Vorhersage von Flugzeugpositionsdaten mittels Methoden des maschinellen Lernens

Sonja Heinze[1]

Universität Leipzig
sonja.heinze@studserv.uni-leipzig.de

Zusammenfassung. In dieser Arbeit geht es um die Ermittlung von Flugzeugpositionen, insbesondere der von Längengraden, anhand von ADB-S Sensordaten. Diese Arbeit untersucht dabei verschiedene Verfahren aus dem Bereich des maschinellen Lernens zur Lösung dieses Problems und evaluiert diese hinsichtlich der Fehlermetriken MAPE, MAE, MSE und RMSE. Mit steigender Komplexität der Verfahren konnte eine Verbesserung der Metriken erreicht werden. Das beste Verfahren stellt der Extra Trees Regressor mit einem MAPE von 3.589, MAE von 0.106 und einem RMSE von 0.290 dar.

Schlüsselwörter: Maschinelles Lernen, ADS-B, Empirische Daten

1 Fragestellung

Die Bestimmung präziser Flugzeugspositionsdaten ist ein sehr wichtiger Aspekt in der Flugsicherung und dem Flugverkehrsmanagement, um Sicherheit im Luftraum zu gewährleisten, sowie den Flugverkehr möglichst effizient zu gestalten. [3,4] Hierbei spielt das sogenannte „Automatic Dependent Surveillance – Broadcast“ (ADS-B) System zur Lokalisierung eine zunehmend große Rolle und es wird vorhergesagt, dass dieses System langfristig das klassische Radarsystem ablösen wird.[5] Das ADS-B System verwendet Satellitennavigationstechnologie, um die Positionsinformationen des Flugzeuges zu erfassen und sendet diese über einen Transponder unverschlüsselt aus. Diese Informationen können dann von Empfängersensoren am Boden empfangen und entschlüsselt werden.[6] Seit Januar diesen Jahres ist es für Flugzeuge im europäischen und amerikanischen Luftraum bereits verpflichtend mit einem ADS-B System ausgestattet zu sein. [3]

Der Einsatz dieser Technologie eröffnet neue Möglichkeiten für eine bessere, flächendeckendere Lokalisierung von Flugzeugen, allerdings bestehen durch die unverschlüsselte Übertragung auch Sicherheitsrisiken und Störanfälligkeiten, sodass die Übermittlung der Positionsdaten von Flugzeugseite her oftmals nicht fehlerfrei bzw, vollständig ist. Um diese Probleme zu entschärfen und die Standorte der Flugzeuge zu bestimmen, die noch keine Positionsmeldemöglichkeiten haben oder möglicherweise falsche Positionen melden sind ergänzende oder redundante, vom Flugzeug unabhängige Lokalisierungsmethoden erforderlich. [3,2]

Für eine flugzeugunabhängige Lokalisierung ist vor allem die Nutzung der Daten von den Empfängersensoren für ADS-B Daten am Boden aufgrund der Vielzahl von verfügbaren Daten von besonderem Interesse. Es finden unter anderem preisdotierte Wettbewerbe zur Positionsbestimmung anhand von diesen Daten statt. [2]

Das OpenSky Netzwerk ist in diesem Zusammenhang ein gemeinnütziges, gemeindebasiertes Empfängernetzwerk, das seit 2013 kontinuierlich Daten zur Flugverkehrsüberwachung sammelt und diese für akademische und institutionelle Forscher für die Luftverkehrsforschung zugänglich macht. [1] Bei den Daten, die dieser Arbeit zugrunde liegenden, handelt es sich entsprechend um ADS-B Daten des OpenSky Netzwerkes.

Während Multilateration zur Positionsbestimmung in der Luftverkehrssicherung und im Luftverkehrsmanagement ein etabliertes Verfahren ist [5,2] ist dieses Verfahren typischerweise auf vier Antennen mit bekannten Standorten, die zusammen arbeiten müssen, um den Standort zu bestimmen, angewiesen. Bei großen Distanzen in Gebieten mit großen Freiflächen und Ozeanen wird dieses Verfahren allerdings schnell ungenau.

