

Fallstudien II

Laura Kampmann, Christian Peters, Alina Stammen

18. Dezember 2020

1. Task I - Vorhersage der Datenrate

Extreme Gradient Boosting

Regression mit ARMA-Fehlern

Modellvergleich

2. Task II - Handover Vorhersage und Link Lifetime

Feature Importance

Task I - Vorhersage der Datenrate

Extreme Gradient Boosting

Intro

Features

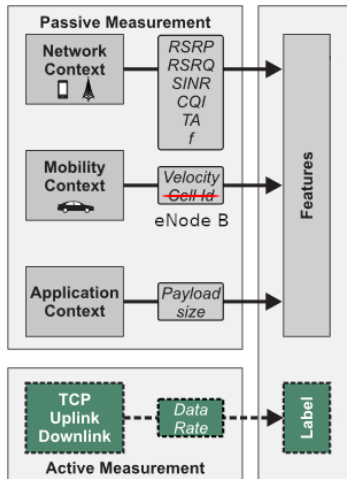


Abbildung 1: Modellfeatures [3].

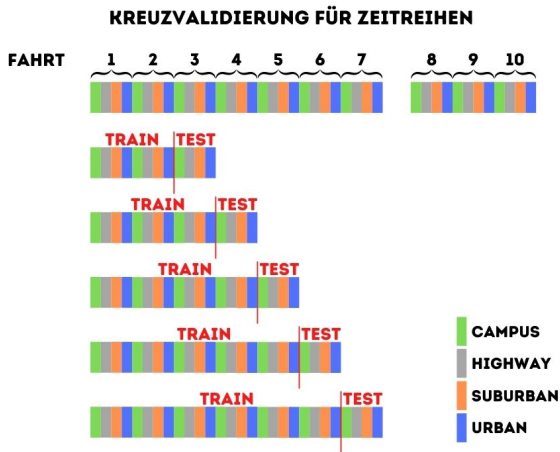


Abbildung 2: Einteilungen in Trainings- und Testdatensätze bei der Kreuzvalidierung für Zeitreihen.

Suchraum der Hyperparameter:

- Anzahl der Boosting Runden $n_rounds \in [100, 1000]$
- „Shrinkage“ Faktor (Lernrate) $\eta \in [0.01, 1]$
- Strafterm für Anzahl Baumblätter $\gamma \in [0, 10]$
- Strafterm für Vorhersagen der Baumblätter $\lambda \in [0, 10]$

⇒ Randomisierte Gittersuche

- 20 Gitterpunkte in jeder Dimension
⇒ Insgesamt $20^4 = 160.000$ Gitterpunkte
- Ausgewertet an 50 zufälligen Stellen
- Berechnung des MAE mit Zeitreihenkreuzvalidierung für die Fahrten 1-7

