Vorhersage der Datenübertragungsraten und eNodeB-Verbindungsdauern in LTE-Netzen

Christian Peters

3. Januar 2021

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

Inhaltsverzeichnis

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 [3]. 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 sekündlich 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 [3] 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 [3] 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 Zeitpunk 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 Wechselzeitpunk 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 Allgemeine Vorgehensweise

3.2 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.

3.2.1 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:

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

Hierbei ist $\mathcal{F}$ die Klasse der besagten Entscheidungsbäume, welche in jedem ihrer T Blätter einen konstanten Wert vorhersagen: $\mathcal{F} = \{f(\mathbf{x}) = w_{q(x)}\}$, wobei $q : \mathbb{R}^m \to T$ 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.

3.2.2 Zielfunktion

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

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

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:

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

Das Ziel von Ω ist es, eine zu hohe Komplexität der einzelnen Entscheidungsbäume in der Optimierung zu bestrafen und somit während des Trainings simplere 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.

3.2.3 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:

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

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.

Beim Extreme Gradient Boosting wird $\mathcal{L}^{(t)}$ nun im Punkt $\hat{y}_i^{(t-1)}$ durch ein Taylor-Polynom 2. Grades approximiert, welches sich analytisch minimieren lässt. Streicht man alle konstanten Terme, welche für die Minimierung keine Rolle spielen, erhält man so die folgende Taylor-Approximation:

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \tag{5}$$

Hierbei sind $g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$ und $h_i = \partial^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$ die erste und zweite partielle Arbleitung der Verlustfunktion l.

Wie in [1] gezeigt wurde, lassen sich dann die optimalen Gewichte $w_j^*, j = 1 \dots T$ für eine gegebene Baumstruktur q durch analytische Minimierung von $\tilde{\mathcal{L}}^{(t)}$ berechnen. Die Bestimmung einer optimalen Baumstruktur q hingegen ist rechnerisch durch Enumeration aller erdenklichen Möglichkeiten im Normalfall keine Option. Daher wird analog zum CART-Algorithms ein Greedy-Verfahren eingesetzt, welches den Baum durch sukzessives Hinzufügen neuer Verzweigungen aufbaut. Jede neue Verzweigung wird dabei so gewählt, dass der Wert von $\tilde{\mathcal{L}}^{(t)}$ durch die Bestimmung der optimalen Gewichte zum aktuellen Baum soweit wie möglich minimiert wird. Der Regularisierungsterm $\Omega(f_t)$ verhindert dabei direkt durch seine Anwesenheit in $\tilde{\mathcal{L}}^{(t)}$, dass die neue Baumstruktur zu komplex wird.

3.3 Regression mit ARMA-Fehlern

3.4 Validierung

Die Aufgabe der Modellvalidierung ist es, Aussagen darüber zu treffen, wie sich ein trainiertes Modell auf neuen und ungesehenen Daten verhalten wird. Ein bekanntes Verfahren zur Modellvalidierung ist die k-fache Kreuzvalidierung [2], welche auch in [3] zum Einsatz gekommen ist. Hierbei wird der gesamte Datensatz zunächst zufällig in k gleich große Partitionen unterteilt, um im Anschluss das Modell jeweils auf $k-1$ Partitionen zu trainieren und die übrige Partition zum testen zu verwenden. Dies wird solange wiederholt, bis jede der k Partitionen genau einmal zum testen verwendet wurde. Obwohl dieses Verfahren sehr weit verbreitet ist, gibt es in der vorliegenden Situation jedoch Anhaltspunkte dafür, dass sich die k-fache Kreuzvalidierung möglicherweise als problematisch erweisen könnte.

In Abbildung 1 ist eine der durchgeführten Messfahrten einmal beispielhaft zu sehen. Man erkennt sofort, dass es sich bei den gemessenen Daten offenbar um eine Zeitreihe handelt. Würde man in dieser Situation eine k-fache Kreuzvalidierung einsetzen, bei der die Daten zufällig partitioniert werden, so würde der zeitliche Zusammenhang zwischen den Beobachtungen dadurch verloren gehen. Es wäre also fraglich, ob durch diese Art der Validierung verlässliche Aussagen über das Modellverhalten auf zukünftig erhobenen Messdaten getroffen werden können. Aus diesem Grund wurde in diesem Projekt ein eigenes Validierungsverfahren eingesetzt, welches speziell auf die vorliegende Situation zugeschnitten wurde.