Es stellt sich somit die Frage, inwiefern mittels der vorhandenen ADS-B Daten des Empfängernetzwerkes und mit Methoden des maschinellen Lernens aufgrund der Vielzahl von vorhandenen Daten eine präzisere Positionsbestimmung möglich ist. In dieser Arbeit sollen daher Modelle zur Vorhersage von Längengraden anhand von ADS-B Daten implementiert und ausgewertet werden.

2 Stand der Technik

Während traditionelle Lokalisierungsmethoden wie z.B. Multilateration schon seit langem bekannt sind, ist die Anwendung von Machine Learning noch ein neuer Ansatz zur Bestimmung von Flugzeugpositionsdaten. [2]

ADS-B Sensorempfängerdaten stellen zudem eine neue Herausforderung dar, da die meisten der kostengünstigen Sensoren nicht zeitsynchronisiert oder kalibriert sind. Hinzu kommen verschiedenen Arten von Rauschen z.B. Taktabweichungen, ungenaue Sensorpositionen oder unterbrochene Zeitstempel aufgrund von Softwarefehlern, die berücksichtigt werden müssen.[2]

Die erwartete Benchmark in dem aktuell laufenden Wettbewerb zur Bestimmung von Flugzeugpositionsdaten anhand von mit unserem Datensatz vergleichbaren Daten ist für die Entfernung zwischen Koordinatenpaaren ein RMSE von maximal 1000 m, was nach deren Aussagen bekanntlich mit klassischen Multilaterations- und Lokalisierungsalgorithmen auf synchronisierten, softwaredefinierten Funkempfängern erreichbar sei. [2]

3 Methodik

3.1 Datensatz

Der verwendete Datensatz enthalt ADS-B Daten zu insgesamt 1.951.877 Flugzeugpositionen, sowie ergänzend Informationen zu den Empfängersensoren in Form von CSV-Dateien. Jede Zeile in dem Messdatensatz stellt den Empfang eines Flugzeugpositionsberichts dar und enthält die folgenden Informationen:

- eindeutige Flugzeugkennung (ID)
- Unix-Zeitstempel, der angibt, wann die Nachricht von OpenSky empfangen wurde
- eindeutige Kennungen aller Sensoren, die dieses Signal empfangen haben
- Nanosekunden-Zeitstempel von jedem der Sensoren
- Signalstärkemessungen von jedem der Sensoren
- Position des Flugzeugs (Breitengrad, Längengrad, Höhe)
- barometrische Höhe des Flugzeugs

Zusätzlich werden für alle Sensoren die folgenden Metadaten zur Verfügung gestellt:

- eindeutige Sensorkennung
- Position des Sensors (Breitengrad, Längengrad, Höhe)
- Art der Hard- und Software

3.2 Preprocessing

Bevor der Datensatz für die Verfahren des maschinellen Lernens verwendet werden konnte, war zunächst eine Vorverarbeitung der Daten erforderlich.

Für Methoden des maschinellen Lernens ist es erforderlich, dass die Daten für das Training in numerischer Form vorliegen. Dies war bis auf bei den Informationen zum Sensortyp bereits der Fall, sodass lediglich für dieses Feature eine Transformation mithilfe des LabelEncoders von scikit-learn erforderlich war.

Die Informationen zu den Sensoren, die ein Signal empfangen haben, lagen im Datensatz zunächst in JSON-Array-Form in einer Zelle pro Zeile vor. Diese Informationen wurden aufgelöst in eigene Spalten. Da für etwa die Hälfte des Datensatzes lediglich zwei Sensormessungen vorlagen, wurden pro Sample die zwei stärksten Signalstärkemessungen als Grundlage für das Training behalten. Außerdem wurden der Datensatz mit den Metadaten zu den Sensoren angereichert.

