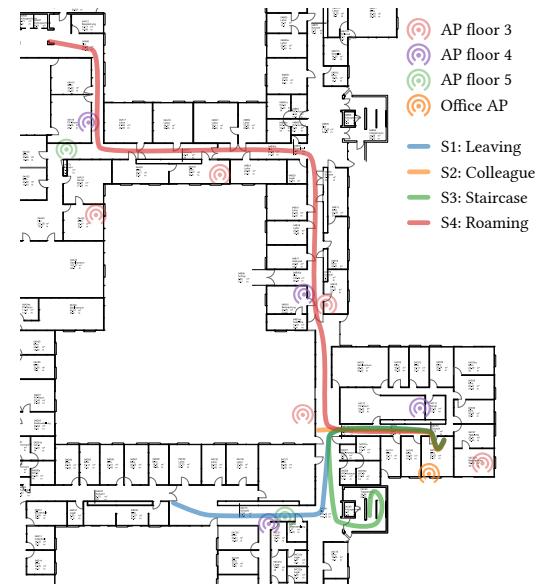


Learning Wi-Fi Connection Loss Predictions for Seamless Vertical Handovers Using Multipath TCP



Jonas Höchst, Artur Sterz,
Alexander Frömmgen, Denny Stohr,
Ralf Steinmetz, Bernd Freisleben

TU Darmstadt and Philipps-Universität Marburg



Introduction: Smartphones - Daily Companions

Communication

Information

Entertainment

Wi-Fi

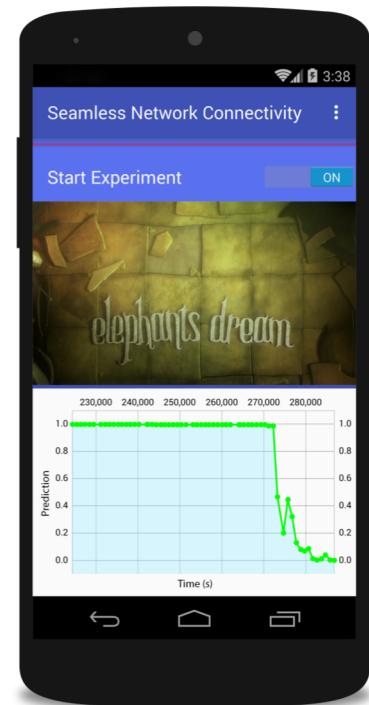
Cellular

→ Handover



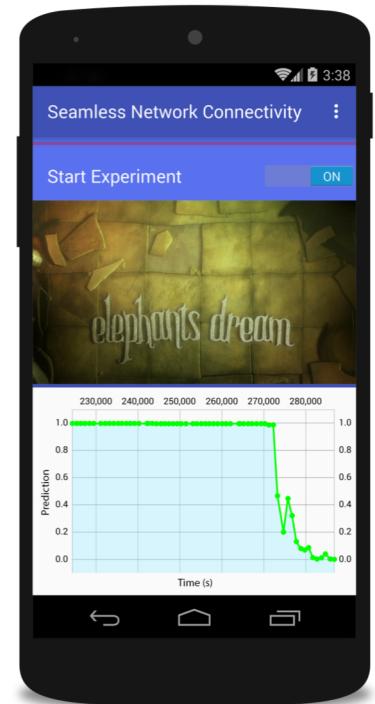
Introduction: Vertical Handovers Today

- Handovers are performed reactively:
 - Based on weak RSSI or high packet loss
 - Change of default gateway
 - Application has to deal with connection loss
- Multipath-TCP enables seamless handovers
 - Multiple subflows on all available network interfaces
 - Drawback: energy usage, use of limited data plans

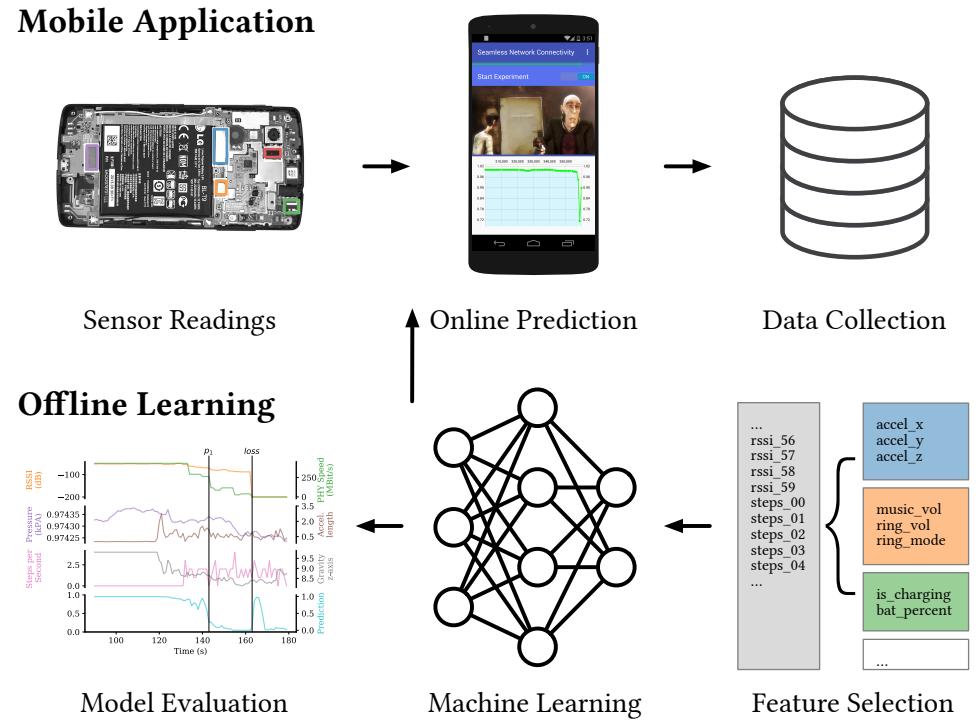


Introduction: Contributions

- Novel data-driven, proactive approach for seamless vertical Wi-Fi/cellular handovers
- Multiple heterogeneous smartphone sensors to predict Wi-Fi connection loss
- Multipath-TCP based seamless connection handover
- Experimental evaluation based on Quality of Experience
- Open demo implementation and experimental artifacts



