpunkte je Tag und werden täglich auf der Open-Data-Plattform Mobilität Schweiz (ODMCH) vom Team Systemaufgaben Kundeninformation Plus (SKI+) im Auftrag des Bundesamtes für Verkehr (BAV) publiziert[3].
Für die Ist-Daten des gesamten Monats Mai wären dies 66'575'360 Datenzeilen à 21 Spalten. Aufgrund der Grösse des Datensatzes und der benötigten Rechnerleistung wurde der Ist-Datensatz auf die ersten zwei Mai-Wochen eingeschränkt. Dieses Subsample beinhaltet noch immer 29'419'770 Datenbeobachtungen.

Von den 21 Variablen des Datensatzes sind für diese Arbeit insbesondere von Relevanz die Verkehrsträger (PRODUKT_ID) sowie die Ankunfts- bzw. Abfahrtszeit (AB_PROGNOSE) im Vergleich zur Prognose bzw. zum Fahrplan (ABFAHRTSZEIT). Somit lässt sich aus Kundensicht eine Abfahrtsverspätung (DELAY = AB_PROGNOSE - ABFAHRTSZEIT)[1] berechnen, welche als Variable im Datensatz ergänzt wird. Um diese Verspätung zu definieren, wird der gesamte Datensatz zunächst auf Datensätze mit effektiven Ist-Abfahrtszeiten (AB_PROGNOSE_STATUS = "Real") eingeschränkt. Dies ist im gesamten Datensatz in 75.26 Prozent der Fälle gegeben. Die restlichen Beobachtungen werden aus dem Datensatz entfernt, da es sich beim Rest um geschätzte effektive Abfahrten oder vom System berechnete effektive Abfahrtszeiten handelt, welche in dieser Arbeit als nicht relevant betrachtet werden.

Nach dieser Bereinigung verbleiben insgesamt noch 19'061'133 Beobachtungen im Datensatz. Davon fehlen bei über 1'699'105 Beobachtungen die Namen der Haltestellen (HALTESTELLEN_NAME), was jedoch vernachlässigt werden kann, da die Namen der Haltestellen nicht weiter betrachtet werden und die numerische und international standardisierte Codierung der Haltestellen (BPUIC), welche für eine eindeutige Identifikation verwendet wird, vollständig vorhanden ist.

[1] AB_PROGNOSE mit Status "Real" entspricht effektiver Ist-Abfahrtszeit und ABFAHRTSZEIT entspricht dem Soll-Wert gemäss Fahrplan

Insgesamt zählt der Datensatz 18'687 verschiedene Haltestellen, welche von 106 verschiedenen Betreiberinnen und Betreibern angefahren werden. Insbesondere häufig vertreten im Datensatz sind die Bus- und Trambetriebe (viele Haltestellen, dichter Fahrplan). Bei 35 Beobachtungen im Datensatz ist nicht ersichtlich, welcher Produkttyp das Verkehrsmittel beinhaltet - diese Datenpunkte werden für die weiteren Analysen entfernt.

Bevor die Verspätungen genauer kalkuliert werden, ist es interessant zu ermitteln, welche Transportunternehmen im Rahmen der über 19 Millionen Beobachtungen vertreten sind. Zu den sechs grössten Unternehmen zählen insbesondere Busunternehmen. So zählt "PostAuto AG" mit 3'204'569 angefahrenen Haltestellen die meisten Positionen. Dies ist aufgrund der hohen Fahrplandichte und der Grösse des Netzes plausibel.

In einem nächsten Schritt werden die Verspätungen kalkuliert und als separate Datenspalte (DELAY) in Minuten ausgewiesen. Um Ausreisser zu analysieren, wurde eine obere (.99) und untere (.01) Schwellenwert-Grenze definiert und dabei 377'715 Beobachtungen herausgefiltert. Gerade die extremen Ausprägungen dieser Ausreisser können bei genauerer Betrachtung aus Datenfehlern entstanden sein. So weist beispielsweise die Fahrt vom 13.05.2024 des Unternehmens "TPF Auto Verspätungen" von -54,40 Jahren auf, was darauf schliessen lässt, dass das effektive Ankunftsdatum auf den 01.01.1970 gesetzt wurde und offensichtlich ein Systemfehler vorlag. Diese extremen Ausreisser (1.98 % aller Beobachtungen) wurden für die weitere Betrachtung des Datensatzes entfernt.

