Lucerne University of
Applied Sciences and Arts

HOCHSCHULE
LUZERN

Engineering & Architecture

KnowlEDGE Project – Presentation Tech-Lunch IET 19/05/2021

Institute of Electrical Engineering IET

Dr. Severin Nowak
Research Associate
CC Digital Energy and Electric Power / Advanced Analytics und Intelligent Energy Systems
T direct +41 41 349 37 12
severin.nowak@hslu.ch

Horw 27.05.2021

Overview

Smart Meters (Opportunities and Challenges)

Context of the KnowlEDGE project

KnowlEDGE project aims, use cases

Building blocks:
- ➢ Data management
- ➢ Partitioned power flow
- ➢ Load prediction
- ➢ Deployment on edge devices

I am happy to answer questions as we go!

Smart Meters

Smart meters are well established as a rich source of customer insights

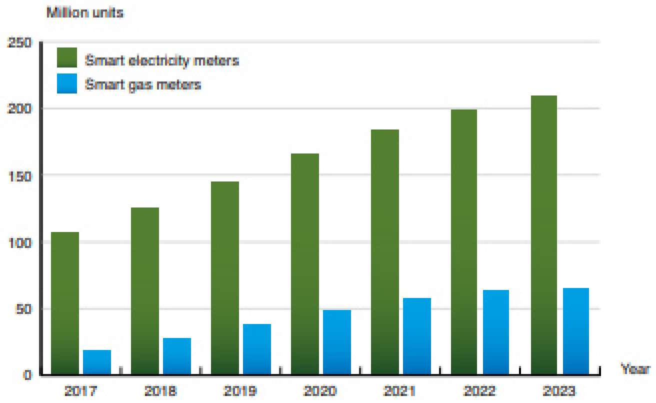
- ➢ **Record power consumption in 15-minute intervals**
- ➢ **Send data to DSO once per day**
- ➢ **Used for energy billing**

Legislation: Electricity supply regulation in effect as of 1.1.2018 which defines:
Target SFOE Smart Meter Roll-out: **80% of end-customers by 2027**

Europe: ~70% of end-customers in 2020 have smart meters installed

Pros:

- Potential energy savings due to awareness (Estimate SFOE at least -2%)
  - ➢ Economic benefits
  - ➢ Resources savings
- Fundamental step in the digitalization of the energy sector
- Meter failures are detected earlier
- Different price signals (i.e. day/night)
- Enables NILM (non-intrusive load monitoring)



Installed base of smart electricity meters and smart gas meters (EU28+2 2017–2023)

Smart Meters

Data-privacy:
- Measurement data belongs to customers (Data Protection Act)

Distribution System Operator (DSO):
- **Must** follow strict data-privacy policies (protect confidentiality and integrity)
- May use aggregated **data for monitoring, control, tariff design, and network planning**
- Third-party access possible under strict anonymization, and in aggregated form
- Use data to assign consumption to balance group/origin of generation

Potential for **much** more elaborate use of data for:
- Detection and prediction of **network congestion**
- **Quality of supply**

⇒ Requires the use of **inidividual load profiles**
⇒ **Problematic when centrally computed**

KnowlEDGE project overview

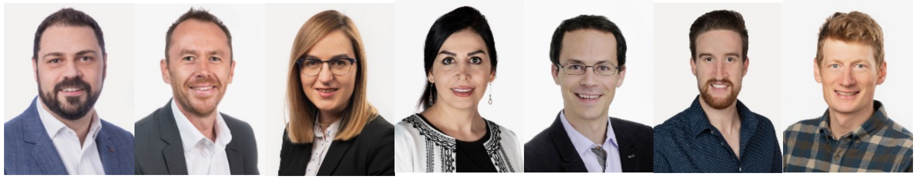
Decentralized, secure and privacy-protecting AI to improve grid reliability, resilience and cost performance for DSOs

SFOE call of Grids Research Program 2020

Project timeline and milestones:

Feb 2020: Pre-proposal submitted to SFOE-GRIDS call
May 2020: Full submission
Sep 2020: Project start
May 2021: Use cases and AI model development in progress
Jul 2022: Field testing and feasibility testing complete

HSLU - Project team:



Antonios, Ben, Viktorija, Mojgan, Patrick, Fabian, Severin

Project partners:

Landis+Gyr
manage energy better

ROMANDE ENERGIE

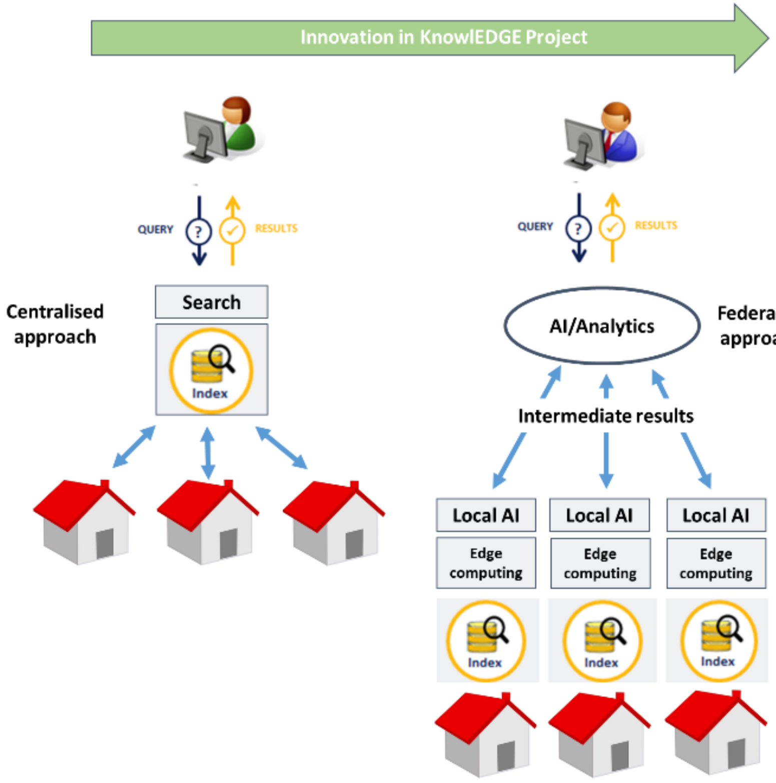
KnowlEDGE project overview

Aim of the project:

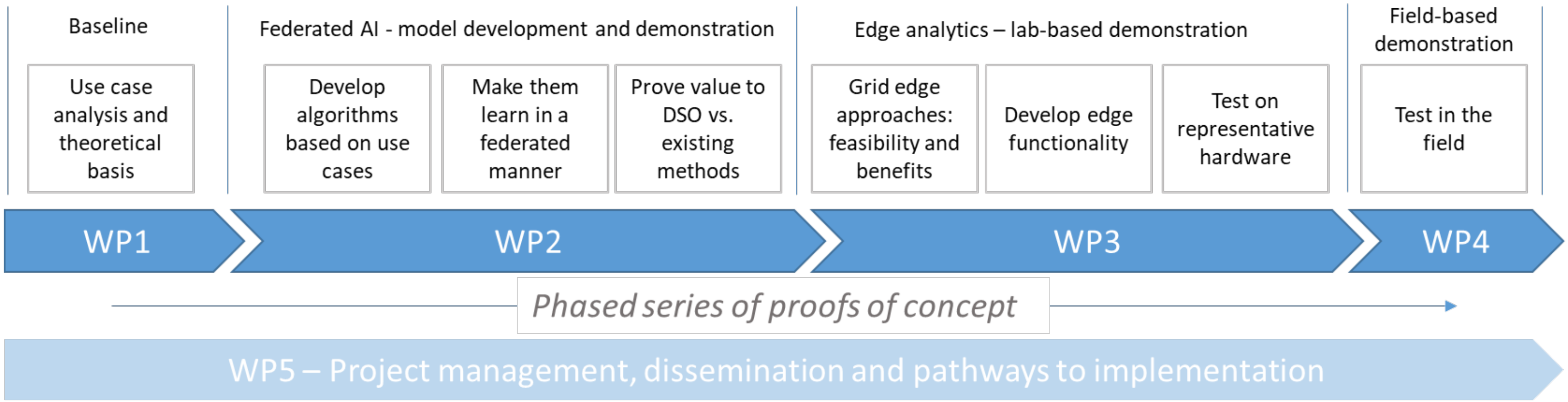
- Investigate the advantages of using a distributed (federated) analysis strategy to analyse smart meter data in multiple locations in particular for DSO use cases
- Investigate the feasibility of conducting advanced analysis of smart meter data at the grid edge

Goals:

- Prove the value of a distributed analytics approach in the use cases identified
- Prove the value of the AI algorithms
- Validate in a lab setting and through field tests
- Document the value created for utilities and their customers
- Extrapolate lessons learnt to further develop policy / regulation



General approach



Use cases under test

The DSO use cases are:

Use case 1: Detecting grid assets' loading for improved visibility of LV network congestion.