Conceptual Overview



Smartphone Sensor Readings

Wi-Fi Properties

Spatial Orientation

Wi-Fi Access Points

Magnetic Field

Step Counter

Gravity

Linear Acceleration

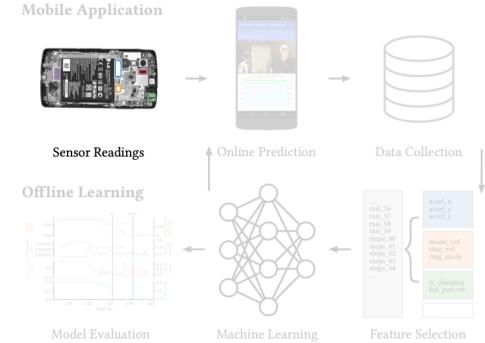
Ringer Mode

Power State

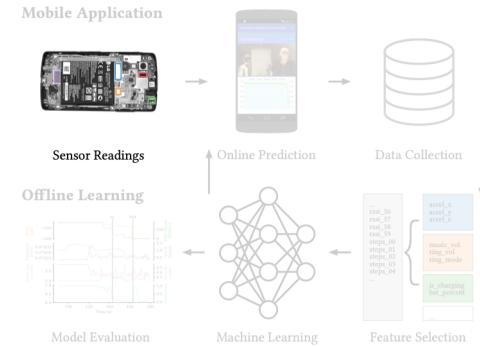
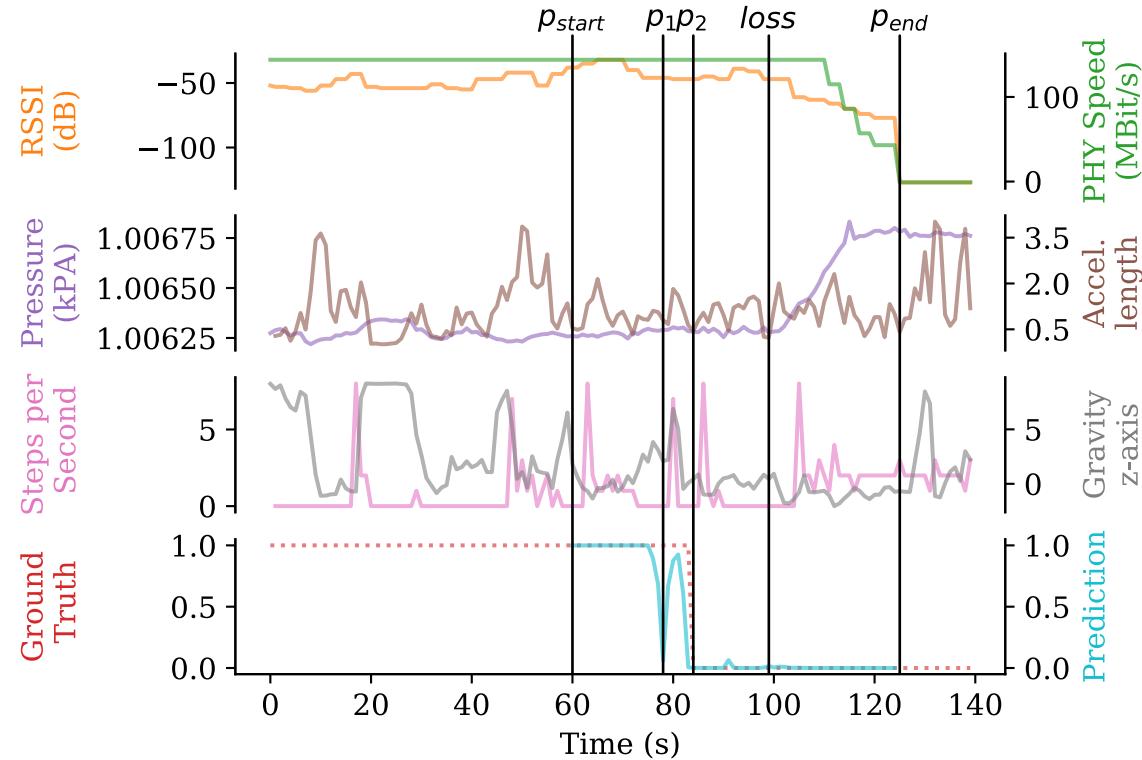
Audio State