Die folgenden Abbildungen geben einen ersten Überblick über den Datensatz im Zusammenhang mit den Abfahrtsverspätungen. Abbildung 1 zeigt die Verteilung der Abfahrtsverspätung über den betrachteten Datensatz. In Abbildung 2 ist die durchschnittliche Verspätung in Minuten je Betriebsart aufgezeigt. Es fällt bereits auf, dass die Busverbindungen hinsichtlich der Pünktlichkeit bei der Abfahrt während der ersten zwei Mai-Wochen am meisten Schwierigkeiten hatten. Abbildung 3 schlüsselt die durchschnittliche Verspätung nach Wochentagen auf. Bei 14 Tagen, für die Daten eingelesen wurden, kam jeder Wochentag zweimal im Datensatz vor. Dienstag waren die Abfahrten im Durchschnitt am meisten mit Verspätung behaftet, hingegen war Sonntag vergleichsweise sehr pünktlich, was die Abfahrten betrifft. Abbildung 4 zeigt die Verteilung der Fahrten mit Verspätung der sechs im Datensatz am häufigsten vertretenen Transportunternehmen. Abbildung 5 bzw. 6 zeigen über die ersten zwei Mai-Wochen im Jahr 2024 die Haltestellen mit den meisten bzw. wenigsten Verspätungen.

Abbildung 1: Verteilung der Verspätung bei Abfahrt (Status: Real, ohne Ausreisser)