Use case 2: Predicting future demand profiles and load forecasts in the LV and MV network, so helping with grid management, congestion, and curtailment.

Use case 3: Detecting quality of supply (QoS) issues, network anomalies

Use case 4: Predicting network anomalies, quality of supply issues.

Use case 5: Supporting the implementation of localized tariffs or services.

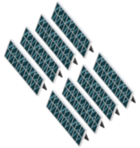
	Approach for KnowlEDGE				
Project component	Lit. study & SOTA analysis	Develop theoretical basis for	Model-based demonstration	Lab-based demonstration	Field-based demonstration
Federated analytics					
Edge computing approach					
Use case 1 – Detecting loading					
Use case 3 – Detecting QoS issues					
Use case 2 – Predicting loading					
Use case 4 – Predicting QoS issues					
Use case 5 – Localised tariffs					

Field activities

SCCER-FURIES Romande Energie ELectric network in local balance Demonstrator

Smart meter

Infrastructure: 700 smart meters



Forecasting

Infrastructure: 6 measurement boxes and 1 all-sky camera

GridEye advanced monitoring units

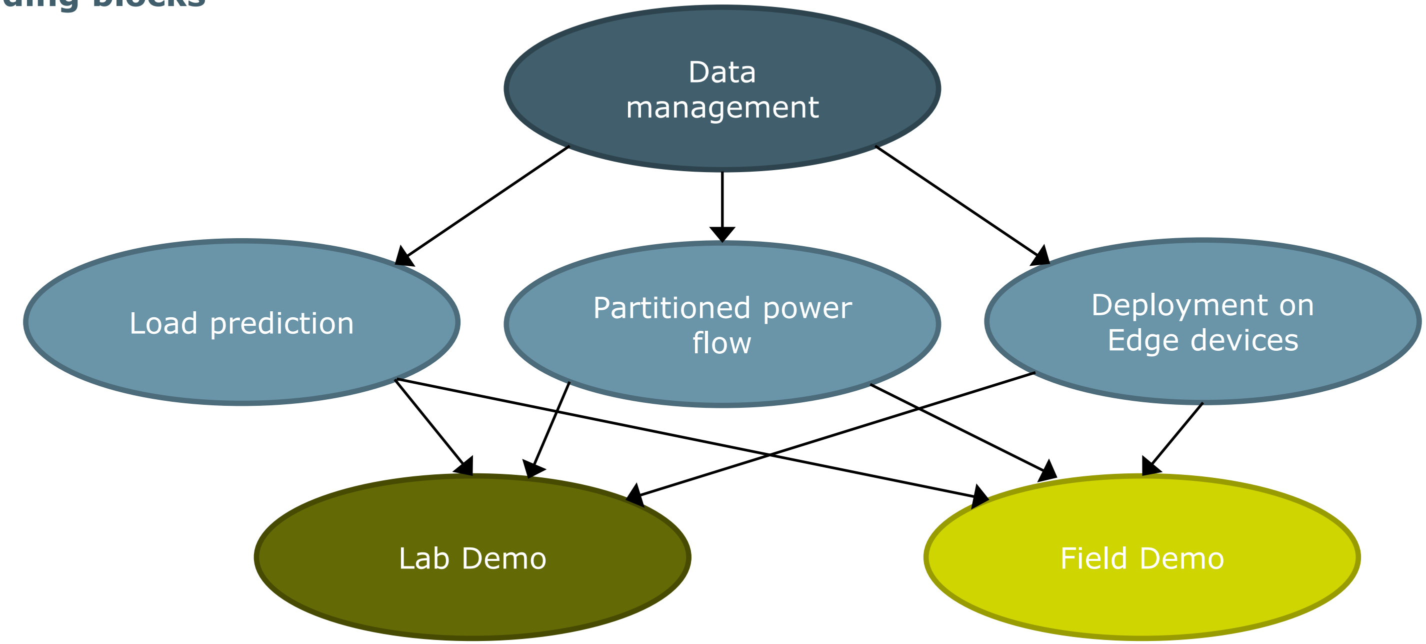
Infrastructure.:100 GridEye (LV)

Phasor Measurement Units (PMU)

Infrastructure.: 60 PMU (MV)

The focus for field testing will be use case 1 and 3.

Building blocks



Data management

One of the priorities:

Demonstrate development of algorithms/ML-models **without accessing the full data set:**

- End-consumer load profiles
- Full network information

1st phase: WP 2 (model development)

- Making historic data accessible for training purposes
- Making network data accessible for the power flow algorithm

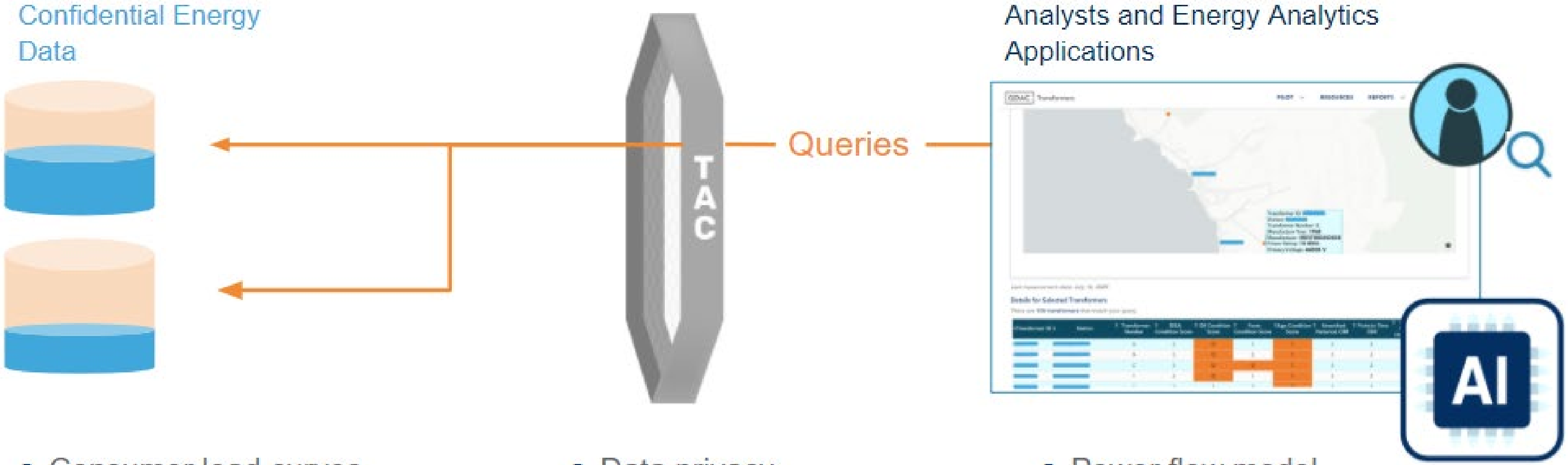
2nd phase: WP3/WP4 (Lab / field demonstration)

- Take the role of **global model computation** in federated approach
- **Orchestrate edge-devices** (provide network data of adjacent assets, orchestrate learning)

- Load data **never leaves smart meter** for congestion detection/prediction

Data management

Trusted Analytics Chain (TAC™) protects data privacy by **sending analyst queries** to data.