Bluetooth Neighborhood

Atmospheric Pressure

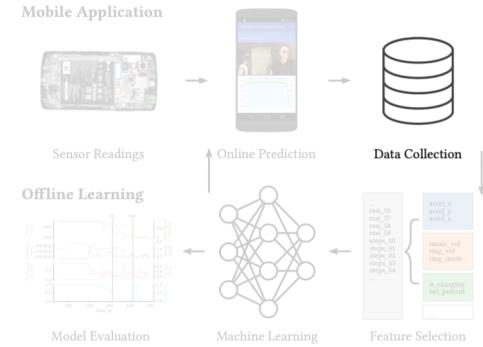


Sensor Data Example

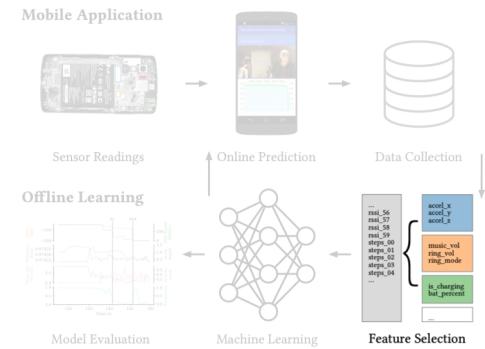
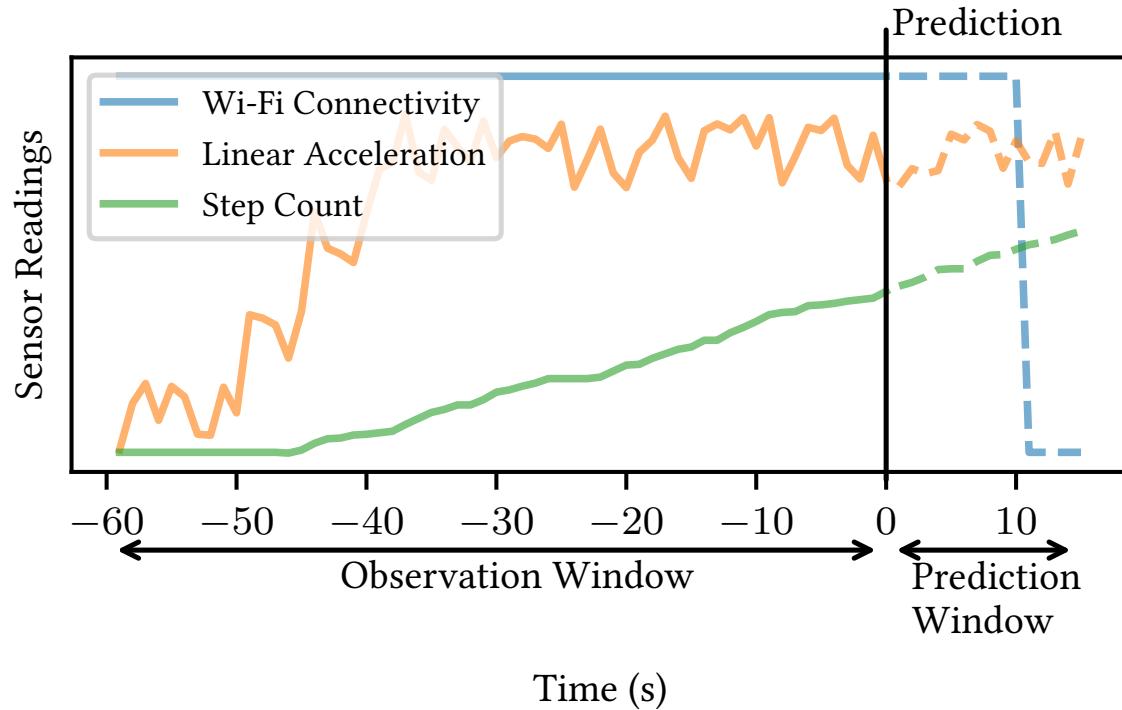


Data Collection

- 20 GB of sensor data from five different users
 - Running for the whole day – daily lives of users
- 900,000 unique samples, collected in three months
- Training and test set
 - a) Random split of all available samples (70:30)
 - b) User-based split: learn with some users, test with the others

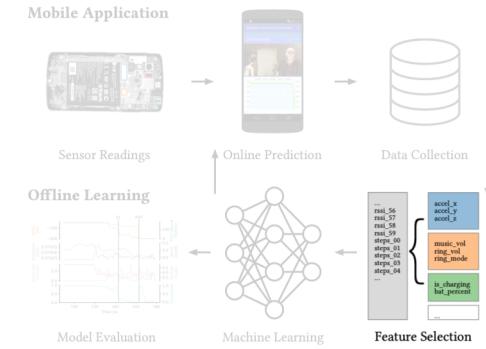


Feature Selection: Observation & Prediction



Feature Selection: Input Vectors

- *Full Feature Vector*
 - All 25 available sensors, $25 \times 60 = 1500$ features
- *Reduced Feature Vector*
 - Atmospheric pressure, linear acceleration, power, step counter, gravity, Wi-Fi (frequency, speed, RSSI)
 $8 \times 60 = 480$ features
- *Ground Truth*
 - Wi-Fi RSSI > -70 dBm, shifted



Machine Learning: Random Forest

- Requires equally distributed samples:
down-sampling, 10 random trees

Event	Prec.	Recall	F_1 -score	Support
Loss	0.86	0.98	0.91	52503
No Loss	1.00	0.98	0.99	438772
Total	0.98	0.98	0.98	491275

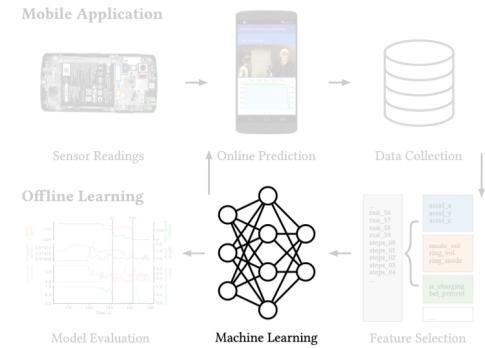
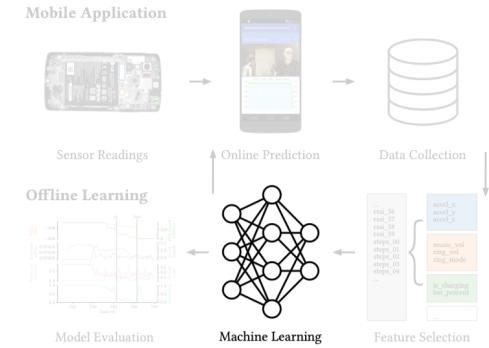


