

Vorhersage der Datenübertragungsraten und eNodeB-Verbindungsauern in LTE-Netzen

Christian Peters

31. Dezember 2020

Veranstaltung: Fallstudien II
Dozent: Prof. Dr. Markus Pauly
Gruppe: Laura Kampmann, Christian Peters, Alina Stammen

Inhaltsverzeichnis

1	Einleitung	1
2	Problemstellung	1
2.1	Datenbeschreibung	1
2.2	Zielsetzungen	2
2.2.1	Task I – Vorhersage der Datenübertragungsraten	2
2.2.2	Task II – Vorhersage der eNodeB-Verbindungsdauern	3
3	Methodik	3
3.1	Extreme Gradient Boosting	3
3.1.1	Ausgangssituation	3
3.1.2	Zielfunktion	3
3.1.3	Training	4
3.2	Regression mit ARMA-Fehlern	4
3.3	Validierung	4
4	Ergebnisse	4
5	Zusammenfassung	4
	Literatur	5

1 Einleitung

In dieser Arbeit geht es um die Grundlagen der Datenwissenschaften. Wir beschäftigen uns speziell mit dem Thema XYZ, welches sehr vielseitig ist und neben der theoretischen Tiefe auch viele praktische Anwendungen hat.

2 Problemstellung

2.1 Datenbeschreibung

Die vorliegenden Daten wurden im Zuge mehrerer Testfahrten durch das deutsche LTE-Netz der Netzbetreiber O2, T-Mobile und Vodafone im Raum Dortmund erhoben [2]. Die Testfahrten verliefen über vier zuvor festgelegte Routen, welche sich hinsichtlich der Art ihrer Umgebung unterscheiden:

- **Campus:** Direkte Umgebung der TU Dortmund, Routenlänge 3km.
- **Urban:** Stadtbereich, Routenlänge: 3km.
- **Suburban:** Vorstadtbereich, Routenlänge: 9km.
- **Highway:** Autobahn, Routenlänge: 14km.

Jede dieser Messfahrten wurde zehnmal wiederholt. Hierbei wurden sowohl passive Messungen der Netzqualität mithilfe verschiedener Indikatoren, als auch aktive Messungen der Up- und Downloadraten durchgeführt. Die Messungen der Datenübertragungsraten wurden alle 10s vollzogen, die Messungen der passiven Indikatoren alle 1s. Um die Datenübertragungsraten erfassen zu können, wurden Datenpakete zufälliger Größe von 0.1, 0.5, 1, ..., 10 MB an einen Server zur Messung übertragen. Die insgesamt erhobenen Variablen seien in der folgenden Auflistung kurz beschrieben:

- **RSRP:** *Reference Signal Received Power* gibt die Empfangsstärke eines Referenzsignals an. Je höher der Wert, desto besser ist der Empfang.
- **RSRQ:** *Reference Signal Received Quality* ist ein weiterer Indikator für die Verbindungsqualität. Er wird unter anderem aus dem RSRP berechnet und kann vom Funkmast verwendet werden, um die Notwendigkeit eines Funkmastwechsels abschätzen zu können.
- **SINR:** *Signal-to-interference-plus-noise Ratio* gibt das Verhältnis des tatsächlichen Signals zum Rauschen oder anderen Störeinflüssen an.
- **CQI:** *Channel Quality Indicator* ist ein Indikator, welcher Aufschluss über die Qualität des Übertragungskanals gibt.
- **TA:** *Timing Advance* gibt den Zeitversatz an, der zur Synchronisation zwischen Up- und Downlink verwendet wird. Damit gibt er indirekt Aufschluss über die Entfernung zum Funkmast.

- **f:** Gibt die *Frequenz* des LTE-Signals an.
- **Velocity:** Die Geschwindigkeit, mit der sich das Messgerät fortbewegt.
- **Cell ID:** Identifiziert eine Zelle im LTE-Netzwerk. Nicht zu verwechseln mit der eNodeB-ID, welche einen Funkmast identifiziert. Ein Funkmast kann mehrere Zellen haben.
- **Payload Size:** Die Größe des übertragenen Datenpakets zur Ermittlung der Datenübertragungsrate.
- **Data Rate:** Die gemessene Datenübertragungsrate. Es werden sowohl Upload- als auch Downloadraten gemessen.