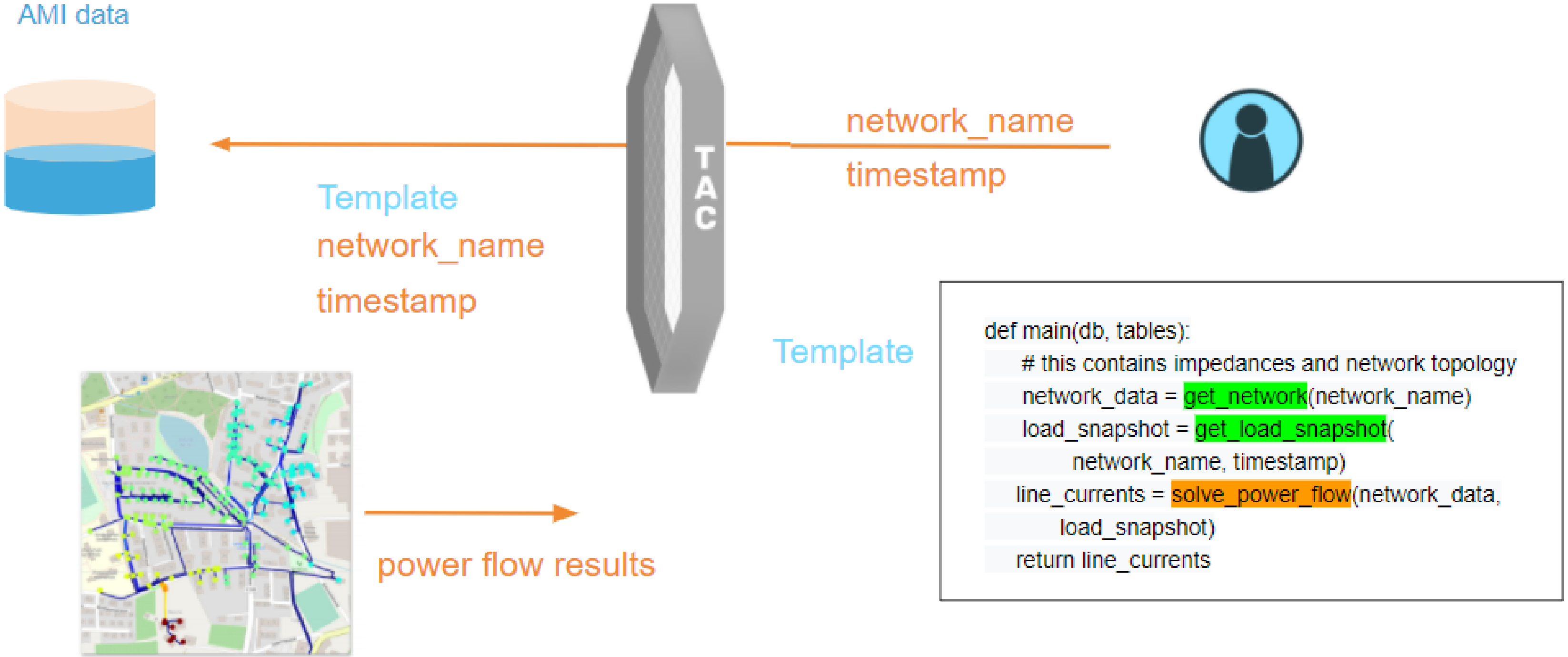


- Consumer load curves
- Grid connection points / Geo data

- Data privacy and security
- Automated data ingestion

- Power flow model
- Congestion calculation

Data management




```
def main(db, tables):
    # this contains impedances and network topology
    network_data = get_network(network_name)
    load_snapshot = get_load_snapshot(
            network_name, timestamp)
    line_currents = solve_power_flow(network_data,
            load_snapshot)
    return line_currents
```

Partitioned power flow

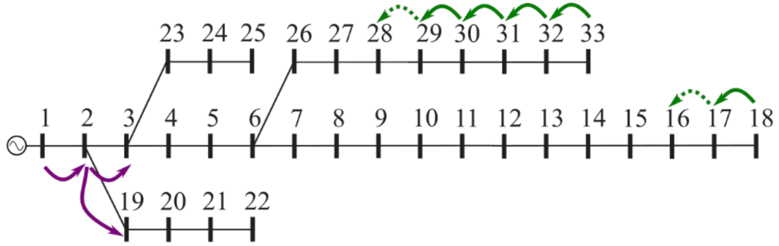
Goal: Obtain line congestion levels

Standard approach: Run load flow algorithm (typically at a centralized location)
- Requires full network data, and complete end-consumer load data
- May lead to computationally expensive solution (i.e. NR Jacobian inversion)
- **Privacy concerns!**

Proposed approach: Partitioned/distributed approach
- End-consumer load data never leaves smart meter
- Leverage smart meter computation resources
- Only line currents at adjacent branches and voltage levels need to be exchanged among edge-devices (smart meters)

Partitioned Power Flow Solution Algorithm

Current summation method



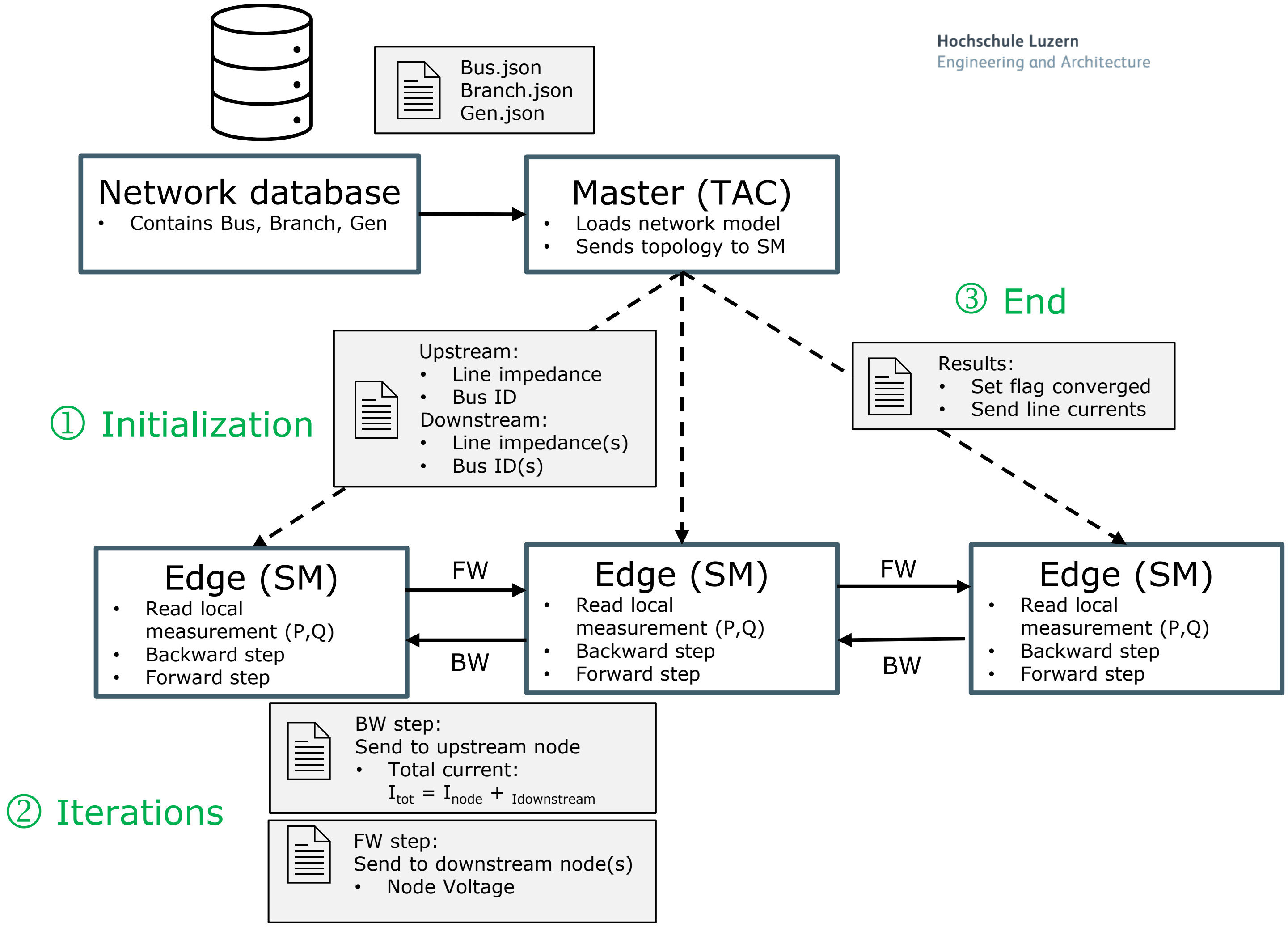
1) Backward sweep
   Calculate all branch currents
2) Forward sweep
   Calculate bus voltages based on branch currents
3) Iterate until convergence

Communication requirements
Adjacent meters
Power/current reading of downstream meters
Voltage reading of upstream meter