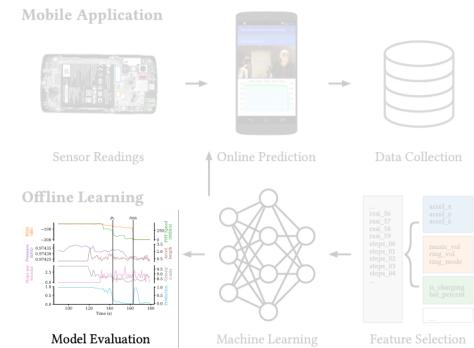
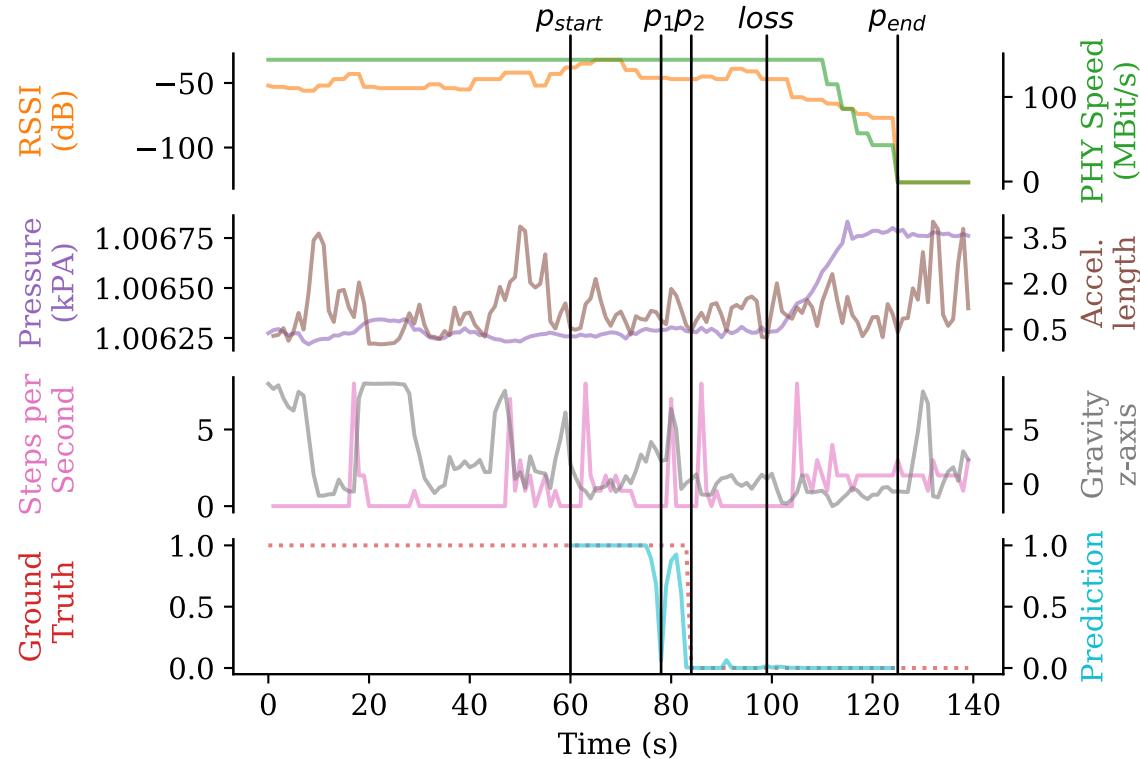
Table: Random Data Split, Reduced Feature Vector

Machine Learning: Neural Networks

- Input Layer: 60 (only-RSSI) to 1500 neurons, depending on feature vector
 - Hidden Layers in different configurations:
 - NN 1: 1 hidden layer of (100) neurons
 - NN 2: 3 hidden layers of (300, 200, 100)
 - NN 3: 5 hidden layers of (400, 400, 400, 400, 400)
 - Output Layer: 1 neuron, indicating loss probability



Model Evaluation: Example



Model Evaluation: Random Data Split

Metric	Forest	NN 1	NN 2	NN 3
Loss Prec.	0.89	0.95	0.97	0.97
Loss Recall	0.98	0.94	0.95	0.95
F_1 -score	0.93	0.94	0.96	0.96

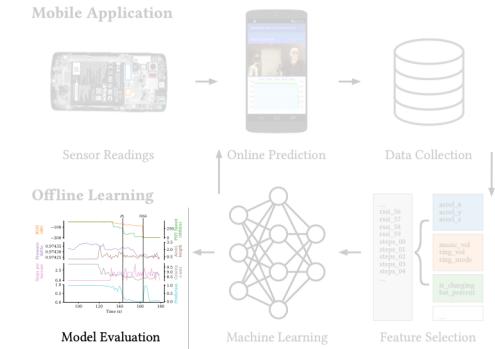


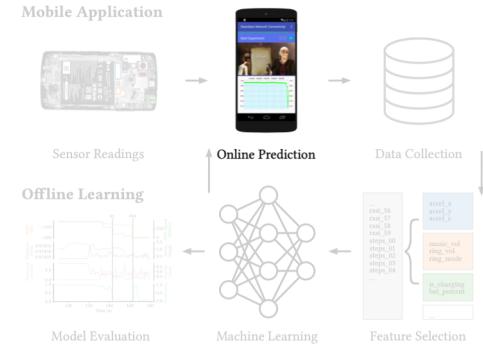
Table: Reduced Feature Vector, Random Data Split