In Abbildung 2 ist die in diesem Projekt eingesetzte Validierungsmethode einmal schematisch dargestellt. Wie bereits beschrieben, besteht der gesamte Datensatz an Messungen für einen Netzbetreiber aus zehn einzelnen Messfahrten für jedes der Szenarien *campus*, *highway*, *suburban* und *urban*. Jeder dieser Fahrten kann also chronologisch eine Nummer von 1-10 zugewiesen werden, welche zusammen mit dem Szenario eine Fahrt eindeutig identifiziert. Im hier eingesetzten Validierungsverfahren wurde nun zunächst der gesamte Datensatz in zwei Teile aufgeteilt. Der erste Teil besteht aus den Fahr-

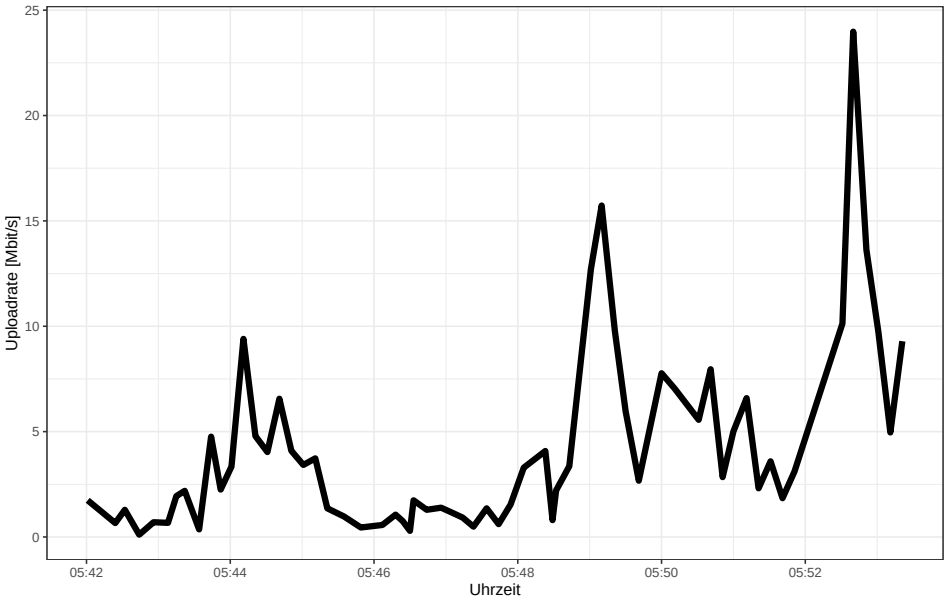

Abbildung 1: Die erste Messfahrt auf der Autobahn für den Netzbetreiber Vodafone am 12.12.2018.

ten 1-7, der zweite Teil besteht aus den Fahrten 8-10. In der Trainingsphase und beim Parametertuning kommt ausschließlich der erste Teil der Fahrten 1-7 zum Einsatz. So wird sichergestellt, dass das Modell beim Training keine Informationen aus zukünftigen Fahrten mit einbeziehen kann, wie es beispielsweise bei der k-fachen Kreuzvalidierung der Fall wäre. Fahrten 8-10 werden also ausschließlich zur Modellvalidierung eingesetzt.

Für eine geeignete Wahl der Hyperparameter werden auf dem Trainingsdatensatz, also Fahrten 1-7, verschiedene Parameterbelegungen getestet und evaluiert. Zur Evaluation einer Parameterkombination kommt hierbei, wie sich in Abbildung 2 ebenfalls erkennen lässt, eine Art Kreuzvalidierung für Zeitreihen zum Einsatz. Hierbei wird der Trainingsdatensatz sukzessive um eine Fahrt erweitert und immer auf der nächsten Fahrt getestet. Die ermittelten Gütemaße werden dann im Anschluss über die Testdatensätze hinweg gemittelt.