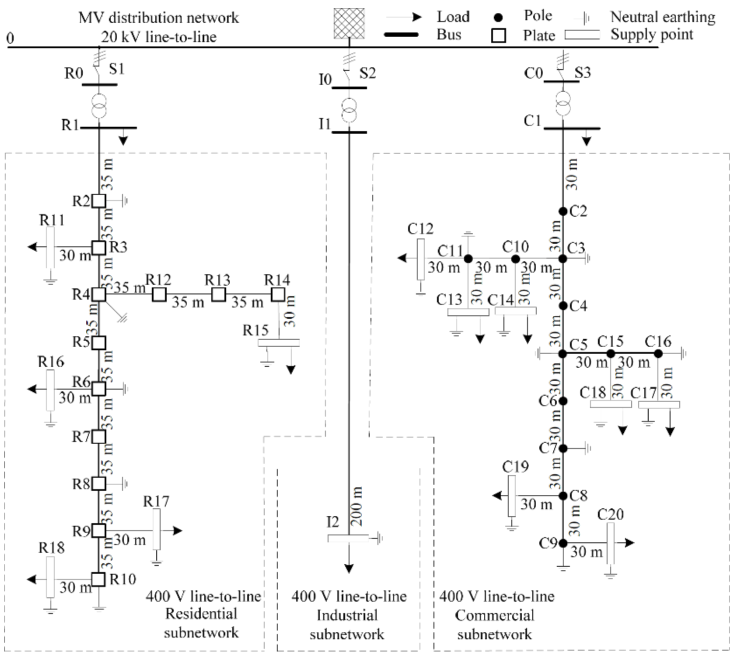
Status: Works on existing test models (IEEE & CIGRE) in Pandapower and within TAC

Next steps: Apply to RE test system with real load profiles
Distributed implementation ready for depolyment on edge-devices

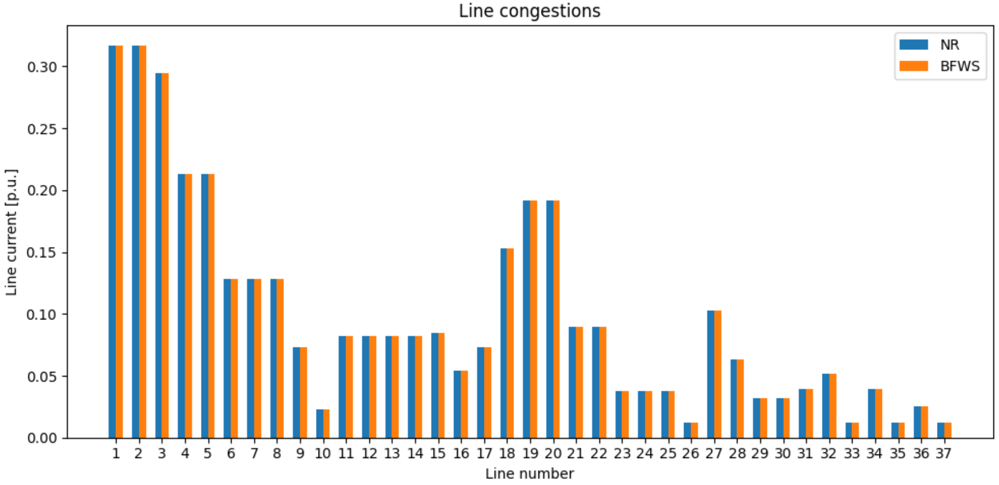
Communication flows



Partitioned Power Flow Solution Test Results



CIGRE LV Distribution Network



Snapshot solution of line currents

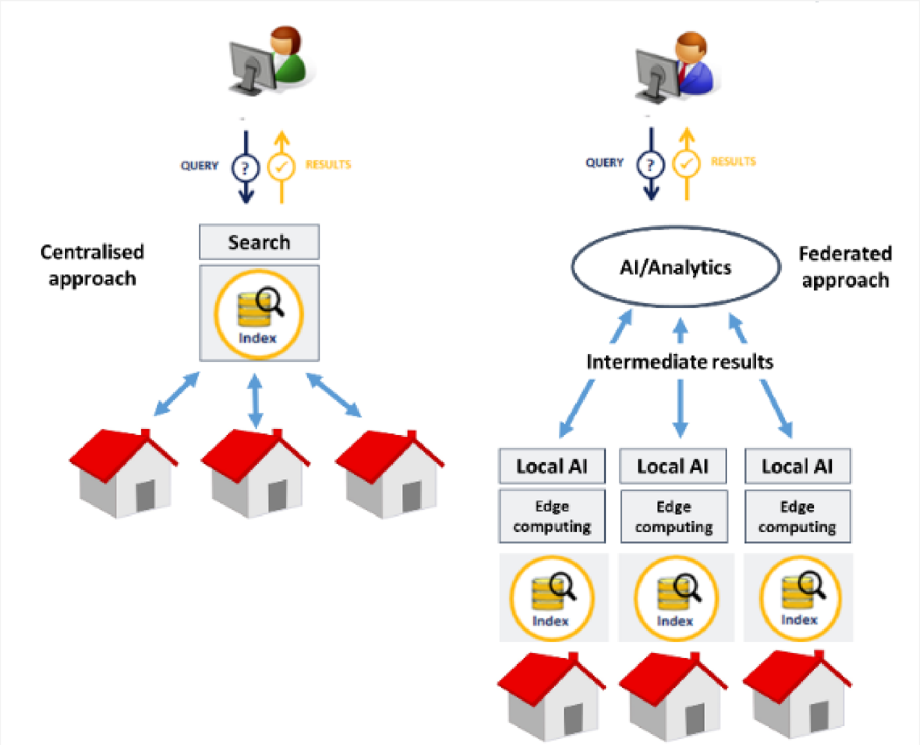
Execution times:
Newton Raphson: 0.01 s / 9 iterations
BFWS: 0.009s / 8 iterations

Load prediction

Goal: Decentralized, privacy-preserving approach for load prediction => **federated learning**

'Conventional' ML

- Nodes upload data to server
- Model training
- Actionable insight



'Federated' ML

- Nodes receive model from server
- Nodes start training
- Nodes send partially trained model to server
- Server combines models – 'federated model'
- Repeat!

Load prediction

Challenges:

- End-consumer load profiles are highly stochastic
- A common global model might be too complex to capture all the individual end-consumers
- Individual end-consumer models may lack data for training

Opportunities:

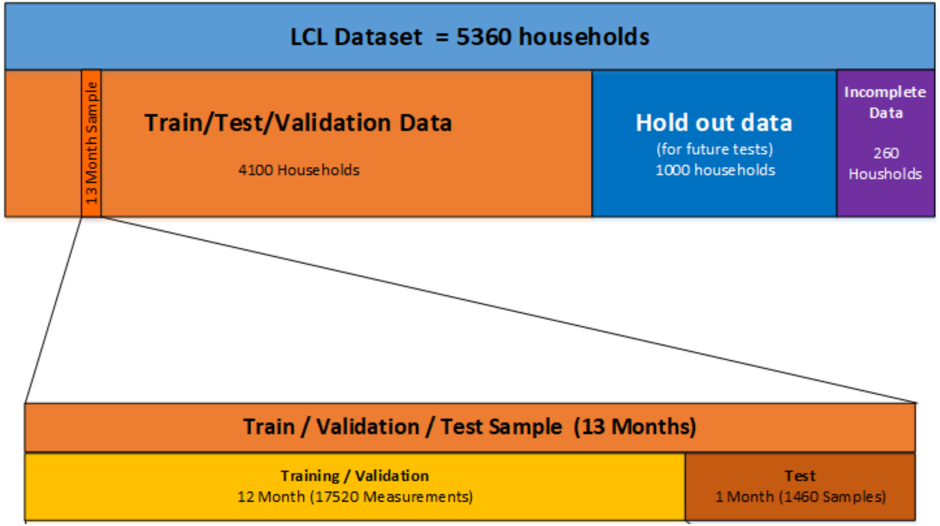
- Predicting probabilistic models
- Clustering of end-consumers with “similar” load patterns

- **Best of both worlds:** Federated approach derives a global “averaged” model and end-consumers specific models at the edge

