

Fallstudien II

Laura Kampmann, Christian Peters, Alina Stammen

11. Dezember 2020

1. Task I - Vorhersage der Datenrate

Gradient Boosted Trees

Regression mit ARMA-Fehlern

Modellvergleich

2. Task II - Handover Vorhersage und Link Lifetime

Feature Importance

Task I - Vorhersage der Datenrate

Task I - Vorhersage der Datenübertragungsrate

- Ziel: Evaluation von neuen *anticipatory vehicular communication systems* durch möglichst realitätsnahe Simulationen [2]
⇒ Ansatz: *Data-Driven Network Simulation*
- Durch Machine Learning Modelle sollen möglichst realistische Vorhersagen der Datenraten generiert werden
- Hoffnung: Bessere Aussagekraft der Simulationen durch Einsatz echten Datenmaterials

Task I - Vorhersage der Datenrate

Gradient Boosted Trees

Gradient Boosted Trees

- Kann man aus vielen "schwachen" Lernern einen starken Lerner konstruieren?
 - ⇒ Ja, Boosting ist eines der mächtigsten Konzepte des Machine Learning [1]
- 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

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

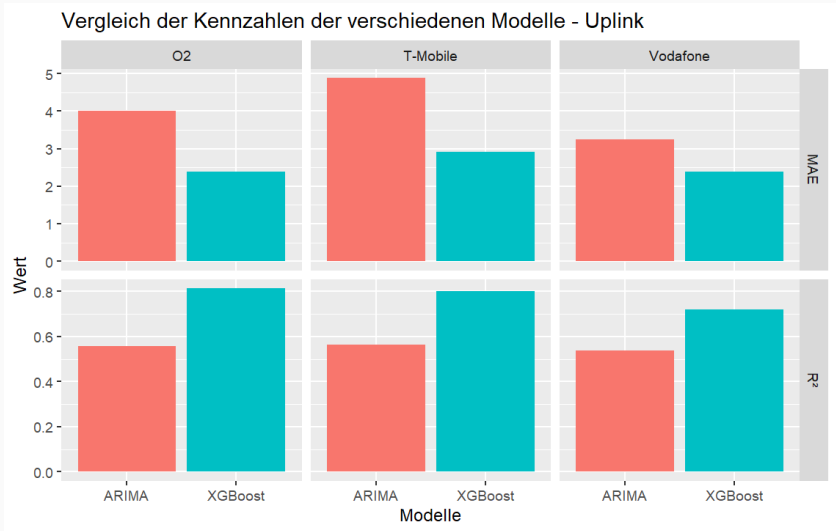
$$y_t = c + \sum_{j=1}^k \beta_j x_t^{(j)} + \eta_t \text{ mit}$$