Online Prediction: Mobile Application

On-Device Model Execution

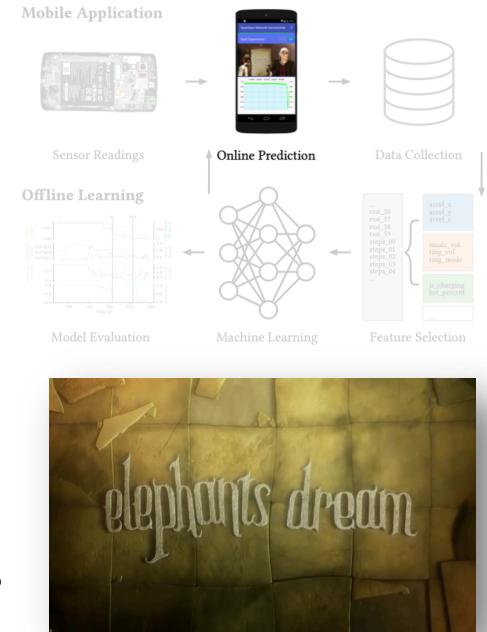
DASH.js Video Playback

MPTCP Handovers



Online Prediction: DASH.js Video

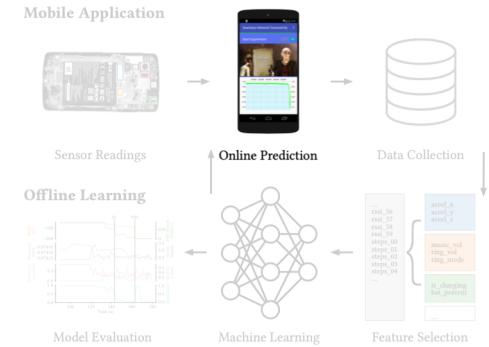
- Dynamic Adaptive Streaming over HTTP(s)
 - Configuration: BOLA adaptation algorithm, 10 s buffer
 - H.264 video, AAC audio
 - Segments of 2 seconds
 - Available bandwidths: 1, 2, and 4 Mbit/s
- Base metrics: Stalls, Bitrate, Adaptations, Buffer levels



Open Movie:
[Elephant's Dream](#)

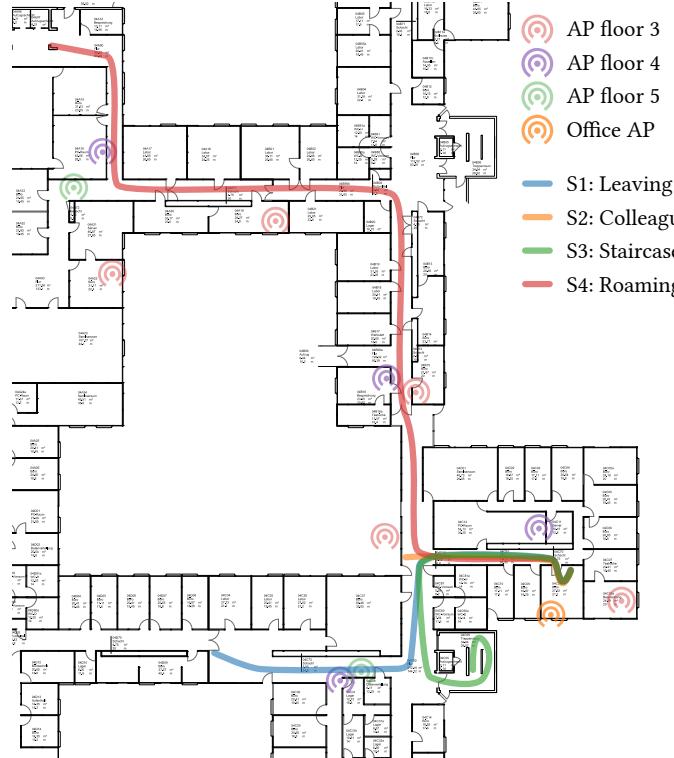
Online Prediction: MPTCP Handovers

- Toggle LTE state based on online prediction
- MPTCP kernel implementation (v0.86) for Android
- MultipathControl (De Coninck et al.)
- Video server: MPTCP v0.92
 - *redundant* scheduler
 - *fullmesh* path manager



Experimental Evaluation: Scenarios

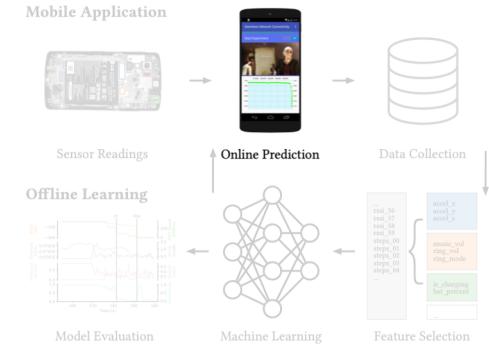
- Four scenarios:
 - Leaving the office (1)
 - Visiting a colleague (2)
 - Using the staircase (3)
 - Wi-Fi roaming support (4)
- Three connectivity modes:
 - Android, MPTCP, Seamless
- Nexus 5, Android 4.4.2