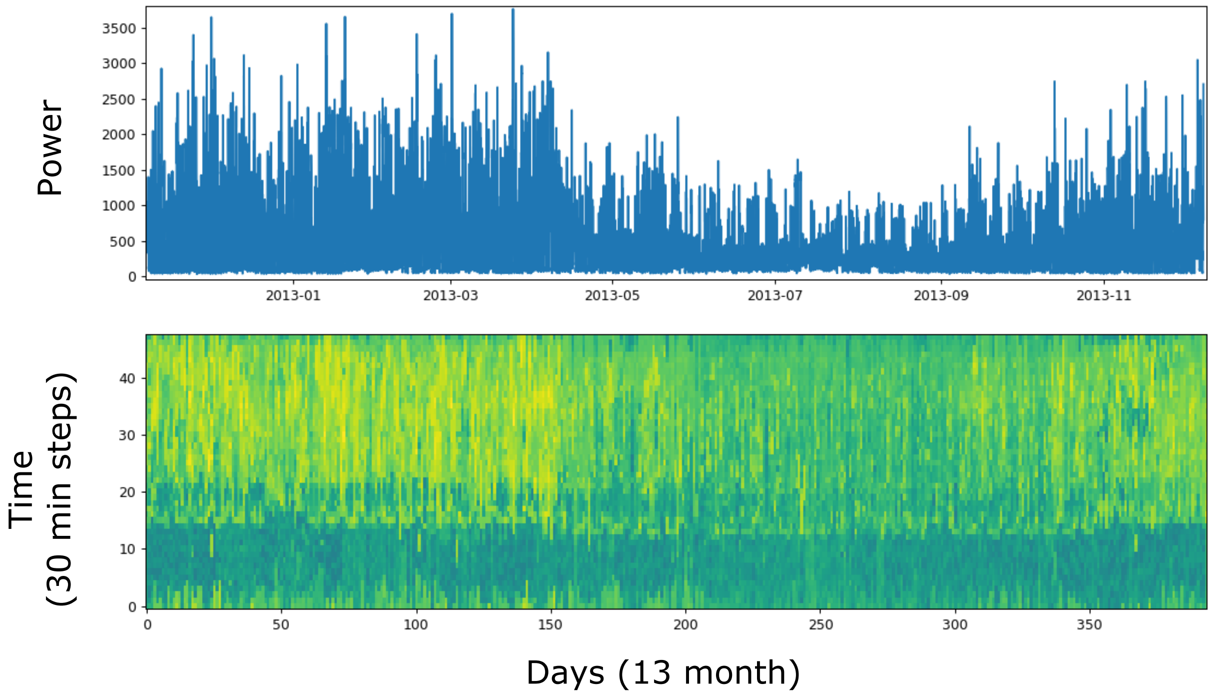
Load prediction

Sample data:

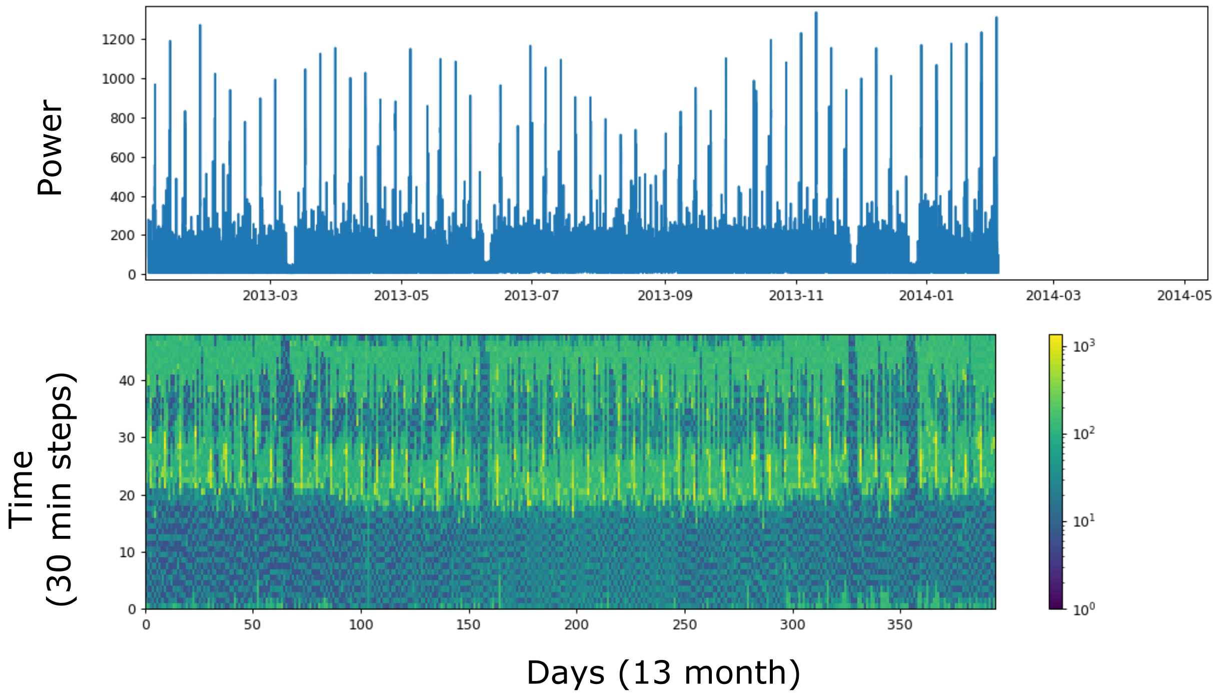
- Publicly available data set **Low-carbon-London** (LCL)
- **13 month / ~400** daily load profiles
- **30** minute intervals
- **12** month for training
- **1** month for testing



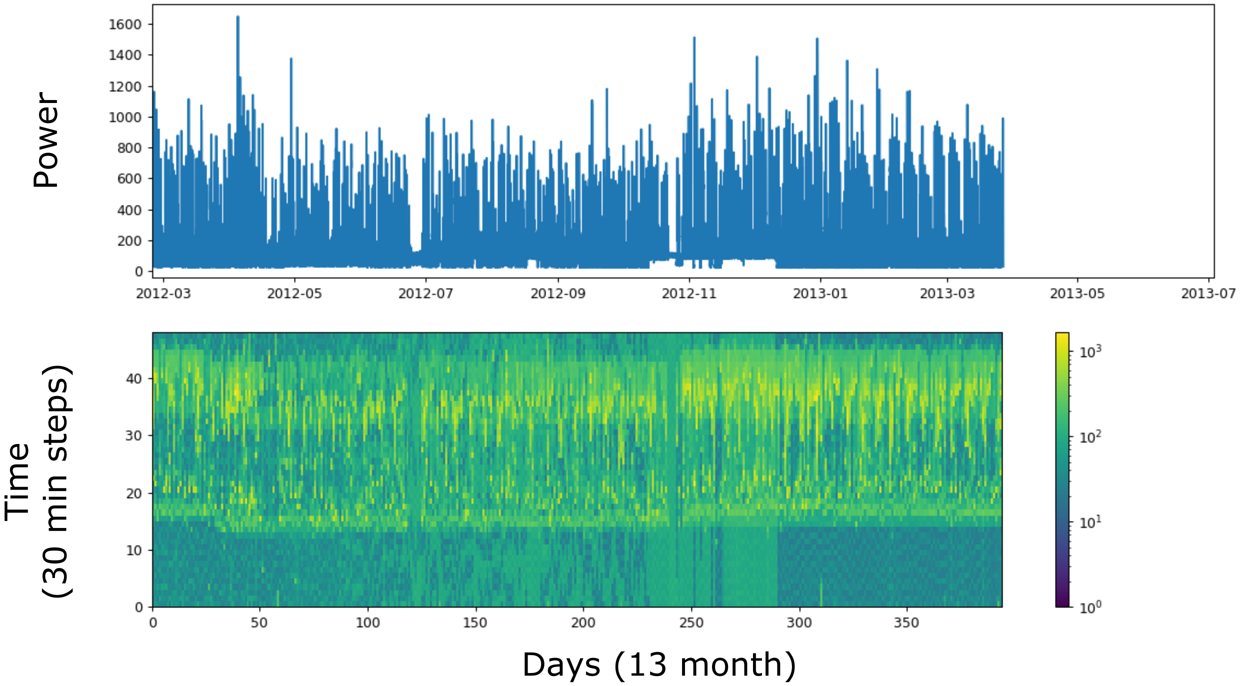
Load prediction: Low-carbon-London dataset (specific end-consumer)