4 Ergebnisse

4.1 Vorhersage der Datenübertragungsraten

4.1.1 Extreme Gradient Boosting

Die Out-of-Sample Vorhersagen der Datenübertragungsraten des Extreme Gradient Boosting Modells finden sich in Abbildung 3. Man erkennt, dass sich die Verteilungen der Datenraten mitunter stark je nach Szenario und Anbieter unterscheiden. Vor allem fällt auf, dass der Anbieter Vodafone eine wesentlich höhere Variation in den Download-Raten vorweist, alls die übrigen Anbieter. Insgesamt lassen sich anhand dieser Vorhersagen allerdings keine systematischen Unregelmäßigkeiten erkennen.

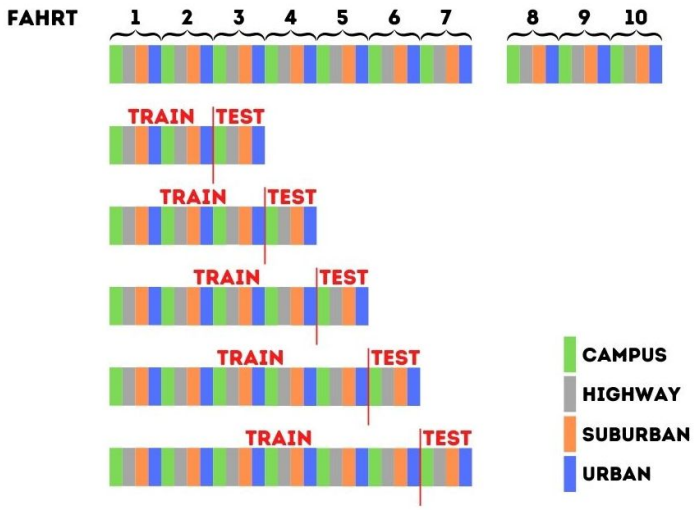

Abbildung 2: Das eingesetzte Verfahren zur Modellvalidierung.

4.1.2 Regression mit ARMA-Fehlern

Für die lineare Regression mit ARMA-Fehlern finden sich die Out-of-Sample Vorhersagen in Abbildung 4. Vergleicht man diese mit den Vorhersagen des Extreme Gradient Boosting, so fallen hier schon etwas stärkere systematische Abweichungen ins Auge. Beispielsweise scheint es so, dass im *urban* Szenario für den Anbieter O2 höhere Upload-Raten systematisch unterschätzt und niedrigere Upload-Raten systematisch überschätzt werden. Dies gibt einen ersten Aufschluss darüber, dass das Modell möglicherweise nicht expressiv genug sein könnte, um die Zusammenhänge in den Daten zu erfassen.

4.1.3 Modellvergleich

Die betrachteten Kennzahlen R^2 und MAE wurden in Abbildung 5 einander gegenübergestellt. Man erkennt sofort, dass Extreme Gradient Boosting für jeden Anbieter die besseren Werte liefert, als die lineare Regression mit ARMA-Fehlern. Dies würde die Vermutung bestätigen, dass das Modell der ARMA-Regression nicht in der Lage ist, sämtliche Zusammenhänge in den Daten zu erfassen und das somit das Extreme Gradient Boosting besser zur Vorhersage der Datenübertragungsraten geeignet ist.

Die Relevanz der einzelnen Kovariablen für die beiden Prädiktionsmodelle wurde in Abbildung 6 dargestellt. Diese wurden bei der linearen Regression mit ARMA-Fehlern durch die normierte absolute Größe der Modellkoeffizienten ermittelt. Beim Extreme Gradient Boosting kam die Permutationsmethode zum Einsatz, die Ergebnisse daraus wurden zur besseren Vergleichbarkeit ebenfalls normiert.

Hierbei fällt auf, dass der Variable *Payload* in jeder Situation eine hohe Relevanz besitzt. Für Exterme Gradient Boosting wird dies sogar besonders deutlich, hier erhält *Payload* in jeder Situation den höchsten Wichtigkeitswert.