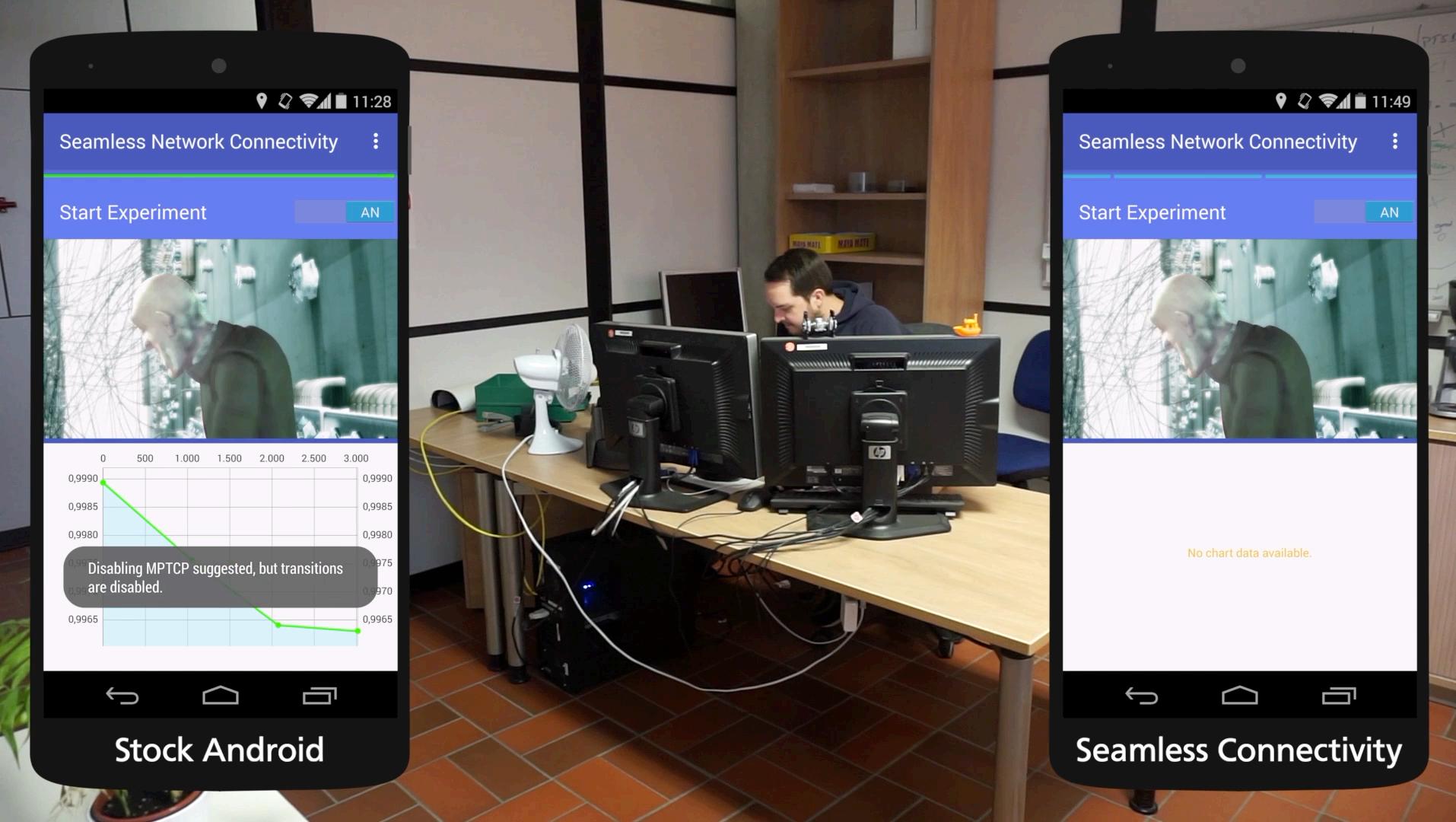


Online Prediction

Demo

<https://youtu.be/E0CFLk82s6s>





Experimental Evaluation

(a) Scenario 1: Leaving

Mode	# St.	\varnothing St.	# A.	HQ	\varnothing TD
<i>Stock</i>	3	1.46 s	23	87 %	21.75 MB
<i>MPTCP</i>	0	0 s	20	89 %	41.32 MB
<i>Seamless</i>	0	0 s	27	88 %	36.11 MB

(b) Scenario 2: Colleague

Mode	# St.	\varnothing St.	# A.	HQ	\varnothing TD
<i>Stock</i>	0	0 s	10	92 %	0 MB
<i>MPTCP</i>	0	0 s	10	91 %	9.98 MB
<i>Seamless</i>	0	0 s	17	92 %	9.59 MB

(c) Scenario 3: Staircase

Mode	# St.	\varnothing St.	# A.	HQ	\varnothing TD
<i>Stock</i>	3	2.06 s	49	80 %	0 MB
<i>MPTCP</i>	0	0 s	32	87 %	33.92 MB
<i>Seamless</i>	0	0 s	28	85 %	16.81 MB

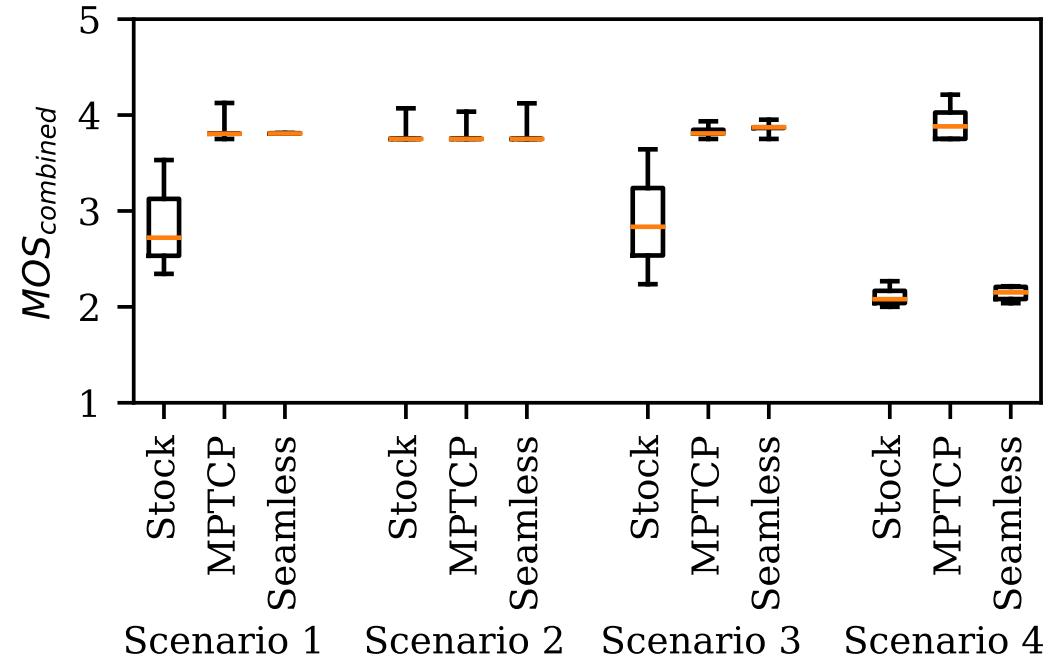
(d) Scenario 4: Wi-Fi Roaming

Mode	# St.	\varnothing St.	# A.	HQ	\varnothing TD
<i>Stock</i>	18	14.98 s	42	53 %	0.89 MB
<i>MPTCP</i>	0	0 s	38	86 %	71.99 MB
<i>Seamless</i>	15	5.47 s	23	84 %	15.50 MB