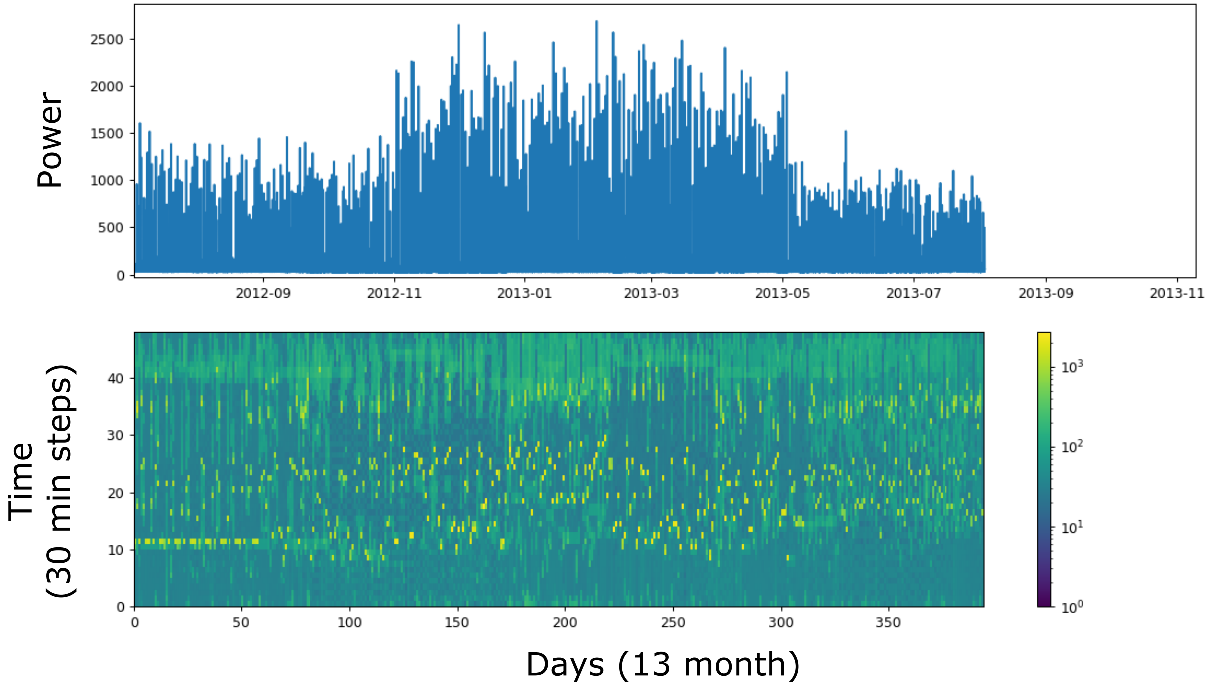
Load prediction: Low-carbon-London dataset (specific end-consumer)



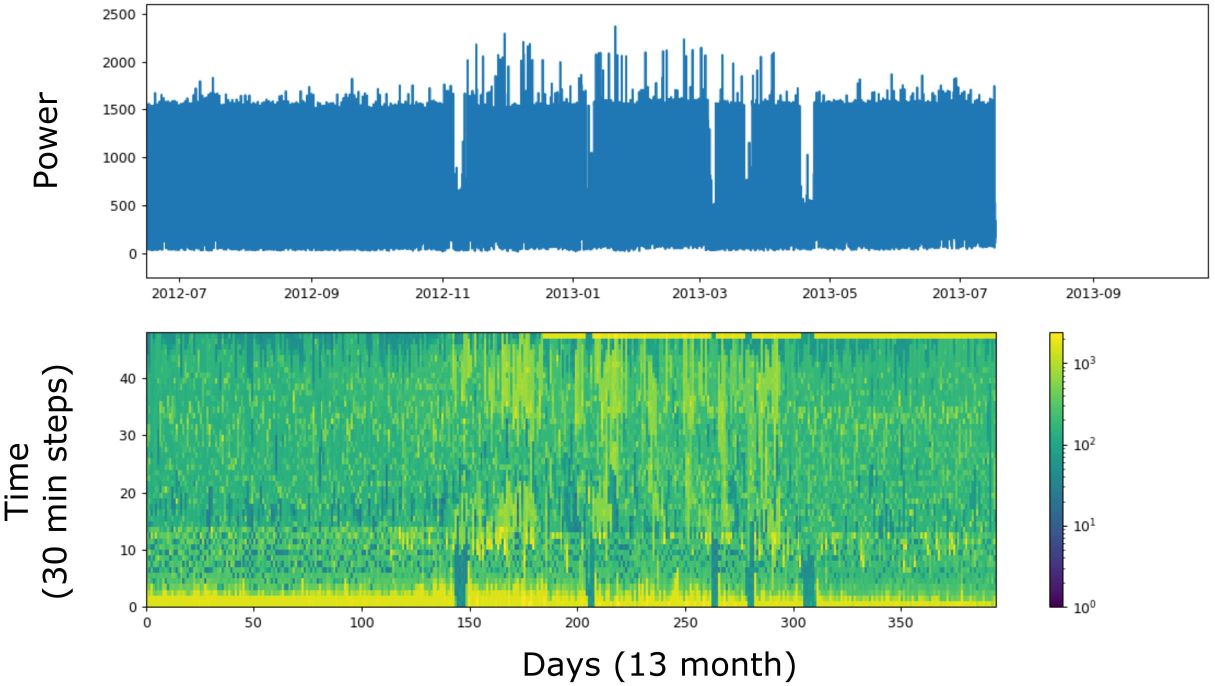
Load prediction: Low-carbon-London dataset (specific end-consumer)



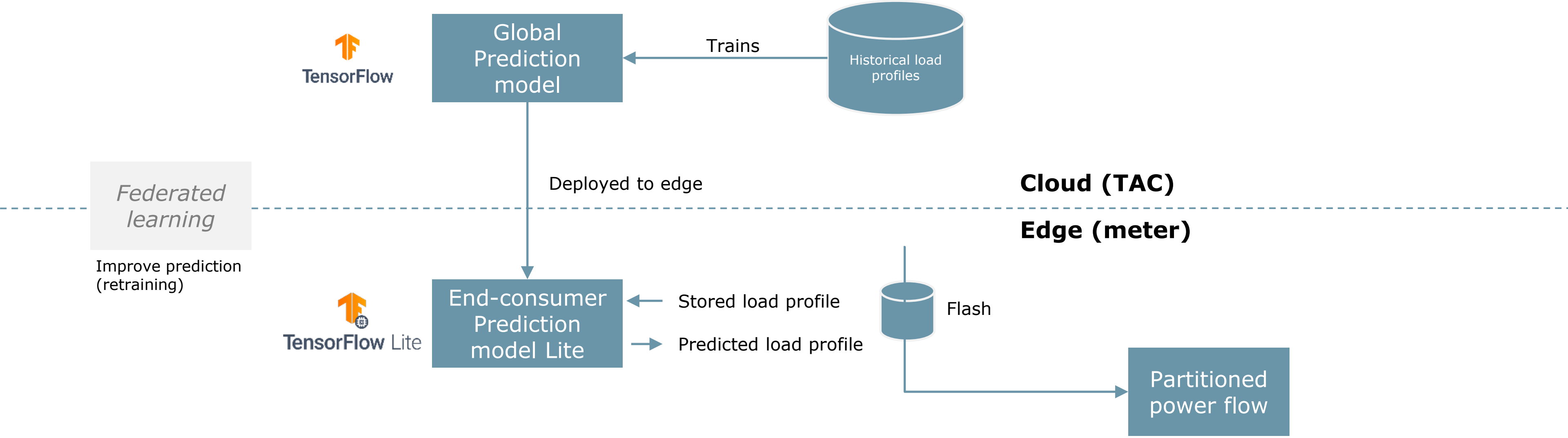
Load prediction: Low-carbon-London dataset (specific end-consumer)



Load prediction: Low-carbon-London dataset (specific end-consumer)