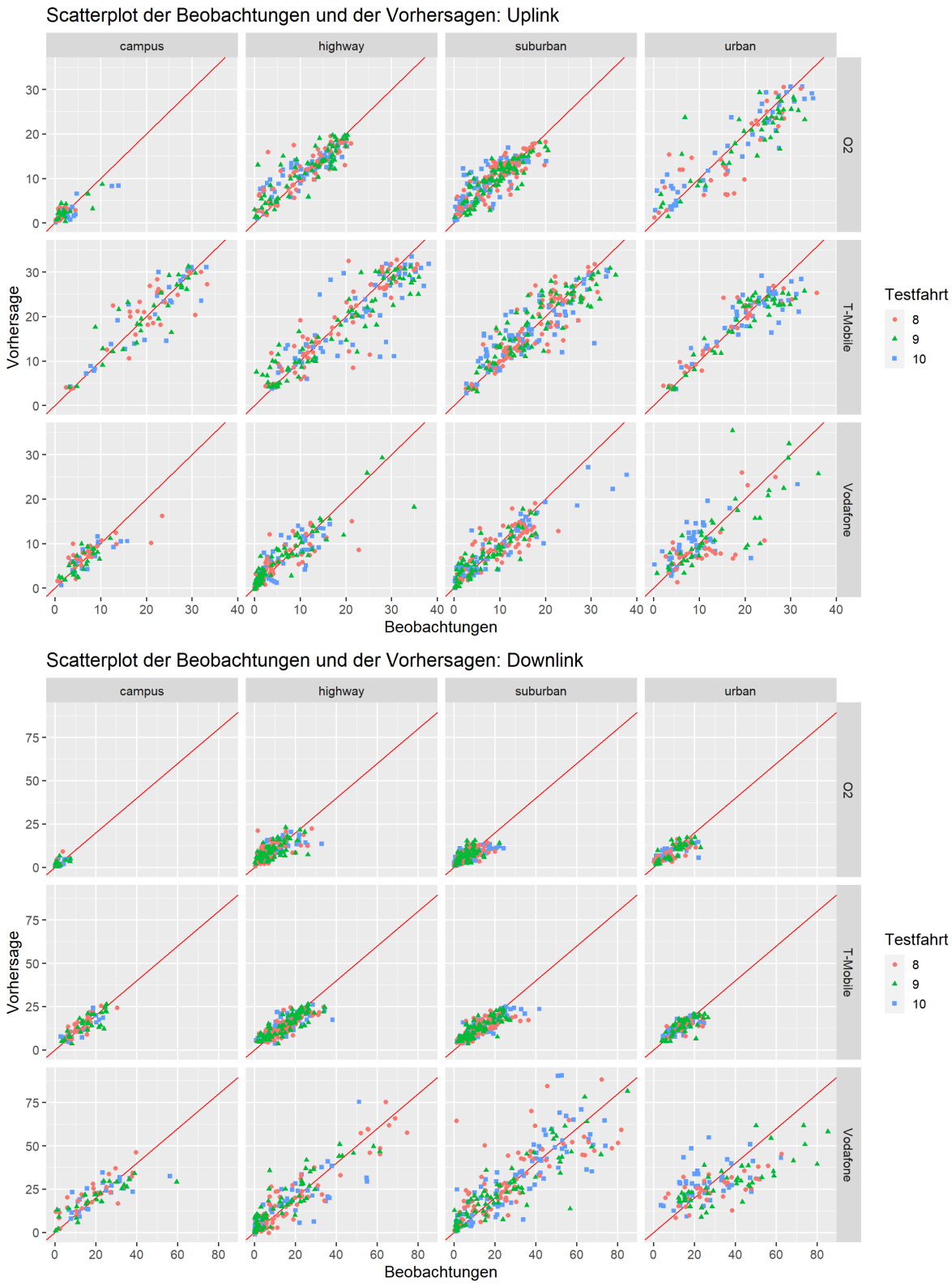

Abbildung 3: Out-of-Sample Vorhersagen der Datenraten für Extreme Gradient Boosting.

Abbildung 4: Out-of-Sample Vorhersagen der Datenraten für die Regression mit ARMA-Fehlern.