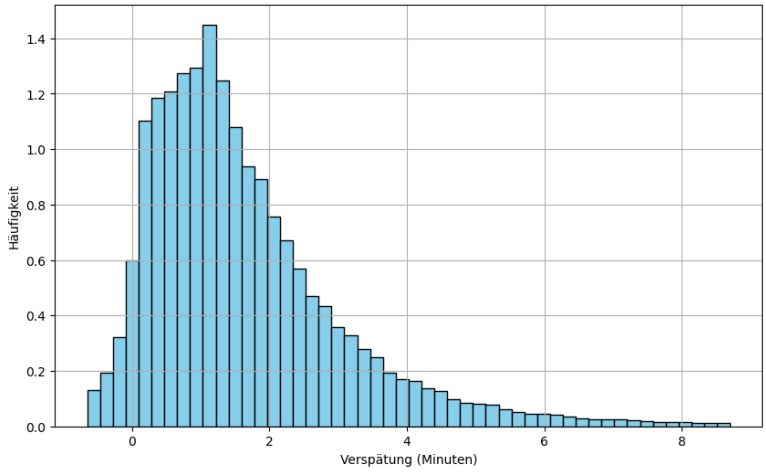


Abbildung 2: Durchschnittliche Verspätung bei Abfahrt nach Verkehrsart

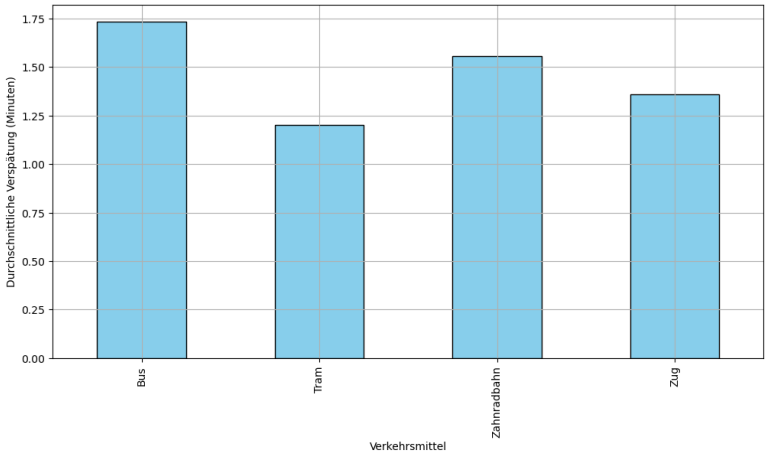


Abbildung 3: Durchschnittliche Verspätung bei Abfahrt nach Wochentag

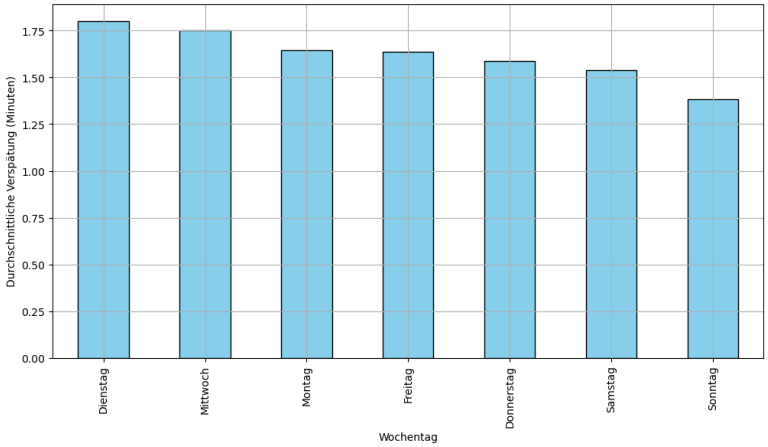


Abbildung 4: Verteilung der Fahrten der sechs grössten Transportunternehmen

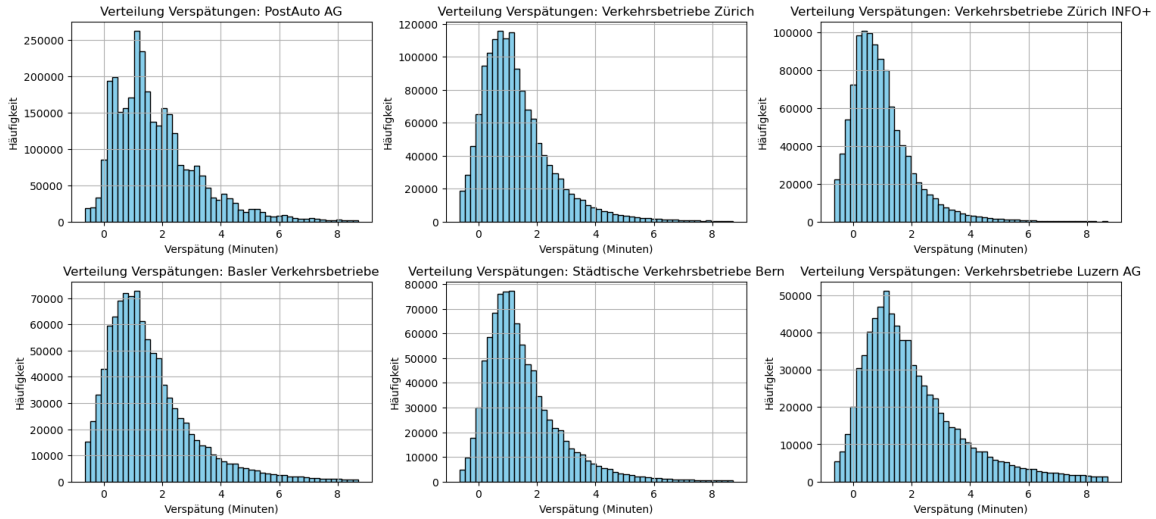


Abbildung 5: Top 10 Haltestellen mit der höchsten durchschnittlichen Verspätung bei Abfahrt

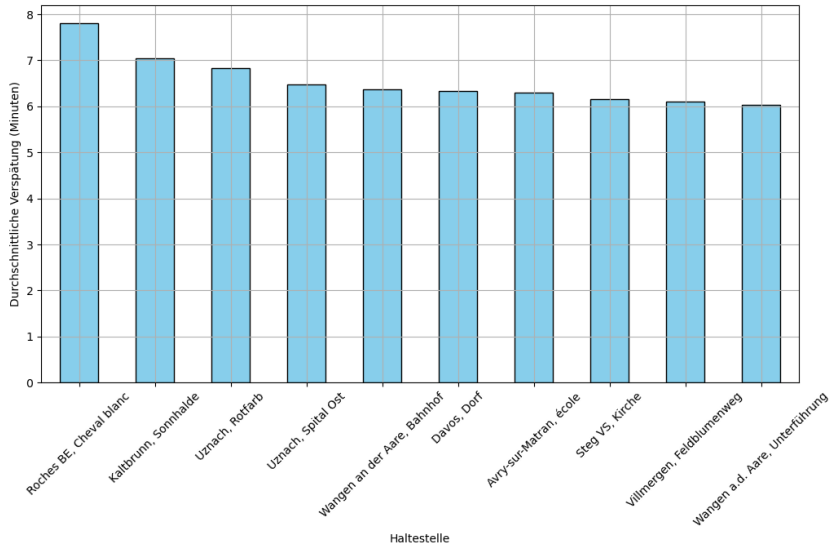
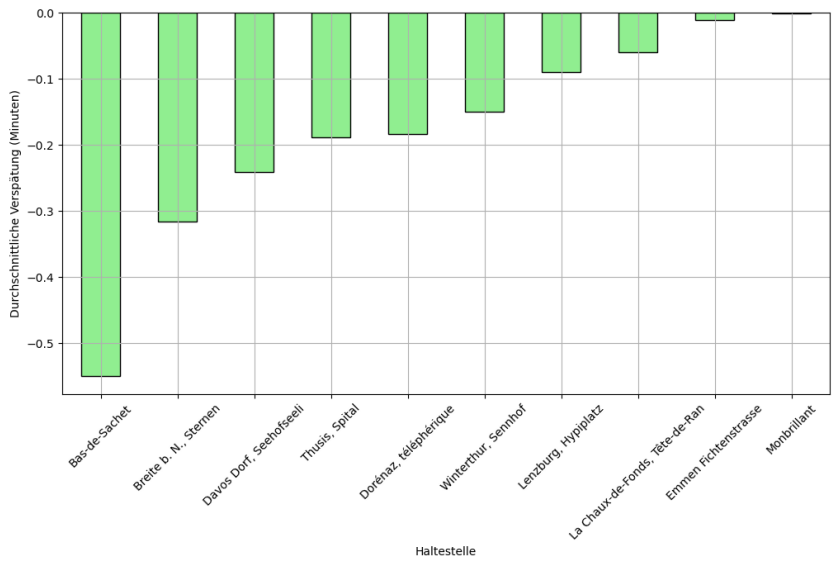


Abbildung 6: Top 10 pünktlichste Haltestellen mit der geringsten durchschnittlichen Verspätung bei Abfahrt



3 Modellierung

Bei der Untersuchung von Abfahrtsverspätungen werden nun zwei unterschiedliche Modellarchitekturen gewählt: Supervised Learning mit Hilfe eines Random Forest Regressor Modelles und eine Deep Learning Architektur mit Hilfe eines sequentiellen neuronalen Netzes. Ziel dieser Arbeit ist es, diese Modelle, jeweils mit ihren spezifischen Eigenschaften, auf den Datensatz anzuwenden und für die Verspätungsvorhersage auszuwählen. Anschliessend sollen die Ergebnisse verglichen werden.