Load prediction



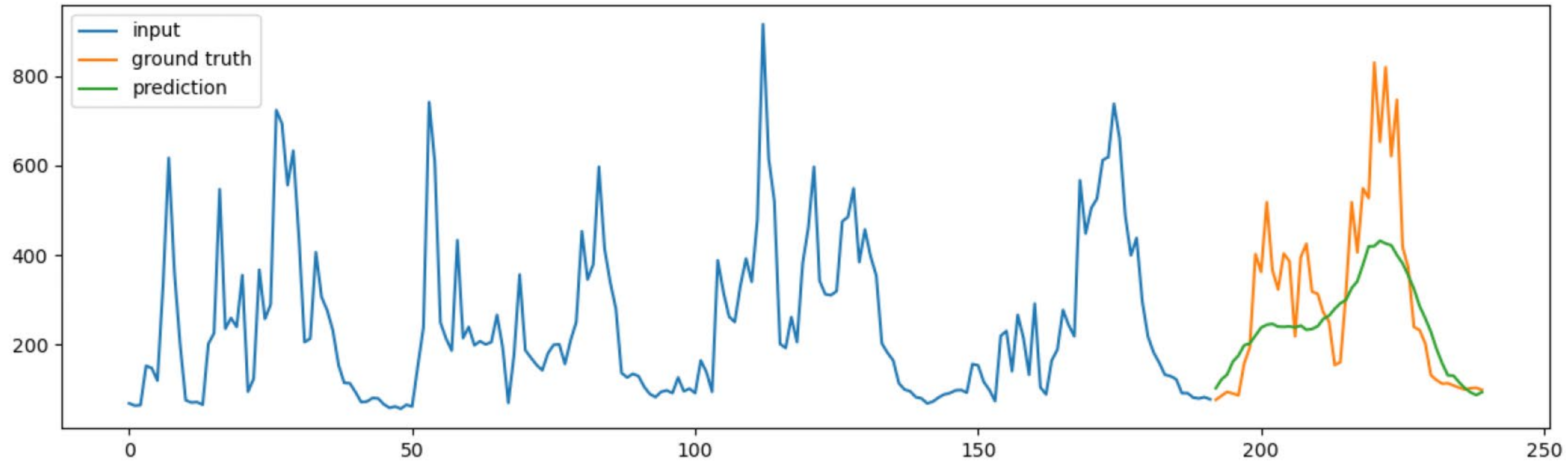
When we can predict end-consumer’s load profiles, we can also predict network congestions by feeding the predictions to the partitioned power flow.

Load prediction: Examples

Example settings:

- 359 days for training
- 7 days for validation
- Rest for testing (1 Month)
- Network: simple feed forward NN
- 6 layers, each with 50 units
- Currently no recursion

- Input window 4*48 values = 4 days
- Output window 48 values = 1 day

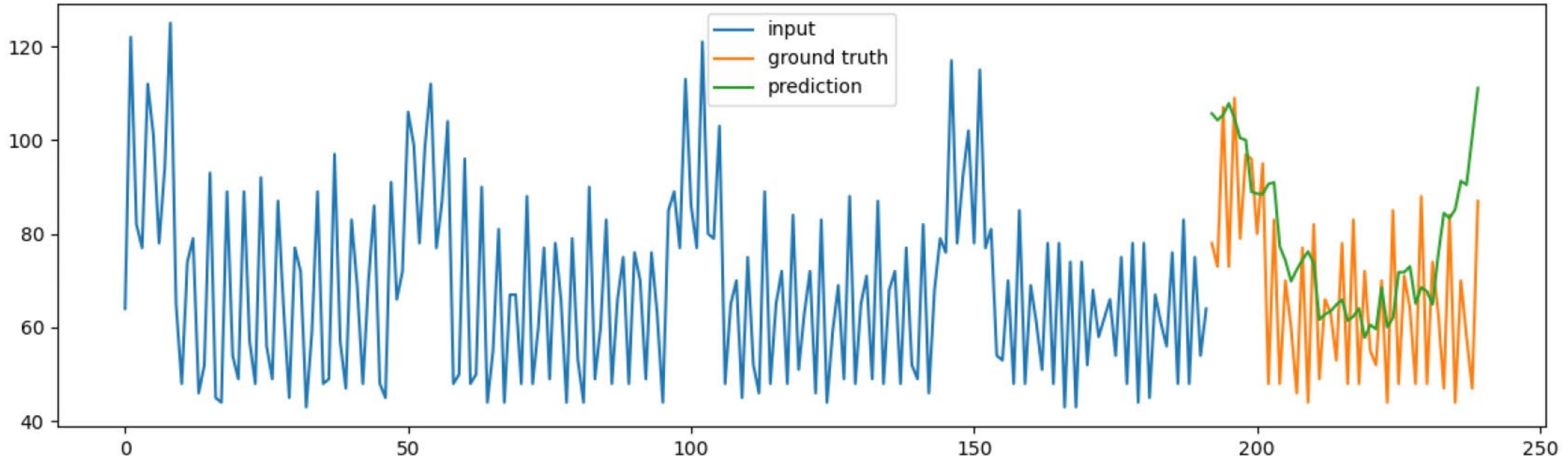


Load prediction: Examples

Example settings:

- ➢ 359 days for training
- ➢ 7 days for validation
- ➢ Rest for testing (1 Month)
- ➢ Network: simple feed forward NN
- ➢ 6 layers, each with 50 units
- ➢ Currently no recursion

- ➢ Input window 4*48 values = 4 days
- ➢ Output window 48 values = 1 day

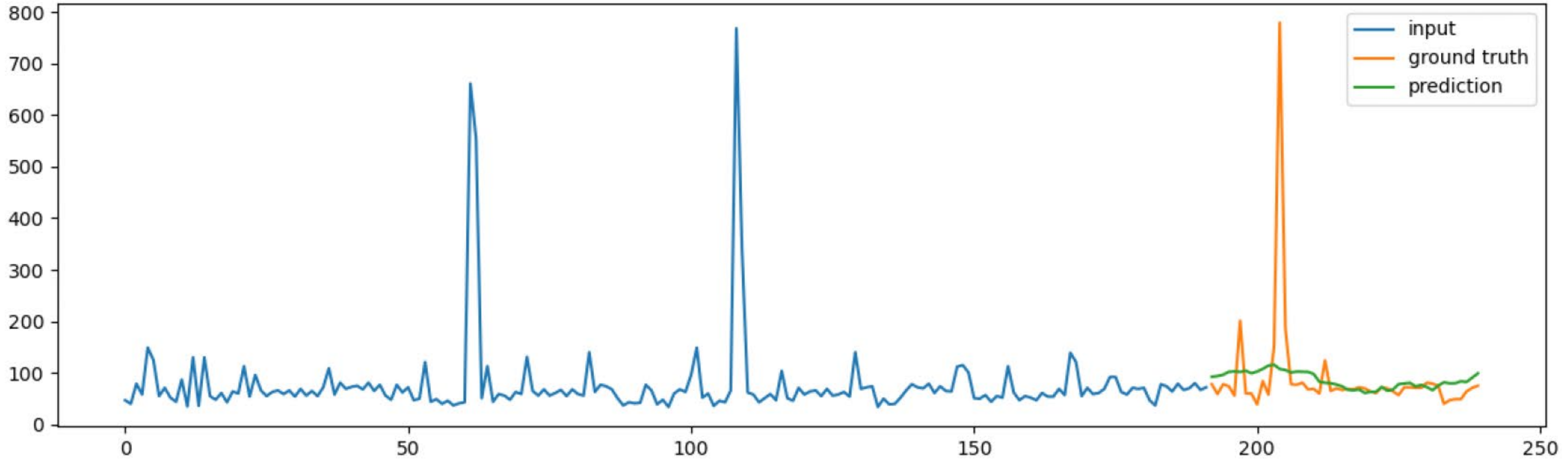


Load prediction: Examples

Example settings:

- 359 days for training
- 7 days for validation
- Rest for testing (1 Month)
- Network: simple feed forward NN
- 6 layers, each with 50 units
- Currently no recursion