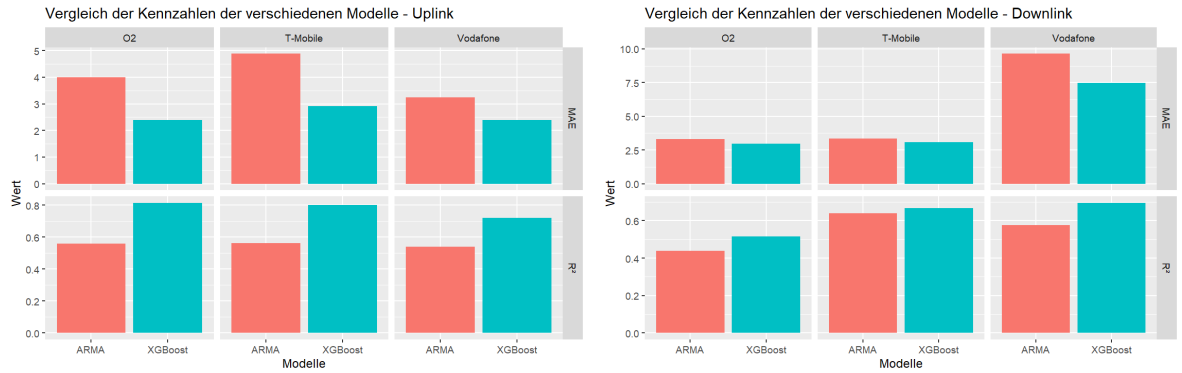

Abbildung 5: Vergleich der Kennzahlen für die Prädiktion der Upload- und Download-Raten.

4.2 Vorhersage der eNodeB-Verbindungsdauern

Zur Vorhersage der eNodeB-Verbindungsdauern wurde ausschließlich das Extreme Gradient Boosting Modell eingesetzt. Die Out-of-Sample Vorhersagen hierzu finden sich in Abbildung 7.