3.1 Random Forest Regressor

Mohri, Rostamizadeh und Talwalkar (2012)[7] erläutern die Grundprinzipien des sogenannten "Supervised Learning": Trainingsdaten werden verarbeitet, um eine Funktion zu bilden, die Daten (X) auf erwartete Ausgabewerte (y) abbildet. In Kontext dieser Arbeit werden die erwarteten Ausgabewerte als Verspätung in der Abfahrt je Haltestelle in Minuten interpretiert.

Entscheidungsbäume sind im Vergleich zu anderen überwachten Lernmodellen leicht verständlich und interpretierbar. Song und Lu (2015)[12] betonen die Vorteile dieser Modelle. Neben der einfachen Interpretierbarkeit sind Entscheidungsbäume flexibel in der Anwendung: Sie können verschiedene Datentypen mischen und nicht-lineare Zusammenhänge darstellen. Da keine Distanzmasse erforderlich sind, ist nur wenig Datenvorverarbeitung nötig und sie sind robust gegenüber Ausreissern. Allerdings leiden Entscheidungsbäume unter hoher Varianz und Overfitting, wenn sie ohne Einschränkungen trainiert werden, was ihre Vorhersagekraft für unbekannte Daten mindert.

Eine Lösung, die die Flexibilität der Entscheidungsbaum-Modelle nutzt und gleichzeitig ihre Neigung zur Überanpassung reduziert, ist das Ensemble Learning. Mohammed und Kora (2023)[6] erklären, dass Ensemble Learning mehrere Basismodelle kombiniert, um ein leistungsfähigeres und genaueres Modell zu erstellen. Beim Bagging, wie von Abellán und Masegosa (2009)[1] beschrieben, werden Modelle auf unterschiedlichen Teilmengen der Daten trainiert. Wird Bagging mit Entscheidungsbäumen und zufällig gewählten Features angewendet, nennt man dies Random Forest. Diese Methode soll die Genauigkeit erheblich verbessern, indem sie die Einfachheit von Entscheidungsbäumen mit der Flexibilität des Ensemble Learning kombiniert.

Laut der Dokumentation von scikit-learn[11] können für die Anwendung der hier betrachteten Regressionsalgorithmen keine Datensätze mit fehlenden Werten (NaN) verwendet werden. Im einem nächsten Schritt werden X und y definiert, wobei X die relevanten Features ohne die Variable DELAY umfasst. DELAY wird als Zielvariable y separiert. Die Features für das Training des

RandomForestRegressor-Modells[8] umfassen neben den Transportunternehmen (BETREIBER_-NAME), Betriebstag (WOCHENTAG) und der Art des Fahrzeugs (PRODUKT_ID) auch eine neu berechnete Variable: die theoretische Fahrtdauer gemäss Fahrplan (FAHRZEIT = ANKUNFTSZEIT - ABFAHRTSZEIT). Diese wird für jede Datenzeile berechnet und in das Datenmodell integriert. Kategoriale Variablen werden mittels One-Hot-Encoding in Dummy-Variablen umgewandelt. Der Datensatz wird dann in 70 % Trainingsdaten (X_train, y_train) und 30 % Testdaten (X_test, y_test) unterteilt. Random-Forest-Modelle, wie auch andere Supervised Learning-Modelle, werden auf den Trainingsdaten trainiert und auf den Testdaten geprüft, um Verzerrungen zu vermeiden. Datenvorverarbeitung wie Skalierung ist nicht erforderlich, da diese Modelle bedingte Wahrscheinlichkeiten nutzen und kein Distanzmass im Hintergrund. Über die Ergebnisse des Modelles wird in Kapitel 3.3 berichtet.

Eine mögliche Massnahme zur Verbesserung des Modells wäre beispielsweise die Durchführung einer GridSearch, um ein Hyperparameter-Tuning zu ermöglichen. Dies ermöglicht die Suche nach einer Optimierung der Parameter im Rahmen des RandomForestRegressor-Modells, was, sofern eine Optimierung gefunden wird, bessere Ergebnisse liefern kann. Ein einzelnes Modelltraining hat bisher rund 120 Minuten gedauert. Sofern eine GridSearch beispielsweise 72 Parameterkombinationen und 3 Folds verwendet, müsste jedes Modell 3 Mal trainiert werden, also insgesamt 216 Trainingsdurchläufe. Dies gäbe eine geschätzte Berechnungszeit von 432 Stunden. Um dennoch optimierende Parameter zu finden, wird GridSearch auf ein zufälliges Subsample von n=100.000 Beobachtungen durchgeführt. Mit diesem Ansatz wurden jedoch keine optimierten Parameter gefunden, die die Modellevaluation des Testsamples verbessert hätten. Genauere Informationen dazu sind im entsprechenden Codeabschnitt im Anhang aufgeführt.