Out-of-Sample Vorhersagen Upload

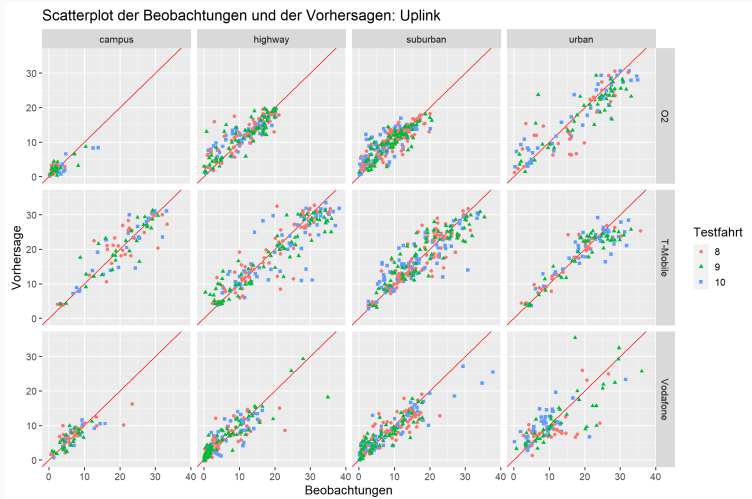


Abbildung 3: XGBoost Out-of-Sample Vorhersagen der Upload-Rate

Out-of-Sample Vorhersagen Download

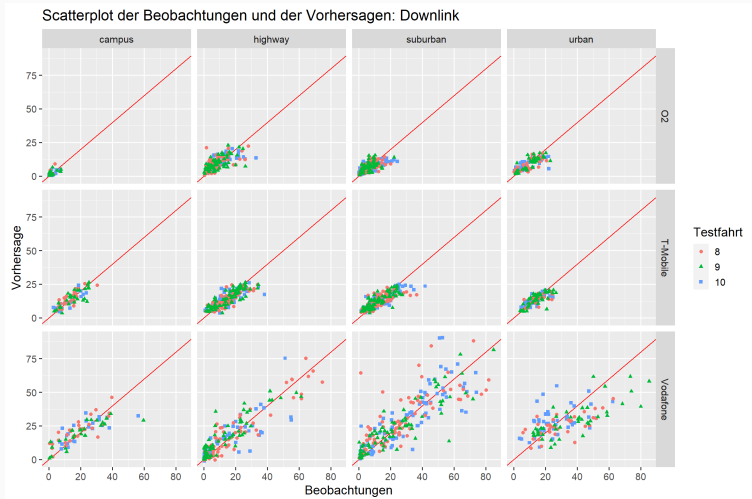


Abbildung 4: XGBoost Out-of-Sample Vorhersagen der Download-Rate

Task I - Vorhersage der Datenrate

Regression mit ARMA-Fehlern

Regression mit ARMA-Fehlern

Gegeben:

- Beobachtungen (y_1, \dots, y_T) der Zeitreihe $(y_t)_t$
- Beobachtungen $(x_1^{(i)}, \dots, x_T^{(i)})$ der Zeitreihen $(x_t^{(i)})_t$ für $i = 1, \dots, k$

Modellgleichung: Regression mit ARMA(p, q)-Fehlern

$$y_t = c + \sum_{j=1}^k \beta_j x_t^{(j)} + \eta_t \text{ mit}$$

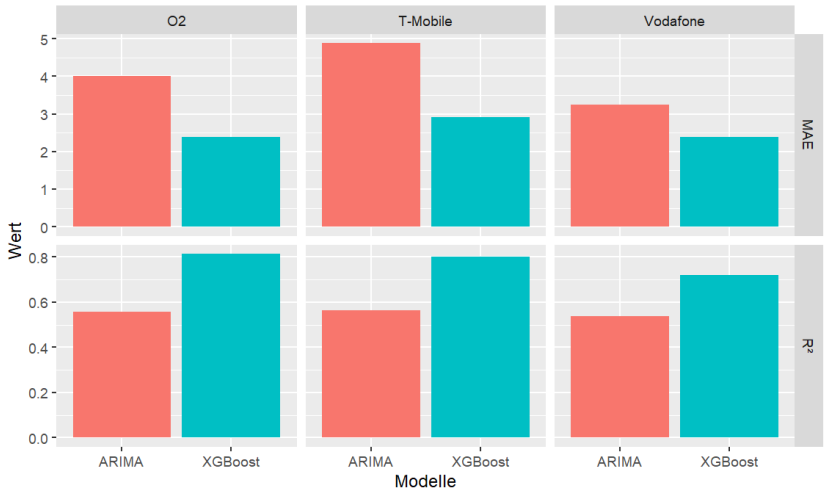
$$\eta_t = \underbrace{\sum_{k=1}^p \phi_p \eta_{t-p}}_{\text{vergangene Fehler: LM}} + \underbrace{\sum_{l=1}^q \theta_l \epsilon_{t-l}}_{\text{vergangene Fehler: ARMA}} + \epsilon_t$$