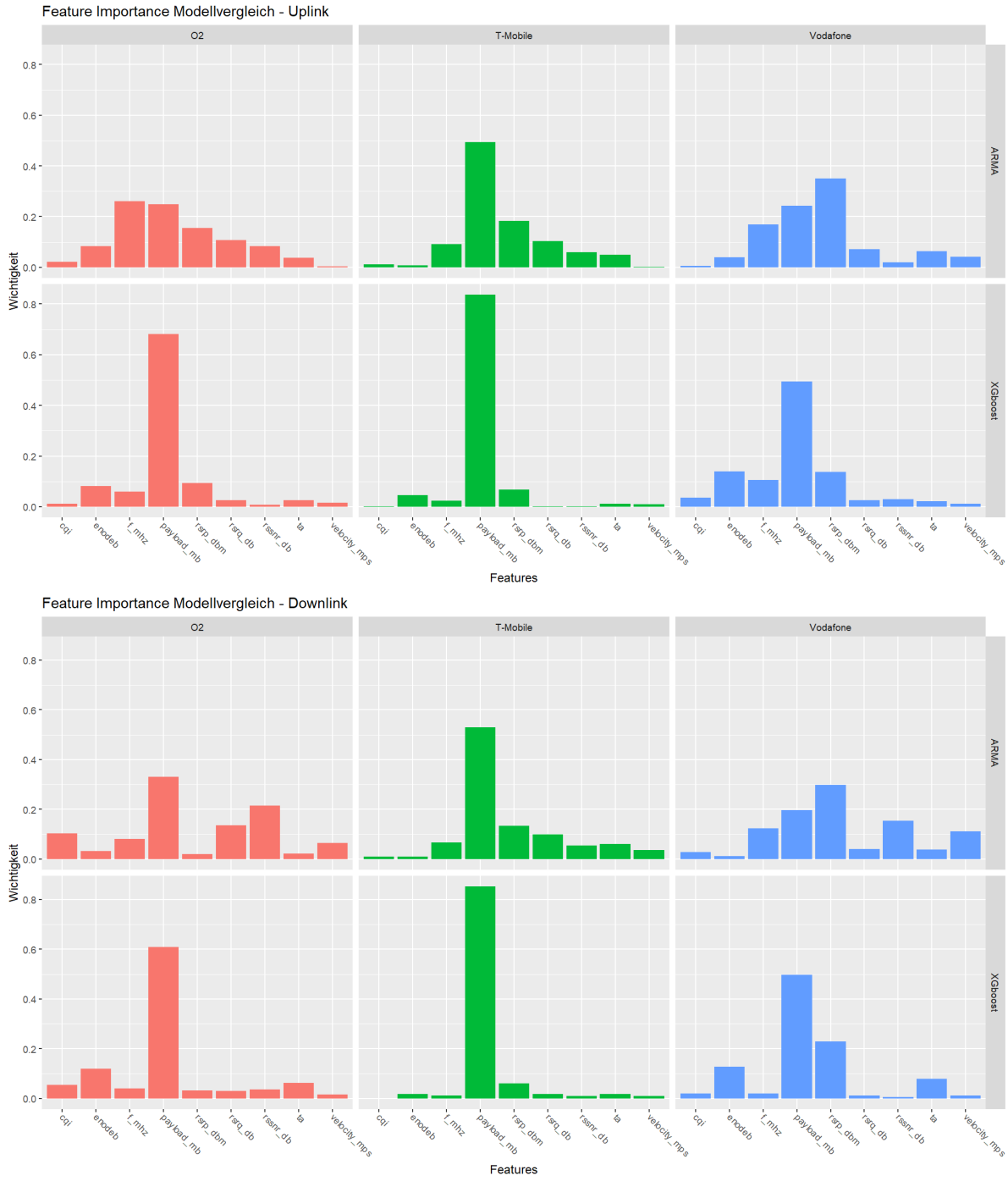

Abbildung 6: Relevanz der Kovariablen bezüglich der beiden Modelle zur Datenratenprädiktion.