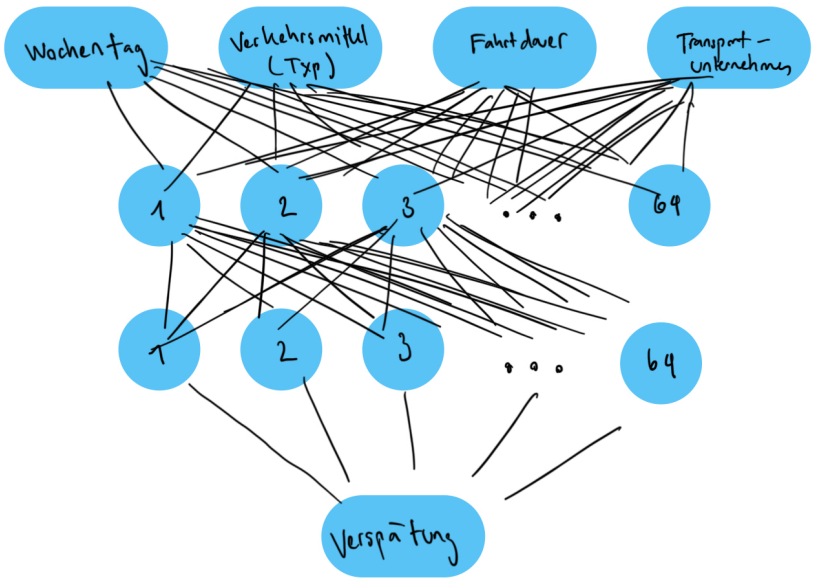
3.2 Neuronales Netz mit Keras TensorFlow

Als zweite Modellarchitektur werden Ansätze des Deep Learnings aufgegriffen, und ein Neuronales Netz wird mit dem von Google stammenden Framework Keras aus TensorFlow aufgebaut, trainiert und verwendet, um Vorhersagen hinsichtlich der Abfahrtsverspätung treffen zu können. Aufgrund der Struktur und Menge der Daten ist es essenziell, ein Overfitting des Modells zu verhindern. In dieser Arbeit wird die Flexibilität von TensorFlow genutzt, indem sogenannte High-Level Keras API Operationen verwendet werden, um das Modell einerseits zu definieren und zu trainieren. Die Daten werden auf die Server von Google hochgeladen, um mit Hilfe von Google Colab Notebooks direkt auf das TensorFlow Framework zugreifen zu können. Mit Hilfe des Google Compute Engine-Back-Ends können somit auf 51.0 GB System-RAM zugegriffen werden.

Ein Merkmal eines neuronalen Netzes ist, dass zwischen den Input-Daten und dem Output sogenannte Hidden States bzw. Dense Layers zwischengeschaltet werden. Mit Hilfe von Aktivierungsfunktionen können nicht-lineare Aspekte in diese Schichten eingerechnet werden. Im Modell dieser Arbeit besteht die letzte Schicht aus einem einzelnen Neuron, da es sich um eine Regressionsaufgabe handelt und eine einzelne kontinuierliche Zielvariable vorhergesagt werden soll (die Verspätung in Minuten). Für die Aktivierungsfunktion wird eine ReLU (Rectified Linear Unit) Funktion (relu(x) = max(0, x)) verwendet. Insgesamt besteht das Modell aus zwei Hidden Layers, jeweils mit 64 Neuronen. Im Artikel ("ReLU Activation Function Explained", o. J.)[10] werden die Vorzüge der ReLU-Funktion anschaulich erklärt. Diese ist einerseits weniger rechenintensiv, aufgrund der mathematischen Charakteristik der max()-Funktion, andererseits werden positive Werte stärker aktiviert.

Das Modell wird mit dem "adam"-Optimierungsalgorithmus kompiliert, gemäss der Arbeit von Kingma (2017)[5]. Zudem wird die Kostenfunktion auf den Mean Absolute Error (MAE) gesetzt, damit die Ergebnisse mit der Random Forest Architektur verglichen werden können. Das Modell wird schliesslich zehnmal über den gesamten Trainingsdatensatz trainiert (epochs=10), wobei innerhalb eines Epochs die Trainingsdaten in Batches von 32 Zeilen aufgeteilt werden, sodass der Optimierungsalgorithmus die jeweiligen Modellparameter laufend aktualisieren kann. Das Modell durchläuft die Epochen, um seine Parameter zu optimieren und den Verlust zu minimieren, um die Vorhersagegenauigkeit auf den Validierungsdaten zu maximieren. Ein niedrigerer Verlust auf den Validierungsdaten deutet darauf hin, dass das Modell gut generalisiert.
Abbildung 7 zeigt eine vereinfachte Veranschaulichung des neuronalen Netzes mit den jeweiligen Neuronen.