- Input window 4*48 values = 4 days
- Output window 48 values = 1 day



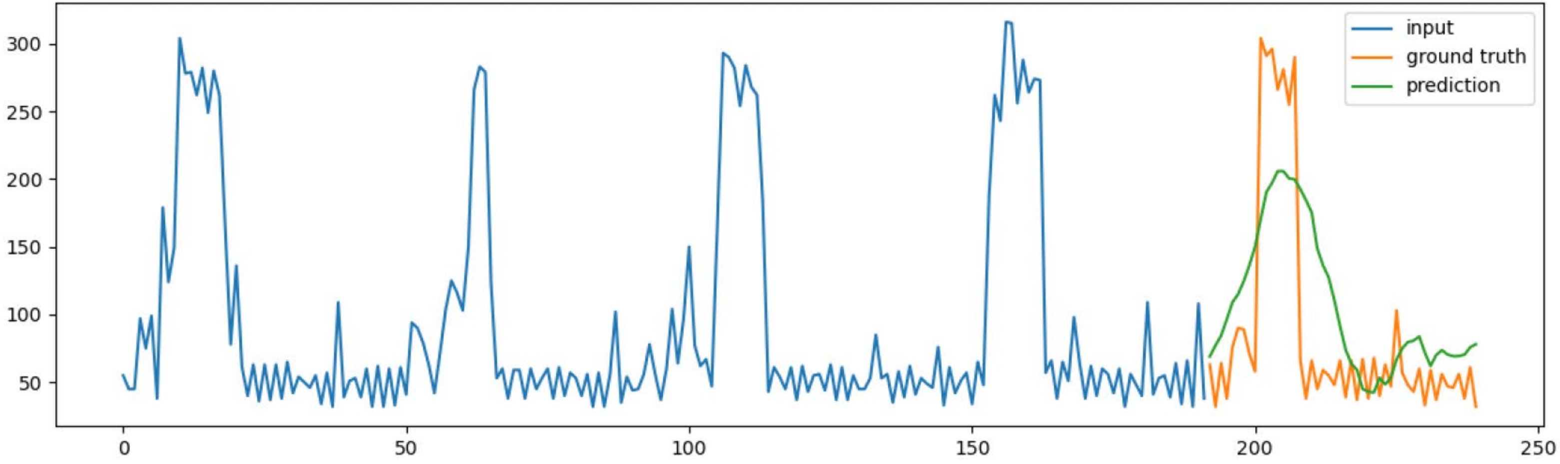
Load prediction: Examples

Example settings:

- 359 days for training
- 7 days for validation
- Rest for testing (1 Month)
- Network: simple feed forward NN
- 6 layers, each with 50 units
- Currently no recursion

- Input window 4*48 values = 4 days
- Output window 48 values = 1 day

Next steps: Federate approach/fine tune models/ use different model



Deployment on Edge devices

Challenges:

- Minimal power consumption of meters
- Limited computational resources and memory
- Large neural networks may not be stored/executed on edge devices

Opportunities:

- Limit size of ML-models (number of layers/neurons)
- Use ML-models with Integer numbers (less memory)
- Device capabilities are increasing – we can do more and more at the edge

Deployment on Edge devices: Hardware capabilities

NDA

Project partners

Research contacts

Open data
Open source meters
Online resources
Papers

Parameter	Low Specs	Medium Specs	High Specs
What?	This is the basic set of parameters – the minimum set for smart meter functionality	**This is the base case for KnowlEDGE. We will assume this configuration in KnowlEDGE**	This is the set of enhanced parameters of possible Edge Devices. They may be used for high demand applications
Typical Processors	ATmega168 ATMega328 ARM Cortex M0 ARM Cortex M4	**ARM Cortex M7** **Infineon Aurix TC275TP**	(Linux Processors) Atheros AR9331 Pi 3A + Pi 4B
Processor Frequency	...160 MHz	**200....600 MHz**	400 MHz ... 1.6GHz
RAM	0.5kB...96kB	**550kB ... 8 MB**	16 MB ... 8 GB
Flash	32-512 kB	**256 kB... 8 MB**	...64 MB (SD Card on RasPi)
SoC examples	Arduino Nano Arduino Mega TinyK22 TinyK20 Raspberry Pi Pico	**Kinetis KV5x Family (*159 CHF)** **Teensy 4.0** **Teensy 4.1**	Raspberry Pi Raspberry Pi 4 Arduino Yún
Language	C / C++	**C / C++**	Linux
Price Range	5 ... 20 CHF	**25 ... 35 CHF**	35 ... 50 CHF

Deployment on Edge devices: Hardware capabilities

Edge-device employed before deploying to smart meters:

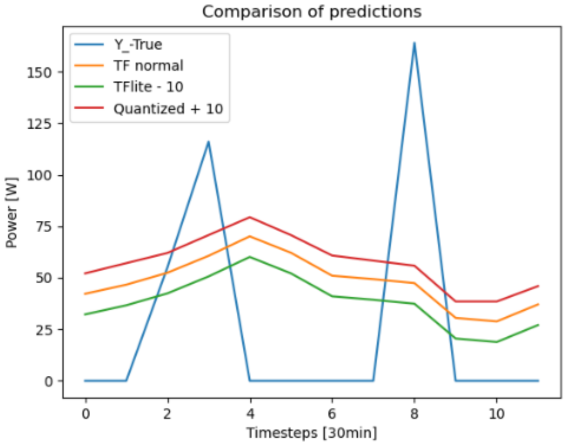
- Arduino Nano 33BLE with Tensorflow Lite support
- 64MHz/ 1MB Flash/ 256KB RAM
- CHF 20.00
- Low/medium spec

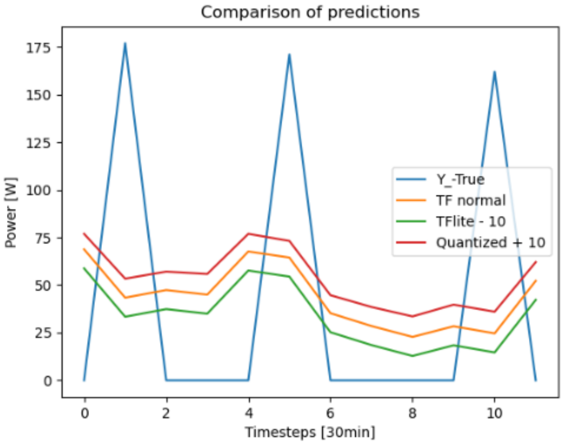
Microcontroller	nRF52840 (datasheet)
Operating Voltage	3.3V
Input Voltage (limit)	21V
DC Current per I/O Pin	15 mA
Clock Speed	64MHz
CPU Flash Memory	1MB (nRF52840)
SRAM	256KB (nRF52840)

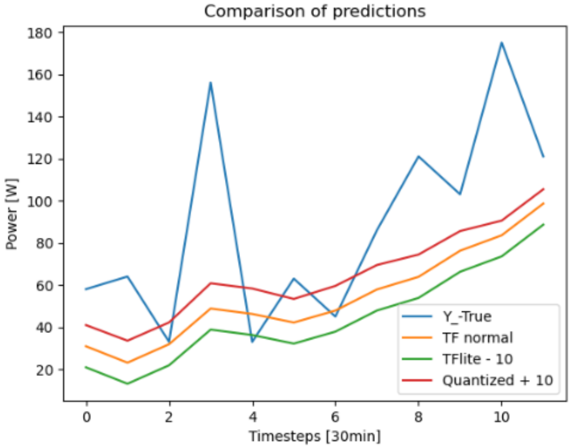
Deployment on Edge devices: TF vs. TFLite model

TF: Full global model (in Python)
TF Lite: Conversion to hex file
Quantized: All Int8 values (1 byte), ready for deployment on edge device

(disregard prediction accuracy, focus on “deployment on edge”)







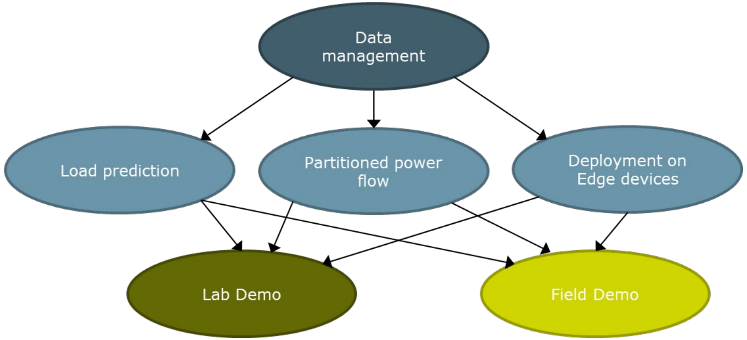
Summary - Next steps:

Data management:
- Include load data and network topology from RE
- Enable development with actual data from RE

Partitioned power flow:
- Implement in distributed «simulated» environment

Load prediction:
- Federate learning approach
- Personalization of local models (end-consumer specific)
- Deal with local conditions/overfitting

Deployment on edge devices:
- Deploy on actual smart meters together with industrial partner Landis+Gyr



Summary – Take away:

Area of great opportunity – high regulator and utility interest

Device capabilities are increasing – we can do more and better at the edge

Slow process to develop trust and confidence, but KnowlEDGE will set a benchmark for Swiss market

Thanks for your attention!

Happy to answer your questions online or offline.