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

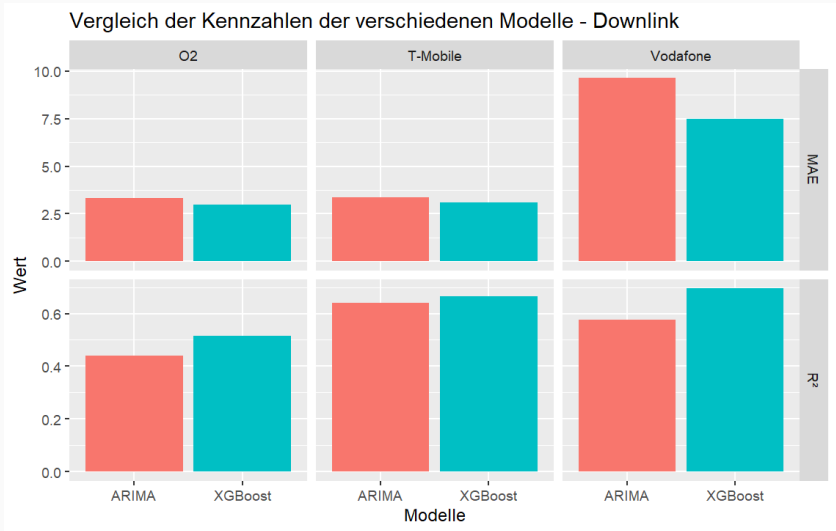
Task I - Vorhersage der Datenrate

Modellvergleich

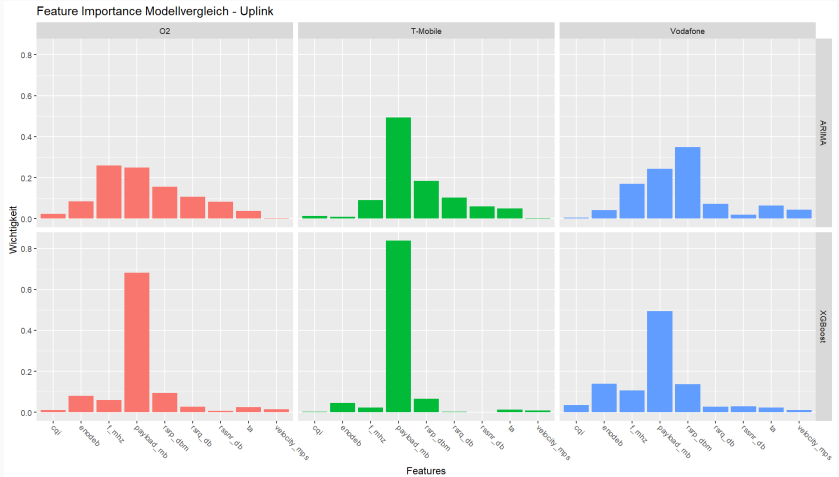
Modellvergleich Uplink - Kennzahlen



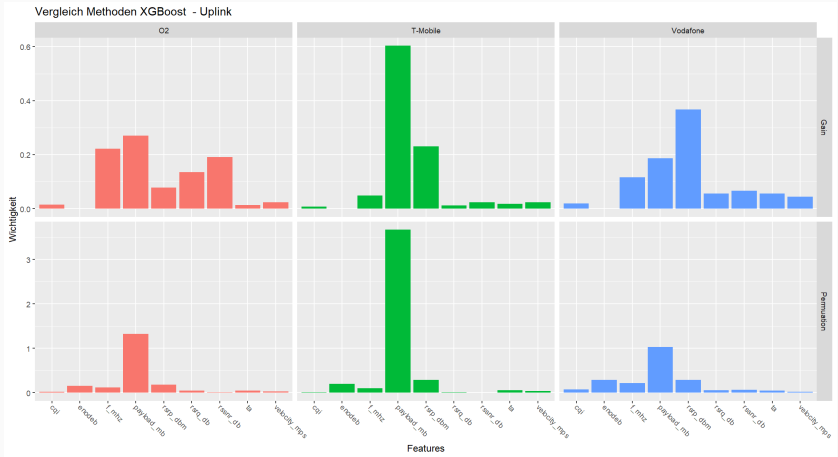
Modellvergleich Downlink - Kennzahlen



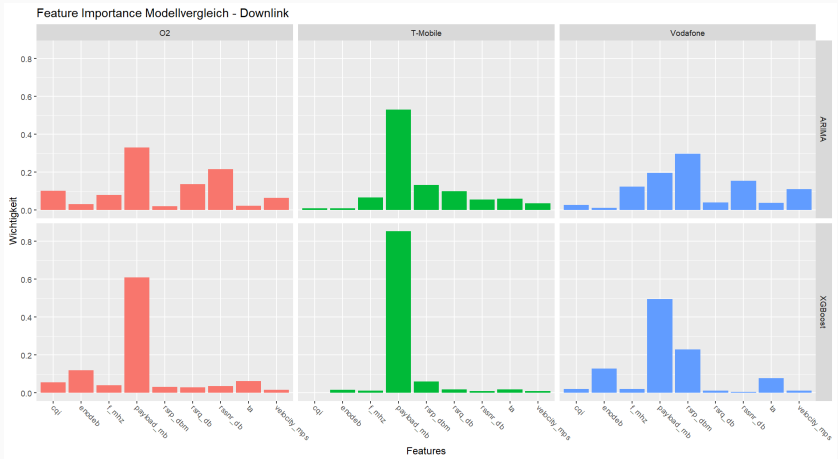
Modellvergleich Uplink - Feature Importance