Abbildung 7: Neuronales Netz – Veranschaulichung



3.3 Evaluation der Ergebnisse

Modell	MSE	RMSE	Mean
Random Forest Regressor	1.8579	1.3631	–
Keras TensorFlow	1.8614	1.3643	–
	–	–	1.6373

Der Mean Squared Error (MSE) spiegelt die durchschnittliche quadratische Abweichung der Vorhersagen von den tatsächlichen Verspätungen wider – hier die Abfahrtsverspätung aus Kundensicht in Minuten. Sowohl der MSE-Wert des Random Forest Regressor Modells (1.8579) als auch der MSE-Wert des TensorFlow-Modells (1.8614) liegen nahe beieinander. Der Root Mean Squared Error (RMSE) beider Modelle ist noch etwas intuitiver zu interpretieren als der MSE, da er in der gleichen Einheit wie die Zielvariable (Minuten Verspätung) angegeben ist. Der RMSE des Random Forest Modells und des TensorFlow Modells liegt bei ca. 1.36 Minuten. Die durchschnittliche Verspätung im Datensatz beträgt 1.64 Minuten. Ein RMSE von 1.36 Minuten deutet darauf hin, dass beide Modelle die Variabilität in den Daten gut erfassen und etwas besser sind als eine naive Schätzung (Durchschnitt).

Das neuronale Netz ermöglicht keine genauere Vorhersage als das Random Forest Regressor Modell. Gleichzeitig gibt das R2-Mass an, wie gut die Modellvorhersagen die tatsächlichen Werte erklären. Ein Wert von 1 bedeutet, dass das Random Forest Regressor Modell die Daten perfekt erklären kann. Das Random Forest Regressor Modell errechnet einen Wert von 0.089. Dies bedeutet, dass das Modell nur etwa 8.9 % der Varianz in den Daten erklärt, was ziemlich niedrig ist.

Ein schlechter R2-Wert trotz gutem RMSE kann darauf hinweisen, dass das Modell nicht die gesamte Varianz in den Daten erklärt, möglicherweise aufgrund von Ausreissern oder nicht-linearen Beziehungen. Ein sehr niedriger R2-Wert kann auf Underfitting hinweisen, insbesondere wenn der RMSE relativ gut ist. Eine Analyse der Fehlerverteilung und eine Residualdarstellung können Aufschluss darüber geben.

Abbildung 8: Fehlerverteilung (Random Forest Regressor)

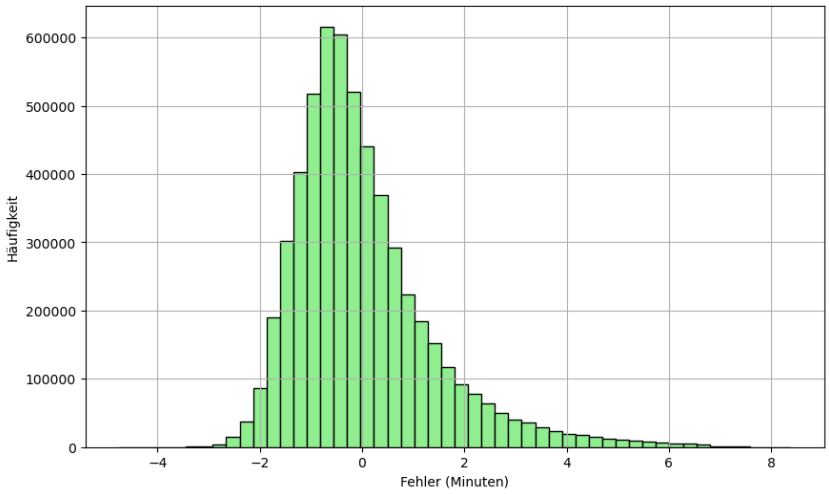


Abbildung 9: Residual Plot (Random Forest Regressor)

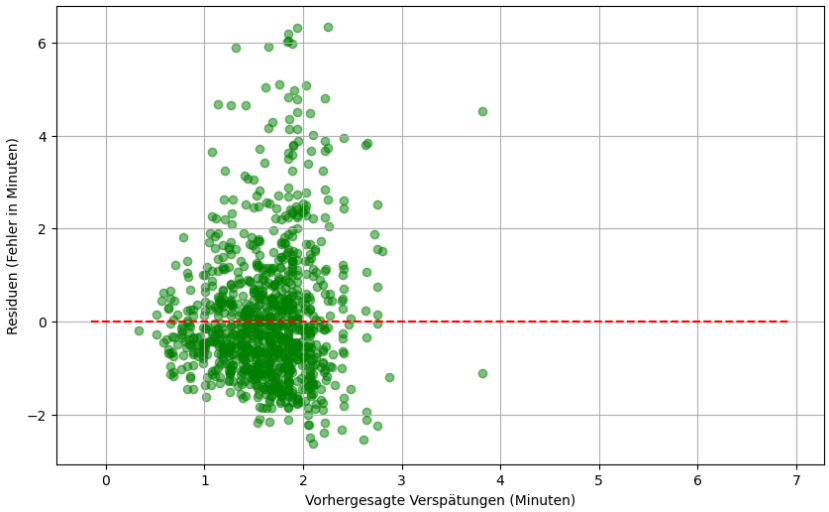


Abbildung 10: Verlustkurven (TensorFlow)