Task I - Vorhersage der Datenrate

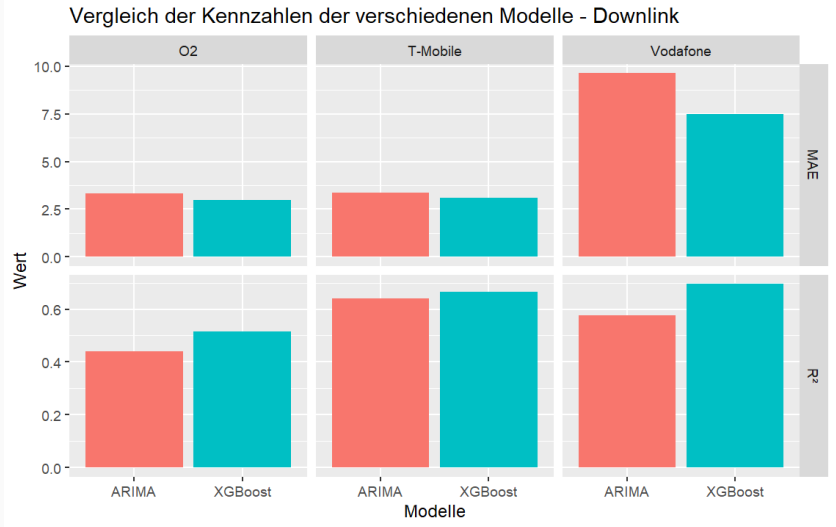
Modellvergleich

Modellvergleich Uplink - Kennzahlen

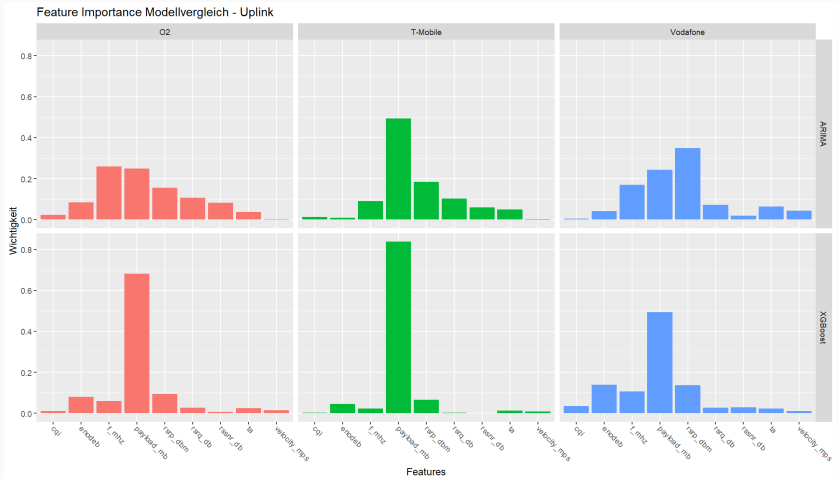
Vergleich der Kennzahlen der verschiedenen Modelle - Uplink



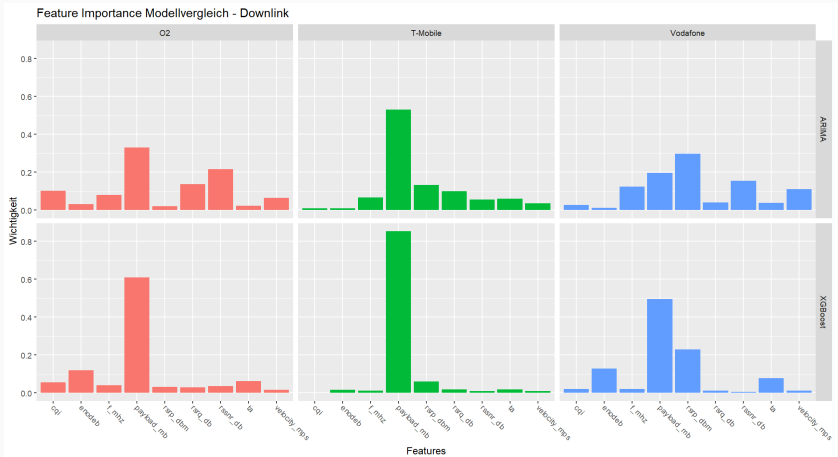
Modellvergleich Downlink - Kennzahlen



Modellvergleich Uplink - Feature Importance

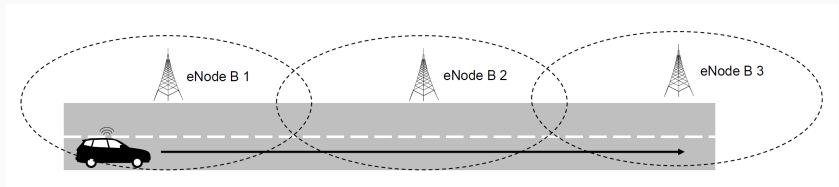


Modellvergleich Downlink - Feature Importance



Vorhersage des Handovers und Link Lifetime

- Vergleich des RSRP Wertes zur verbundenen Zelle sowie zu den Nachbarzellen
- Vorhersage des Handovers durch Angabe der Link Lifetime