Die Messungen aller Testfahrten lassen sich insgesamt zu vier verschiedenen Datensätzen zusammenfassen, welche auch später zur Bearbeitung der Projektziele verwendet werden:

- **Context:** Dieser Datensatz enthält die sekundlich durchgeführten passiven Messungen der Netzwerkindikatoren. Insgesamt enthält dieser Datensatz 68334 Messungen.
- **Cells:** Enthält Messungen des RSRP und RSRQ zu den Nachbarzellen der aktuell verbundenen Zelle. Dieser Datensatz enthält insgesamt 93443 Messungen.
- **Upload:** Dieser Datensatz enthält die Messungen der Upload-Raten, welche alle 10s durchgeführt werden zuzüglich der Indikatoren aus dem Context-Datensatz zum entsprechenden Zeitpunkt. Insgesamt enthält dieser Datensatz 6180 Messungen.
- **Download:** Analog zum Upload-Datensatz, nur dass hier die gemessenen Download-Raten erfasst wurden. Dieser Datensatz enthält insgesamt 6516 Messungen.

2.2 Zielsetzungen

2.2.1 Task I – Vorhersage der Datenübertragungsraten

In [2] wurde ein neuartiger Ansatz der datengetriebenen Simulation von Netzwerken (Data-driven Network Simulation, *DDNS*) vorgestellt, welcher darauf basiert, dass durch datengetriebene Modelle möglichst realitätsnahe Simulationen von Netzwerken erzeugt werden sollen. Ein Aspekt dieser Modelle besteht darin, dass Up- und Downloadraten abhängig von den übrigen Netzwerkindikatoren möglichst realistisch modelliert werden müssen. Hierzu werden Prädiktionsmodelle benötigt, welche diese Datenübertragungsraten entsprechend vorhersagen können.

Das erste Ziel dieses Projektes ist es nun, verschiedene Arten von Prädiktionsmodellen im Hinblick auf diese Problemstellung anzuwenden, und die Güte dieser Verfahren zu untersuchen. Hierbei wird auch analysiert, ob sich das Verhalten der Modelle bezüglich der verschiedenen Netzbetreiber und Testfahrtszenarien unterscheidet. Weiterhin wird auch die Relevanz der verwendeten Kovariablen untersucht.

2.2.2 Task II – Vorhersage der eNodeB-Verbindungsdauern

Bei den ersten Einsätzen von DDNS in [2] hat sich gezeigt, dass es oft zu großen Vorhersagefehlern kommt, wenn der Funkmast gewechselt wird (in der Fachsprache heißen LTE-Funkmasten auch *eNodeB*). Eine Idee, um diesem entgegenzuwirken ist, den Zeitpunkt des eNodeB-Wechsels vorherzusagen. Kennt man diesen Zeitpunkt, könnte man diese Information im nächsten Schritt dazu verwenden, um die Prädiktionsmodelle zu verbessern.

Das zweite Ziel dieses Projektes ist also, die Restdauer der bestehenden Verbindung zu einer eNodeB und damit indirekt auch den Wechselzeitpunkt zur nächsten eNodeB vorherzusagen. Auch hier wird die Güte des eingesetzten Prädiktionsmodells anschließend analysiert und das Verhalten des Modells bezüglich der verschiedenen Netzbetreiber und Einsatzszenarien, sowie die Relevanz der verwendeten Kovariablen untersucht.

3 Methodik

3.1 Extreme Gradient Boosting

Extreme Gradient Boosting ist ein Verfahren aus dem Bereich des maschinellen Lernens, welches sich in den letzten Jahren einer immer größeren Beliebtheit erfreut hat [1]. Die Grundlegende Funktionsweise dieses Verfahrens sei im Folgenden kurz beschrieben.