Overview of Experimental Results



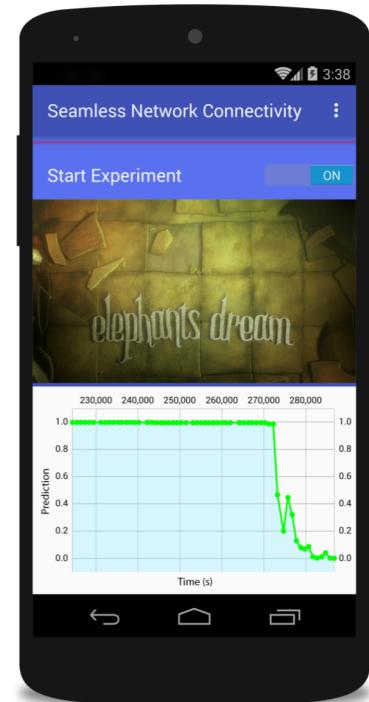
Experimental Evaluation: QoE Results



$MOS_{combined}$ values grouped to connectivity modes and scenarios

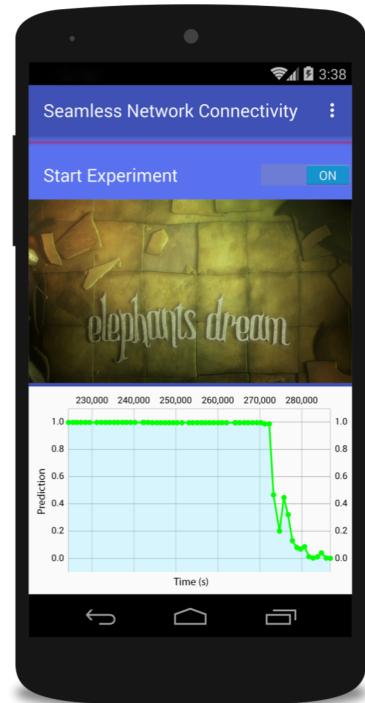
Conclusion

- Novel data-driven approach for Wi-Fi loss prediction
 - Precision of up to 0.97; Recall of up to 0.98
- Promising results with MPTCP-based handovers
 - QoE improvements of 2.7 to 3.8 in certain scenarios
 - Lower cellular data usage (50%) compared to traditional MPTCP handovers

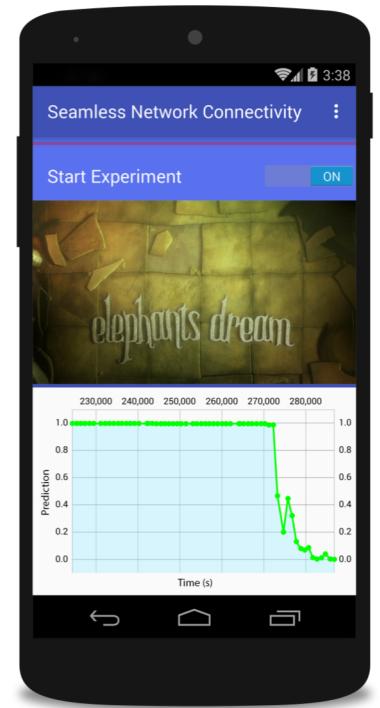
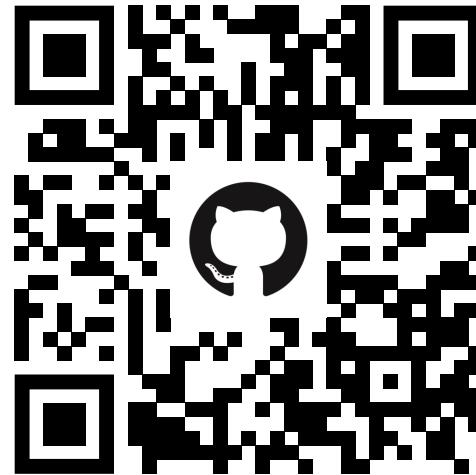


Future Work

- Enlarge sensor variety: from contextual sensors to domain specific sensors, i.e., to detect high network load
- Online learning on smartphones
 - User-specific models, e.g., user / access point combinations
- Multi-RAT handover predictions (Wi-Fi, 3G, LTE, 5G, ...)
- Hardware / low-level implementations
 - Smartphone sensor hub, lightweight neural networks



One more thing...