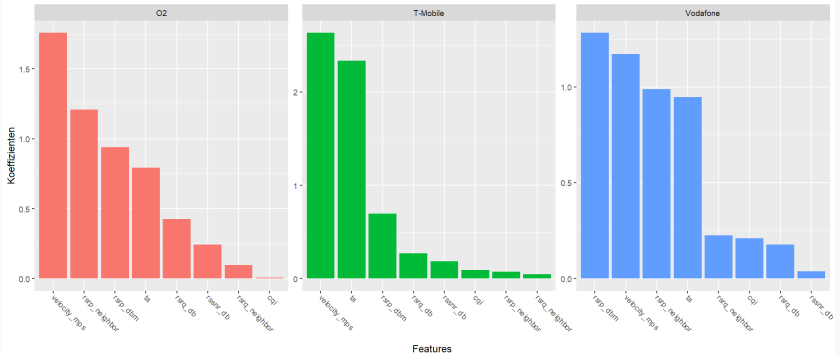


Task II - Handover Vorhersage und Link Lifetime

Feature Importance

Feature Importance

Feature Importance der verschiedenen Provider - Link-Lifetime





T. Chen and C. Guestrin.

Xgboost: A scalable tree boosting system.

CoRR, abs/1603.02754, 2016.



T. Hastie, R. Tibshirani, and J. Friedman.

The elements of statistical learning: data mining, inference and prediction.

Springer, 2 edition, 2009.



B. Sliwa and C. Wietfeld.

Data-driven network simulation for performance analysis of anticipatory vehicular communication systems.

IEEE Access, 7:172638–172653, 2019.

Regression mit ARMA-Fehlern

h-Schritt Punktvorhersage

- Ersetze Beobachtungen zu zukünftigen Zeitpunkten mit deren Vorhersagen
- Ersetze Fehler an vergangenen Zeitpunkten durch das entsprechende Residuum
- Ersetze Fehler an zukünftigen Zeitpunkten durch 0

Beispiel: $h = 2, k = 1, p = 2, q = 2$

$$y_t = c + \beta_1 x_t + \epsilon_t \text{ mit } \epsilon_t = \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + \theta_1 e_{t-1} + \theta_2 e_{t-2} + e_t$$

$$\widehat{y}_{t+1} = c + \beta_1 x_t + \widehat{\epsilon}_{t+1} \text{ mit } \widehat{\epsilon}_{t+1} = \phi_1 \epsilon_t + \phi_2 \epsilon_{t-1} + \theta_1 e_t + \theta_2 e_{t-1} + \underbrace{\widehat{e}_{t+1}}_{=0}$$

$$\widehat{y}_{t+2} = c + \beta_1 x_t + \widehat{\epsilon}_{t+2} \text{ mit } \widehat{\epsilon}_{t+2} = \phi_1 \widehat{\epsilon}_{t+1} + \phi_2 \epsilon_t + \theta \underbrace{\widehat{e}_{t+1}}_{=0} + \theta e_t + \underbrace{\widehat{e}_{t+2}}_{=0}$$

Gradient Boosted Trees

- Kann man aus vielen „schwachen“ Lernern einen starken Lerner konstruieren?
 - ⇒ Ja, Boosting ist eines der mächtigsten Konzepte des Machine Learning [2]
- Kombination von einfachen CART Bäumen zu einem starken Ensemble
 - ⇒ Ähnlich zu Random Forest
- Der Unterschied zum Random Forest liegt im Training!

Training von Gradient Boosted Trees

- Bäume werden nacheinander zum Ensemble hinzugefügt
- Jeder neue Baum versucht, die Schwächen seiner Vorgänger „auszubügeln“
 - ⇒ *Additives Training*
- Je mehr Bäume aufgenommen werden, desto geringer wird der Training-Error (das Modell wird aber komplexer)
 - ⇒ Kontrolle des *Bias-Variance Tradeoffs*
 - ⇒ Zusätzlich gibt es Regularisierungs-Parameter

Implementierung: XGBoost

- Liefert state-of-the-art Performance in einer Vielzahl von ML-Problemen
- In 2015 haben 19/25 Gewinner von Kaggle-Competitions XGBoost eingesetzt
- Kann problemlos auf mehrere Milliarden Training Samples skaliert werden
- Lässt sich aber auch hervorragend auf ressourcenbegrenzten Systemen einsetzen [1]