Aus dem Trainingsdatensatz wurden folgende Features nicht für die maschinellen Lernverfahren verwendet: vorherzusagende Daten zu der Flugzeugposition, sowie die nichts-aussagenden Angaben zu ID, timeAtServer und Sensoren-IDs. Daraus ergaben sich folgende Features, die verwendet wurden:

Tabelle 1. Verwendete Features für maschinelles Lernen

Features		
aircraft	timestamp_1	timestamp_2
baroAltitude	signalstrength_1	signalstrength_2
	latitude_1	latitude_2
	longitude_1	longitude_2
	height_1	height_2
	type_1	type_2

3.3 Supervised Learning Methoden

Bei der Bestimmung von Positionsdaten handelt es sich im Sinne des maschinellen Lernens grundsätzlich um ein Regressionsproblem. Für die Implementierung und Bewertung von verschiedenen Verfahren des maschinellen Lernens wurden deshalb vier verschiedene Regressionsmodelle mit steigender Komplexität ausgewählt. Die Auswahl der Regressoren erfolgte hierbei in Anlehnung an Wang et al [7].

Als Baseline wurde ein **DummyRegressor** verwendet. Dies ist ein Regressor, der mit sehr einfachen Regeln Vorhersagen macht. Somit eignet er sich gut zum Vergleich mit anderen Regressoren als Baseline, da diese mindestens besser als der DummyRegressor sein sollten. Für diese Arbeit wurde als Methode der 'mean' verwendet, dieser sagt den Mittelwert der Zielwerte des Trainingssets voraus. [1]

Im zweiten Schritt wurde die Methode der **LinearRegression** anwendet. Dieser Ansatz ist immernoch verhältnismäßig einfach. Das Verfahren berücksichtigt aber bereits die vorliegenden Daten im Rahmen der Anpassung des Modells. [2]

Im dritten Schritt wurde ein **GradientBoostingRegressor** implementiert. Dieser baut ein additives Modell vorwärtsstufenweise auf. Es ermöglicht die Optimierung beliebiger differenzierbarer Verlustfunktionen. In jeder Stufe wird ein Regressionsbaum an den negativen Gradienten der gegebenen Verlustfunktion angepasst. Für das Training wurden die Default-Parameter von sci-kit learn verwendet. [3]

Zuletzt wurde noch vergleichsweise ein **ExtraTreesRegressor** implementiert. Hierbei handelt es sich um einen Meta-Schätzer, der eine Reihe von klassifizierenden Entscheidungsbäumen auf verschiedene Unterstichproben des Datensatzes anpasst und die Mittelwertbildung verwendet, um die Vorhersagegenauigkeit zu verbessern und das Overfitting zu kontrollieren. Dieses Verfahren ist eng verwandt mit dem RandomForestRegressor. [4]

[1] sklearn.dummy.DummyRegressor
[2] sklearn.linear_model.LinearRegression
[3] sklearn.ensemble.GradientBoostingRegressor
[4] sklearn.ensemble.ExtraTreesRegressor

4 Ergebnisse

Für die Evaluierung der verschiedenen Verfahren und deren Performance wurden die folgenden Metriken, die für die Bewertung bei Regressionsproblemen typisch sind, ausgewählt:

- Mean Absolute Percentage Error (MAPE)
- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- Root Mean Square Error (RMSE)

Die Tabelle 2 zeigt die Ergebnisse, die mittels der verschiedenen Verfahren erzielt werden konnten. Es lässt sich feststellen, dass die ausgewählten Verfahren mit zunehmender Komplexität bessere Vorhersagen treffen konnten. Für eine bessere Einschätzung und Einordnung in Bezug auf den aktuellen Stand der Technik wurde der RMSE von der Einheit *Längengrad* in *km* umgerechnet. Hierzu wurde die Distanz zwischen den Längengraden der Koordinaten 52.0 N 9.0 E und 52.0 N 10.0 E zugrundegelegt, da sich der Schwerpunkt der Daten in diesem Bereich befindet.