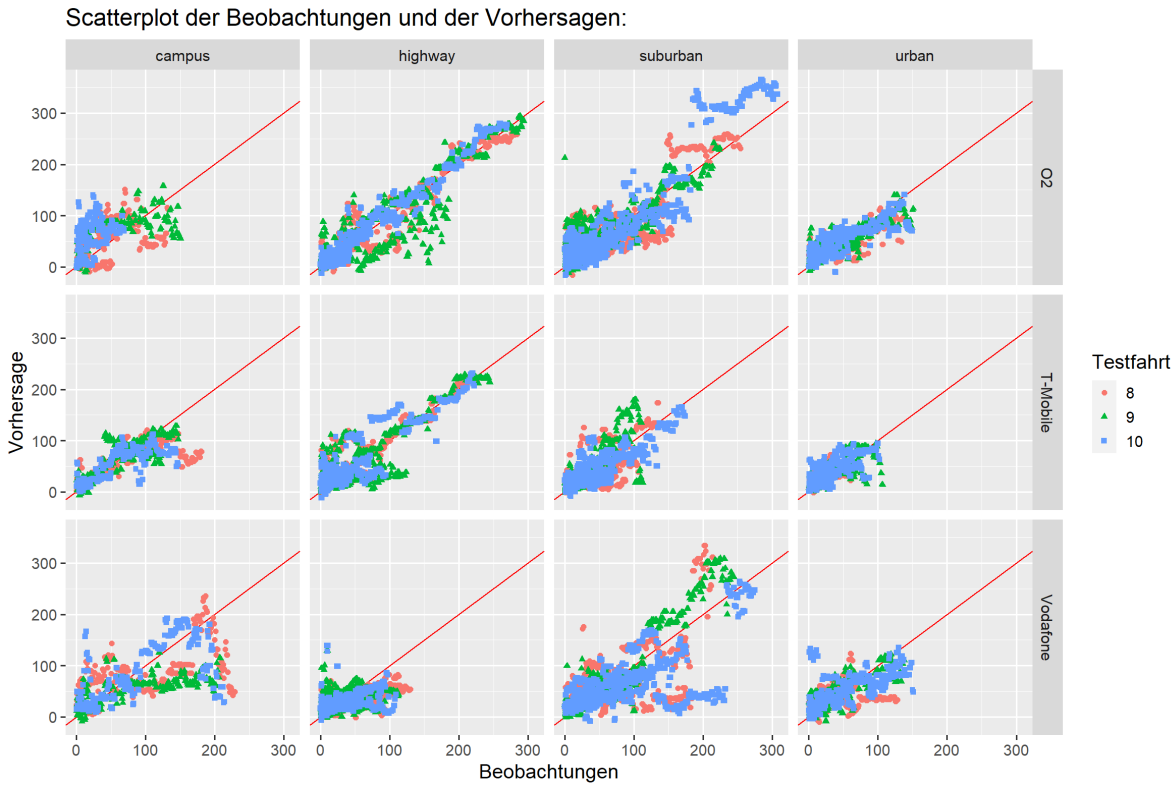

Abbildung 7: Out-of-Sample Vorhersagen der eNodeB-Verbindungsdauern.

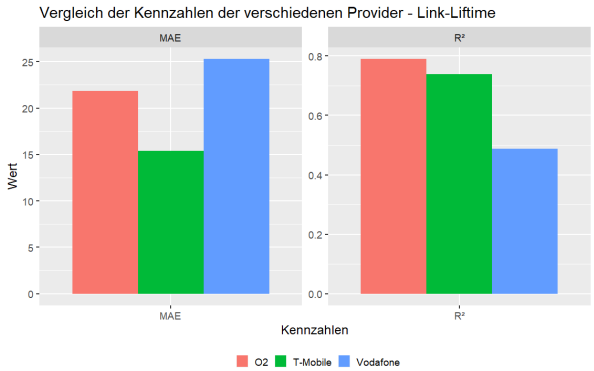

Abbildung 8: Kennzahlen Link-Lifetime.

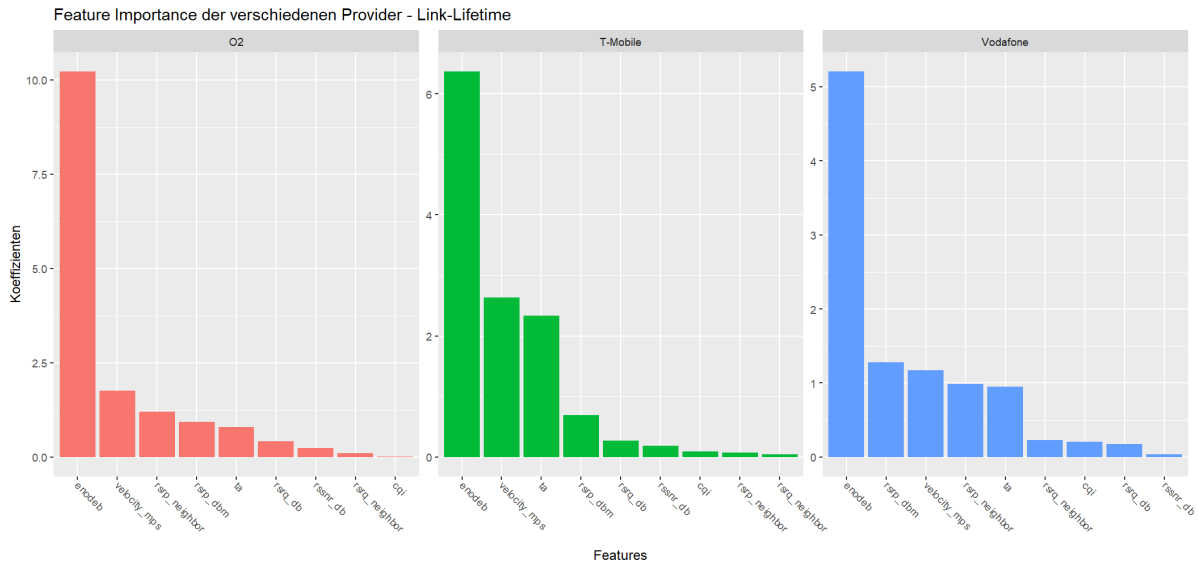

Abbildung 9: Feature Importance Link-Lifetime.

5 Zusammenfassung

Dies und das...

Literatur

[1] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016.

[2] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2009.

[3] B. Sliwa and C. Wietfeld. Data-driven network simulation for performance analysis of anticipatory vehicular communication systems. *IEEE Access*, 7:172638–172653, 2019.