<https://umr-ds.github.io/seamcon/>

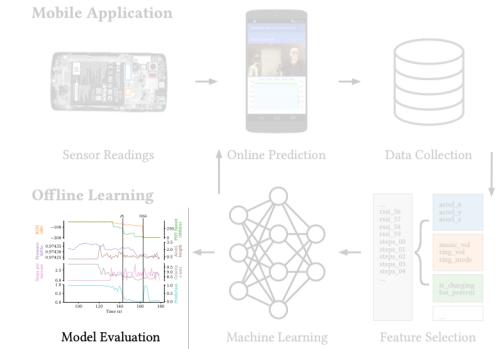
The End

Time for

Questions

Model Evaluation: User-based Data Split

- *Full Feature Vector*: 0.91, 0.72, and 0.68 precision in the Wi-Fi loss class
- *Reduced Feature Vector*: 0.93, 0.92, and 0.79 precision in the Wi-Fi loss class
- Neural networks are capable of predicting Wi-Fi loss.
 - The *Reduced Feature Vector* generalizes better;
 - per-user training significantly improves the results.
- Overall best performance: *NN 3* with the *Reduced Feature Vector*



Experimental Evaluation

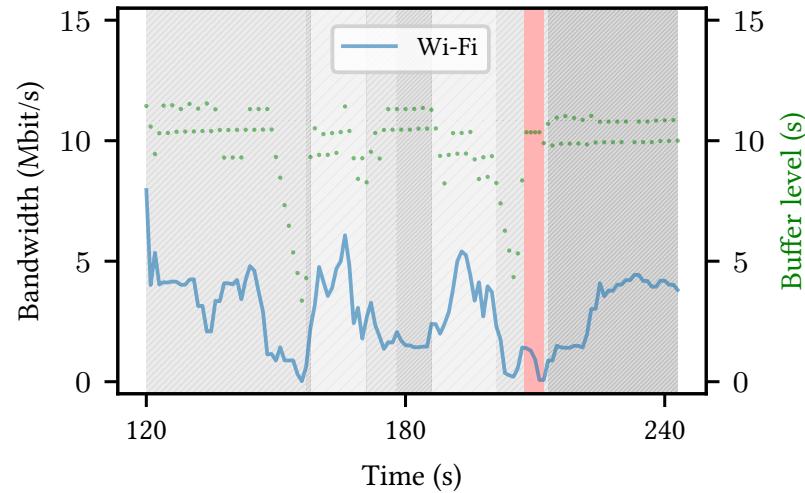
$$MOS_{stall} = 3.5 \times e^{-(0.15 \times L + 0.19) \times N} + 1.5 \quad (1)$$

$$MOS_{quality} = 0.003 \times e^{0.064 \times t \times 100} + 2.498 \quad (2)$$

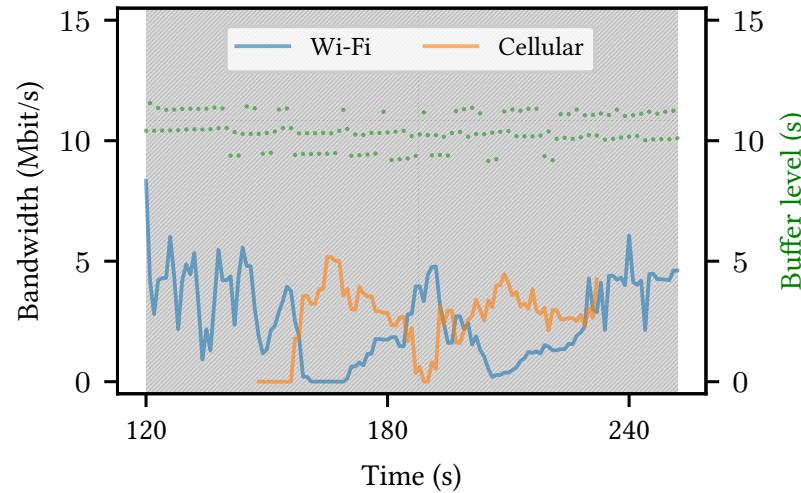
$$MOS_{combined} = \frac{MOS_{stall} + MOS_{quality}}{2} \quad (3)$$

Mean Opinion Score as QoE Metric

Experimental Evaluation: QoE Results



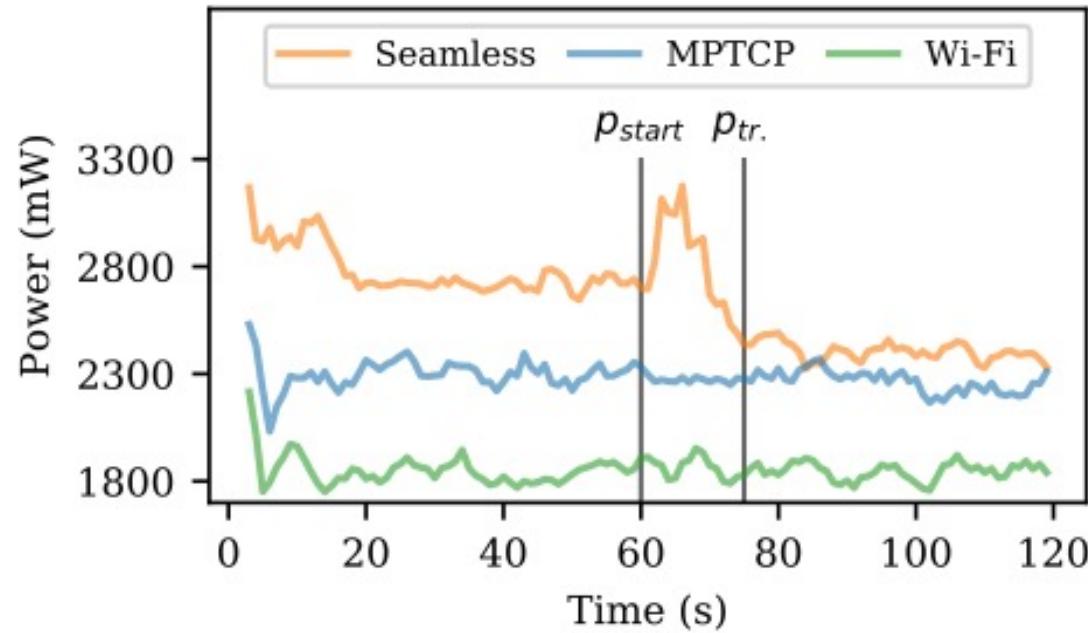
a) *Stock Android*



b) *Seamless*

Stock and Seamless in Scenario 3

Experimental Evaluation: Power consumption



Power consumption across different connectivity modes