3.1.1 Ausgangssituation

Wir gehen davon aus, dass wir über einen Trainingsdatensatz $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$ der Größe $|\mathcal{D}| = n$ verfügen, welcher aus den beobachteten Messungen $\mathbf{x}_i \in \mathbb{R}^m$ und der Zielgröße $y_i \in \mathbb{R}$ besteht, deren Wert wir vorhersagen wollen.

Das Ziel des Tree Boosting ist es, den Wert von y_i durch ein Ensemble von Entscheidungsbäumen (CART) vorherzusagen:

$$\hat{y}_i = \phi(\mathbf{x}_i) = \sum_{k=1}^K f_k(\mathbf{x}_i), \quad f_k \in \mathcal{F} \quad (1)$$

Hierbei ist \mathcal{F} die Klasse der besagten Entscheidungsbäume, welche in jedem ihrer T Blätter einen konstanten Wert vorhersagen: $\mathcal{F} = \{f(\mathbf{x}) = w_{q(\mathbf{x})}\}$, wobei $q : \mathbb{R}^m \rightarrow T$ eine Funktion ist, die der Beobachtung \mathbf{x} eines der T Blätter zuordnet und $w \in \mathbb{R}^T$ der Vektor der Blattvorhersagen (Gewichte) des Baumes ist.

3.1.2 Zielfunktion

Die Zielfunktion, welche während des Trainings zur Anpassung des Modells minimiert wird, setzt sich wie folgt zusammen:

$$\mathcal{L}(\phi) = \sum_{i=1}^n l(\hat{y}_i, y_i) + \sum_{k=1}^K \Omega(f_k) \quad (2)$$

Hierbei ist l eine differenzierbare und konvexe Verlustfunktion, welche Aufschluss über die Güte der Vorhersage \hat{y}_i liefert. Ein Beispiel ist der quadratische Fehler, welcher durch $l(\hat{y}_i, y_i) = (\hat{y}_i - y_i)^2$ gegeben ist. Die Funktion Ω ist ein sogenannter Regularisierungs- oder Strafterm und ist wie folgt definiert:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2 \quad (3)$$

Das Ziel von Ω ist es, eine zu hohe Komplexität der einzelnen Entscheidungsbäume in der Optimierung zu bestrafen und somit während des Trainings einfachere Bäume zu bevorzugen. Dies geschieht mit dem Hintergedanken, eine Überanpassung des Modells an die Trainingsdaten verhindern zu wollen. Der Parameter γ bestraft hierbei die Anzahl der Blätter T eines Entscheidungsbaumes und der Parameter λ bestraft zu große Gewichte in den einzelnen Blättern.

3.1.3 Training

Das Grundprinzip des Boosting ist es, die Ensemble Modelle additiv nach dem Greedy-Prinzip zu trainieren. Dies funktioniert hier so, dass die einzelnen Entscheidungsbäume nicht alle gleichzeitig angepasst werden, sondern nach und nach zum Ensemble hinzugefügt werden. Jeder Baum, welcher in einem Schritt hinzugefügt wird, wird so trainiert, dass er die Zielfunktion soweit wie möglich minimiert.

Wenn im Optimierungsschritt t also der Entscheidungsbaum f_t zum Ensemble hinzugefügt wird, ergibt sich die folgende Verlustfunktion, welche durch f_t minimiert werden soll:

$$\mathcal{L}^{(t)} = \sum_{i=1}^n l(\hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i), y_i) + \Omega(f_t) \quad (4)$$

Die Regularisierungsterme $\sum_{k=1}^{t-1} \Omega(f_k)$ der bereits zum Ensemble hinzugefügten Bäume wurden hierbei weggelassen, da sie im Zuge der Optimierung in Schritt t nicht mehr verändert werden können.

3.2 Regression mit ARMA-Fehlern

3.3 Validierung

4 Ergebnisse

Dies und das...

5 Zusammenfassung

Dies und das...

Literatur

- [1] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016.
- [2] B. Sliwa and C. Wietfeld. Data-driven network simulation for performance analysis of anticipatory vehicular communication systems. *IEEE Access*, 7:172638–172653, 2019.