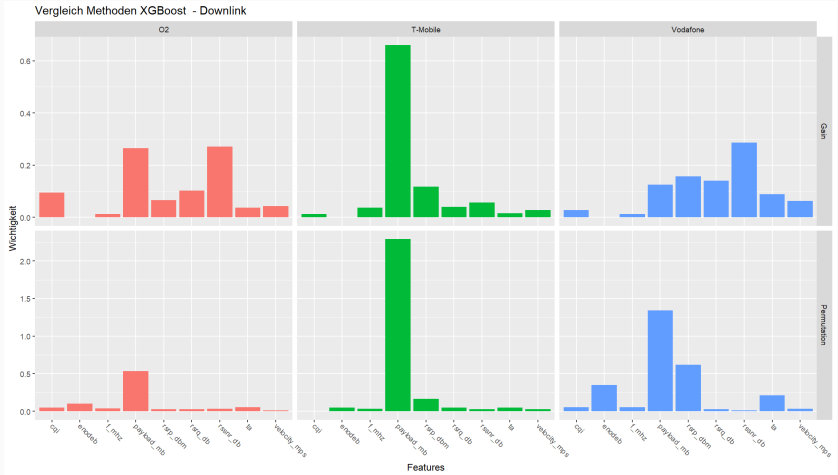
Methodenvergleich XGBoost - Uplink



Modellvergleich Downlink - Feature Importance



Methodenvergleich XGBoost - Downlink



Task II - Handover Vorhersage und Link Lifetime

Idee: Prädiktionsmodell XGBoost für Link Lifetime mit Einfluss des RSRP/RSRQ der verbundenen sowie der Nachbarzellen

→ Datentransformation

- RSRP/RSRQ Nachbarzellen :
 - mehrere Messungen - Filtern des besten Wertes zum aktuellen Zeitpunkt
 - keine Messungen - Übernehmen des letzten Wertes
- eNodeB Wechsel → Response Variable Link Lifetime

- **link_lifetime** : Link-Lifetime
- **rsrp_dbm/rsrq_db** : Signalstärke/Signalqualität (RSRP/RSRQ) der verbundenen Zellen
- **rsrp_neighbor/rsrq_neighbor** : Signalstärke/Signalqualität (RSRP/RSRQ) der Nachbarzellen
- **rssnr_db** : Signal-Rausch-Verhältnis (RSSNR)
- **eNodeB** : Funkmasten im LTE-Netzwerk
- **velocity_mps** : Geschwindigkeit des mobilen Endgeräts
- **ta** : Timing Advance (TA) - Wert zur Synchronisation zwischen Up- und Downlink
- **cqi** : Channel Quality Indicator (CQI)

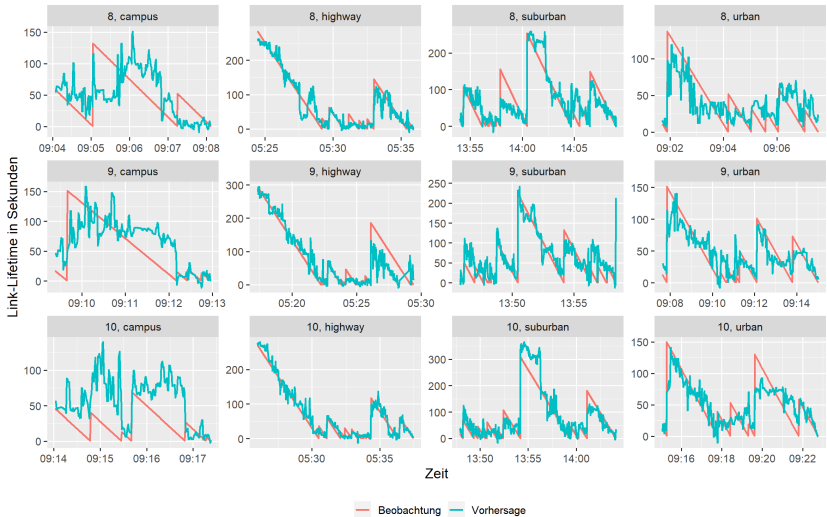
Wichtige Schritte :

- Aufsplitten der Daten - Training/ Test
- Zufälliger Grid-Search
- Tunen der Parameter - Zeitreihenkreuzvalidierung
- Validieren des Modells auf dem Testdatensatz