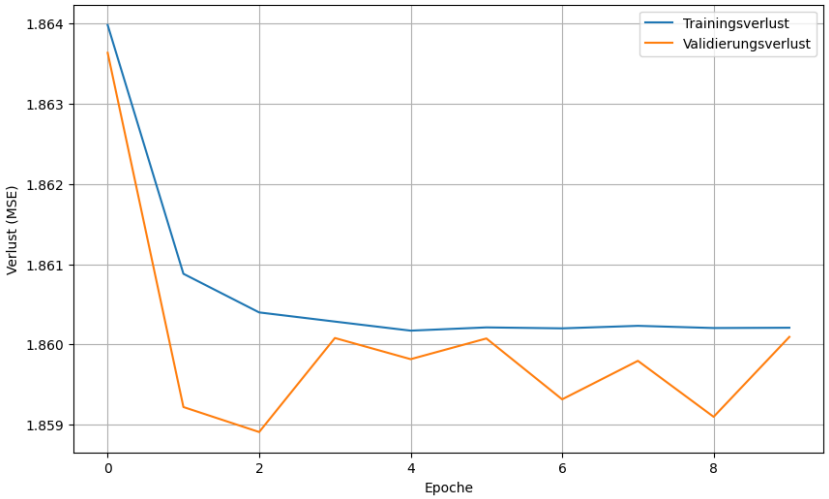


Abbildung 8 visualisiert die Fehlerverteilung des Random-Forest-Modells. Der Residuenplot in Abbildung 9 zeigt die Residuen (Fehler) gegen die vorhergesagten Verspätungen des Random Forest-Modells – zur einfacheren Lesbarkeit wurden 1000 zufällige Stichproben gezogen. Wenn die Residualdarstellung eine zufällige Streuung um den Nullpunkt zeigt, deutet dies gemäss Artikel "Was ist Underfitting?" (2023)[14] darauf hin, dass das Modell gut passt. Würde sich jedoch ein Muster in der Residualdarstellung erkennen lassen, könnte dies darauf hinweisen, dass sich das Modell nicht gut anpasst und möglicherweise Underfitting vorliegt.
Die Punkte in Abbildung 9 streuen zufällig um die Nulllinie, ohne ein klares Muster oder einen Trend zu zeigen. Dies deutet darauf hin, dass das Modell die zugrunde liegenden Muster in den Daten gut erfasst hat und die Fehler zufällig verteilt sind. Eine zufällige Streuung der Residuen spricht gegen Underfitting.

Schliesslich stellt Abbildung 10 den Trainings- sowie Validierungsverlust in Abhängigkeit von der Anzahl der Epochen aus dem TensorFlow-Modell in einer sogenannten Verlustkurve dar. Es wird deutlich, dass sich sowohl der Trainingsverlust als auch der Validierungsverlust zwischen 1.8601 und 1.8603 stabilisieren. Der endgültige MSE des Modells auf den Testdaten beträgt, wie in der Tabelle bereits ersichtlich, 1.8614, was konsistent mit den Trainings- und Validierungsverlusten ist. Die Verlustkurven driften nicht auseinander. Wäre dies der Fall, würde dies auf Overfitting hinweisen.

4 Fazit

Die vorliegende Arbeit befasst sich mit der Analyse von Verspätungen im öffentlichen Verkehr der Schweiz im Monat Mai 2024. Dabei werden Daten der konzessionierten Transportunternehmen mittels supervised und deep learning Modellen untersucht, um Ansätze zur Vorhersage von Abfahrtsverspätungen zu entwickeln.

Die Arbeit zeigt, dass sowohl das Random Forest Modell als auch das neuronale Netz geeignet sind, Verspätungen im öffentlichen Verkehr vorherzusagen, allerdings mit begrenzter Präzision. Die erzielten Ergebnisse sind besser als eine naive Schätzung basierend auf Durchschnittswerten, weisen jedoch noch Optimierungspotenzial auf. Die niedrigen R2-Werte deuten darauf hin, dass weitere nicht-lineare Zusammenhänge und möglicherweise weitere Einflussfaktoren berücksichtigt werden müssen, um die Modellleistung zu verbessern.