Tabelle 2. Fehlermetriken für Längengrad-Vorhersagen (Longitude)

	MAPE	**MAE**	**MSE**	**RMSE**	**RMSE in km**
Dummy (Mittelwert)	414.218	2.860	16.142	4.018	273,224
Lineare Regression	52.147	0.812	1.127	1.061	72,148
Gradient Boosting Regressor	26.967	0.661	0.769	0.877	59,636
Extra Trees Regressor	3.589	0.106	0.084	0.290	19,72

5 Diskussion

Insgesamt sind die Ergebnisse dieser Arbeit als zufriedenstellend einzuordnen. Mit einfachen Implementierungen von Verfahren des maschinellen Lernens konnten mit steigender Komplexität der Verfahren zunehmend Verbesserungen erzielt werden.

Im Vergleich zum geforderten Mindeststandard des aktuell ausgeschriebenen Wettbewerbes sind die Implementierungen allerdings gemessen an den ermittelten Metriken noch nicht wettbewerbsfähig. Hierbei ist jedoch zu berücksichtigen, dass das zugrunde liegende Pre-Processing auf ein Minimum begrenzt wurde und die verschiedenen Verfahren des maschinellen Lernens, die auch bereits von Wang et al [7] getestet wurden, zunächst lediglich mit den Default-Einstellungen von sci-kit learn trainiert wurden. Desweiteren werden bei der Vorhersage und Berechnung der Fehlermetriken in unserem Ansatz alle Werte, d.h. auch Ausreißer, mitberücksichtigt. Die besten Ansätze im aktuellen Wettbewerb hingegen schließen Ausreißer grötenteils aus.

Vermutlich könnten die Metriken verbessert werden, wenn auc im Rahmen des Pre-Processings dieser Art eine stärkere Säuberung der Daten erfolgen würde. Zudem bieten auch Anpassungen der Parameter bei dem Gradient Boosting Regressor und Extra Trees Regressor noch Verbesserungspotential.

6 Schlussfolgerungen

Im Rahmen dieser Arbeit hat sich gezeigt, dass das Pre-Processing ein nicht zu vernachlässigender und wichtiger Teil im Prozess des maschinellen Lernens ist. Gleichzeitig war dieser Teil auch am zeitintensivsten, da sich mit den Daten vertraut gemacht werden musste und die Aufbereitung, sowie Auswahl der Features nicht trivial sind. Für ein bestmögliches Pre-Processing ist außerdem Domänenwissen unerlässlich.

Es bleibt offen, inwieweit sich durch ein differenzierteres Pre-Processing bessere Ergebnisse erzielen lassen oder ob der Trade-Off nicht doch zu groß ist und stattdessen ein Parameter-Tuning der Verfahren zunächst vorgezogen werden sollte.

Es wäre für weiterführende Arbeiten sicherlich interessant zu untersuchen, inwiefern sich durch Anpassungen der Parameter der Verfahren *Gradient Boosting Regressor* und *Extra Trees Regressor* noch bessere Ergebnisse erzielen lassen und wie groß hierbei der Trade-Off zwischen Komplexität und Performance ist.

Literatur

1. Opensky Network — About. https://opensky-network.org/about/about-us, zuletzt abgerufen am 30.07.2020.
2. CYD Campus Aircraft Localization Competition — Challenges, 2020. https://www.aicrowd.com/challenges/cyd-campus-aircraft-localization-competition, zuletzt abgerufen am 30.07.2020.
3. Damilola Adesina, Olutobi Adagunodo, Xishuang Dong, and Lijun Qian. Aircraft location prediction using deep learning. In *MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM)*, pages 127–132. IEEE, 2019.
4. Kaeye Dästner, Elke Schmid, Bastian von Haßler zu Roseneckh-Köhler, and Felix Opitz. Learning from ads-b data for real-time radar applications. In *2019 20th International Radar Symposium (IRS)*, pages 1–10. IEEE, 2019.
5. Stijn Meijer, Veelasha Moonsamy, and Lejla Batina. Secure location verification for ads-b. 2016.
6. J Sun, J Ellerbroek, and JM Hoekstra. Large-scale flight phase identification from ads-b data using machine learning methods. In *7th International Conference on Research in Air Transportation*, 2016.
7. Zhengyi Wang, Man Liang, and Daniel Delahaye. Automated data-driven prediction on aircraft estimated time of arrival. *Journal of Air Transport Management*, 88:101840, 2020.