→ **Analog zu Task I**

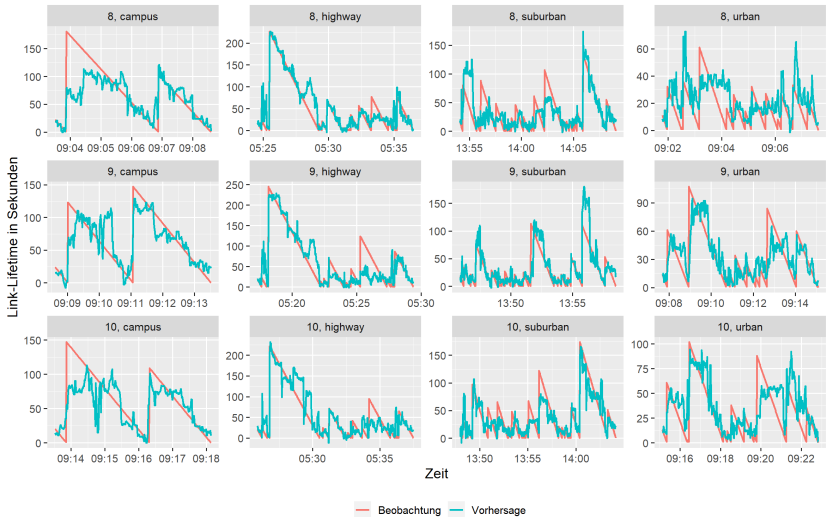
Ergebnisse - Zeitreihenplot O2

Link-Lifetime Vorhersage: O2



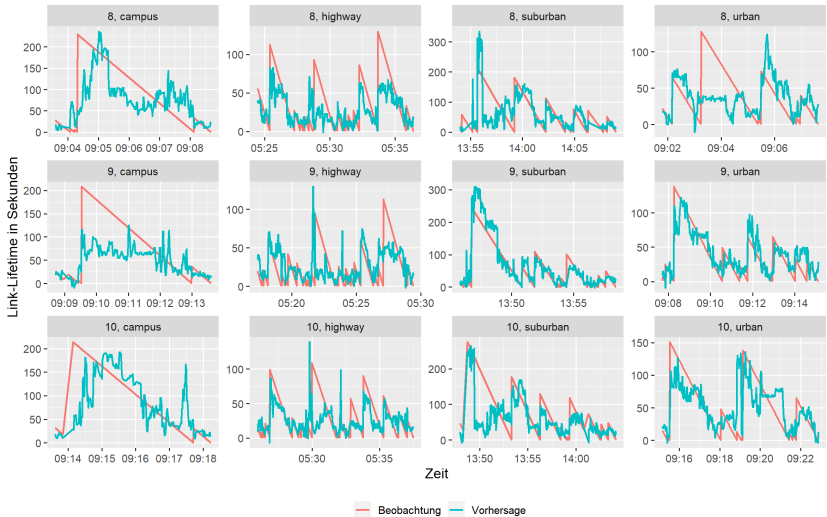
Ergebnisse - Zeitreihenplot T-Mobile

Link-Lifetime Vorhersage: T-Mobile



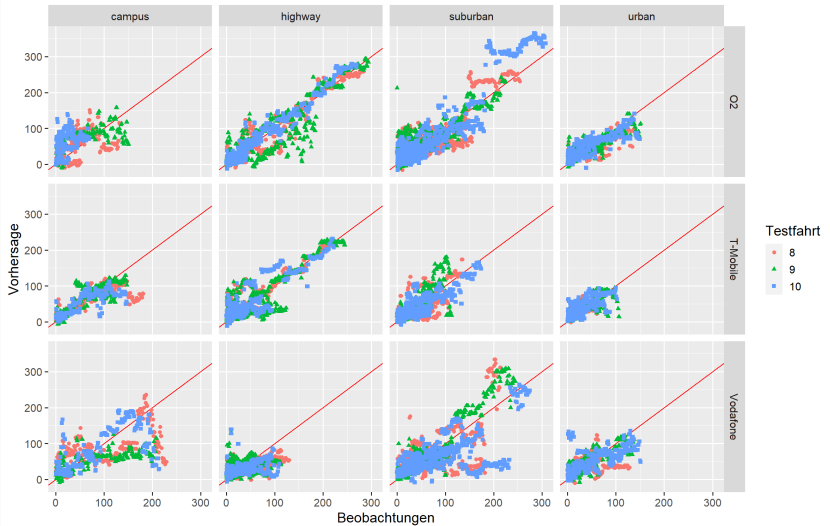
Ergebnisse - Zeitreihenplot Vodafone

Link-Lifetime Vorhersage: Vodafone

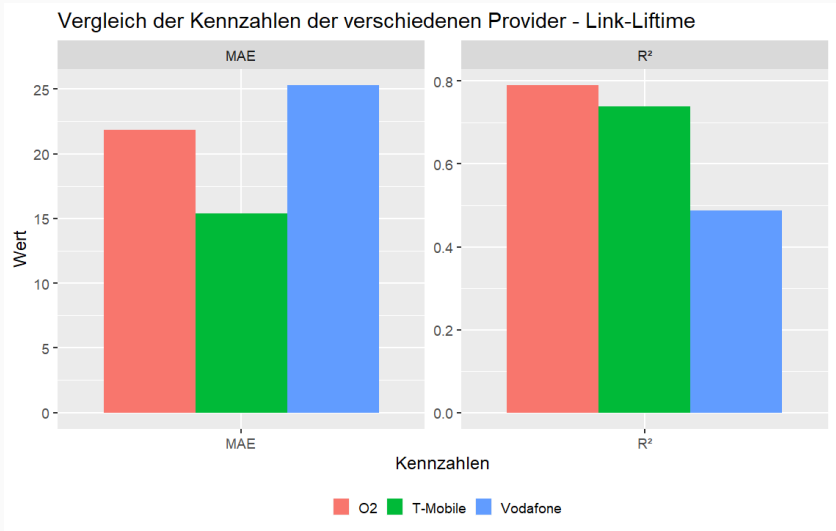


Ergebnisse - Scatterplot

Scatterplot der Beobachtungen und der Vorhersagen:



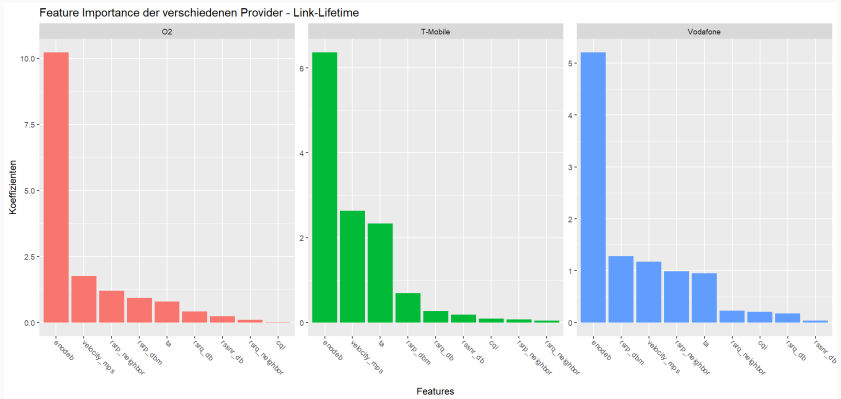
Ergebnisse - Kennzahlen



Task II - Handover Vorhersage und Link Lifetime

Feature Importance

Feature Importance





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.

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

$$\begin{aligned}y_t &= c + \beta_1 x_t + \epsilon_t \text{ mit } \epsilon_t = \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + \theta_1 e_{t-1} + \theta_2 e_{t-2} + e_t \\ \widehat{y}_{t+1} &= c + \beta_1 x_t + \widehat{\epsilon}_{t+1} \text{ mit } \widehat{\epsilon}_{t+1} = \phi_1 \epsilon_t + \phi_2 \epsilon_{t-1} + \theta_1 e_t + \theta_2 e_{t-1} + \underbrace{\widehat{e}_{t+1}}_{=0} \\ \widehat{y}_{t+2} &= c + \beta_1 x_t + \widehat{\epsilon}_{t+2} \text{ mit } \widehat{\epsilon}_{t+2} = \phi_1 \widehat{\epsilon}_{t+1} + \phi_2 \epsilon_t + \theta \underbrace{\widehat{e}_{t+1}}_{=0} + \theta e_t + \underbrace{\widehat{e}_{t+2}}_{=0}\end{aligned}$$