In zukünftigen Arbeiten sollten die Modelle weiter verbessert werden. Einerseits kann dies durch vertieftes Hyperparameter-Tuning und Cross-Validation geschehen, um verschiedene Modellansätze zu vergleichen und Optimierungen zu erzielen. Darüber hinaus wäre es sinnvoll, weitere relevante

Variablen einzubeziehen, die möglicherweise die Nicht-Linearität der Daten besser erfassen können. Beispielsweise könnten meteorologische Messdaten, die von MeteoSchweiz in einem Data Warehouse aufbereitet und zugänglich gemacht werden (wie Bodenstationen, Flugzeugmessungen, Sondierungen und Radar), wertvolle Zusatzinformationen liefern[4]. Mit erweitertem Feature Engineering könnten somit genauere Modelle entwickelt werden.

Zusätzlich könnten komplexere Modelle im Bereich der Deep Learning Architekturen angewendet werden, die zeitliche Abhängigkeiten besser erfassen können. Während in dieser Arbeit lediglich ein sequentielles neuronales Netz (keras.Sequential) verwendet wurde, könnten rekurrente neuronale Netze (RNNs) oder Long Short-Term Memory (LSTM) Netzwerke, die speziell für die Verarbeitung von Zeitreihen entwickelt wurden, signifikante Verbesserungen bieten. Diese Modelle können durch ihre Fähigkeit, zeitliche Muster und Abhängigkeiten zu erkennen, präzisere Vorhersagen liefern.

5 Quellenverzeichnis

[1] Abellán, Joaquín, and Andrés R. Masegosa. 2009. "An Experimental Study about Simple Decision Trees for Bagging Ensemble on Datasets with Classification Noise." In Symbolic and Quantitative Approaches to Reasoning with Uncertainty, edited by Claudio Sossai and Gaetano Chemello, 446–56. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-02906-6_39.

[2] "Bund stellt mehr Mittel für Betrieb und Erneuerung der Bahninfrastruktur bereit". o. J. Zugegriffen 16. Juni 2024. https://www.bav.admin.ch/bav/de/home/publikationen/medienmitteilungen.msg-id-101010.html.

[3] "Cookbook – Open-Data-Plattform Mobilität Schweiz". o. J. Zugegriffen 16. Juni 2024. https://opentransportdata.swiss/de/cookbook/actual-data/.

[4] "Datenintegration - MeteoSchweiz". o. J. Zugegriffen 22. Juni 2024. https://www.meteoschweiz.admin.ch/wetter/messsysteme/datenmanagement/datenintegration.html.

[5] Kingma, Diederik P., und Jimmy Ba. 2017. "Adam: A Method for Stochastic Optimization". arXiv. https://doi.org/10.48550/arXiv.1412.6980.

[6] Mohammed, Ammar, und Rania Kora. 2023. "A comprehensive review on ensemble deep learning: Opportunities and challenges". Journal of King Saud University - Computer and Information Sciences 35 (2): 757–74. https://doi.org/10.1016/j.jksuci.2023.01.014.

[7] Mohri, Mehryar, Afshin Rostamizadeh, und Ameet Talwalkar. 2012. Foundations of Machine Learning. Adaptive Computation and Machine Learning. Cambridge, Mass. London: The MIT Press.

[8] "RandomForestRegressor". 2024. Scikit-Learn. 2024. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html#.

[9] Redaktion. 2022. "Kundenbefragung 2021: Sehr zufriedene öV-Fahrgäste in der Nordwestschweiz". Bahnonline.ch (blog). 15. April 2022. https://www.bahnonline.ch/20349/kundenbefragung-2021-sehr-zufriedene-ov-fahrgaste-in-der-nordwestschweiz/.

[10] "ReLU Activation Function Explained". o. J. Built In. Zugegriffen 22. Juni 2024. https://builtin.com/machine-learning/relu-activation-function.

[11] "Scikit-learn: machine learning in Python — scikit-learn 1.3.2 documentation». o. J. Zugegriffen 9. Dezember 2023. https://scikit-learn.org/stable/.

[12] Song, Yan-yan, und Ying LU. 2015. "Decision tree methods: applications for classification and prediction". Shanghai Archives of Psychiatry 27 (2): 130–35. https://doi.org/10.11919/j.issn.1002-0829.215044.

[13] "TensorFlow". 2024. TensorFlow. 2024. https://www.tensorflow.org/.

[14] "Was ist Underfitting? – Data Basecamp". 2023. 13. September 2023. https://databasecamp.de/ki/underfitting.

A Appendix

A.1 Datensatz

Quelle

[15] "IST-Daten" – Open-Data-Plattform öV Schweiz≫. o. J. – Zugegriffen im Mai und Juni 2024. https://opentransportdata.swiss/de/group/actualdata-group

A.2 Code

Der für diese Arbeit verwendete Programmcode ist in zwei Jupyter Notebooks dokumentiert und online verfügbar:

1. Explorative Datenanalyse EDA & Random Forest Regressor Modell:
https://www.simonsahli.ch/code/verspaetung-a-eda-rf.ipynb
2. Modellierung TensorFlow:
https://www.simonsahli.ch/code/verspaetung-b-tf.ipynb