A NOVEL DNN-HMM-BASED APPROACH FOR EXTRACTING SINGLE LOADS FROM AGGREGATE POWER SIGNALS

Universität Stuttgart

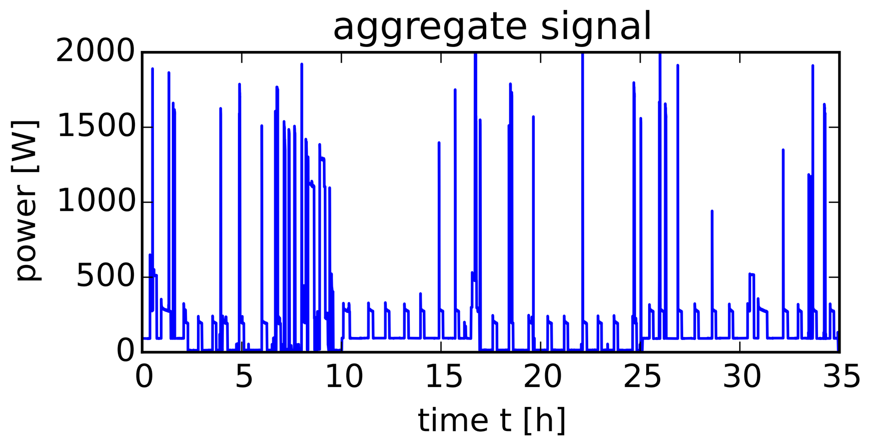
Lukas Mauch and Bin Yang

Institute of Signal Processing and System Theory, University of Stuttgart, Germany

1. Introduction

1.1 Non-intrusive load monitoring

- Estimate power consumed by individual loads from aggregate power signal
- Aggregate signal $x(n) = \sum_{k=1}^{M} x_k(n),\ x_k(n) \in \mathbb{R}^+$ consists of real power measurements of up to hundreds of loads



- Single loads are non-stationary with strong temporal relations from seconds to many hours
- Established techniques for single channel source separation fail

1.2 Contribution

- Combination of Deep Neural Network (DNN) and Hidden Markov Model (HMM) for signal extraction from single channel measurements

2. DNN-HMM Network

2.1 Idea

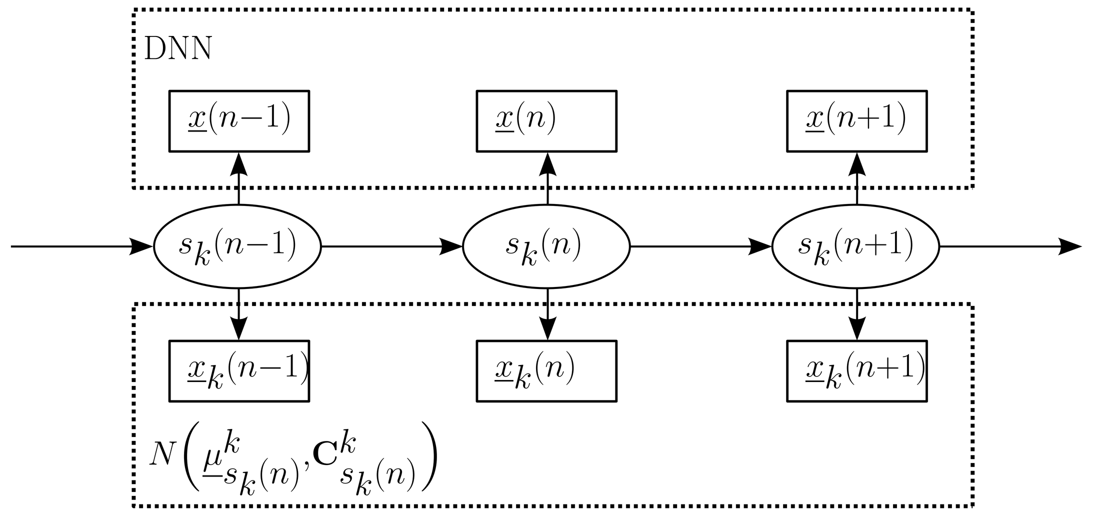


Abbildung 1: *The DNN-HMM network for signal extraction*

- Use DNN-HMM as generative model for
  - Observation sequence of aggregate signal $\{\underline{x}\} = \{\underline{x}(1), \ldots, \underline{x}(N)\}$
  - Observation sequence of single load $\{\underline{x}_k\} = \{\underline{x}_k(1), \ldots, \underline{x}_k(N)\}$
  - Hidden state sequence $\{s_k\} = \{s_k(1), \ldots, s_k(N)\}$
- Train one DNN-HMM for each target load contained in aggregate signal
- For each load, the HMM captures the temporal relations in the signal. The DNN is used to infer the state sequence from the aggregate signal.

2.2 Architecture of HMM

- Parametrization of the HMM of load k

$$
\begin{aligned}
s_k(n) &\in \{1, \cdots, M_k\} \\
\underline{\pi}^k &= [P(s_k(1)=1), \ldots, P(s_k(1)=M_k)]^T \\
p^k_i &= p(\underline{x}_k(n)|s_k(n)=i) \sim N(\underline{\mu}^k_i, \mathbf{C}^k_i) \\
\tilde{p}^k_i &= p(\underline{x}(n)|s_k(n)=i) \\
\mathbf{A}^k &= [P(s_k(n)=i|s_k(n-1)=j)]_{ij}
\end{aligned}
$$

- The joint probability distribution of the sequences is

$$
\begin{aligned}
p(\{s_k\}, \{\underline{x}_k\}, \{\underline{x}\}) = &\ \underline{\pi}^k_{s_k(1)} p^k_{s_k(1)} \tilde{p}^k_{s_k(1)} \\
&\cdot \prod_{n=2}^{N} p^k_{s_k(n)} \tilde{p}^k_{s_k(n)} \mathbf{A}_{s_k(n), s_k(n-1)}
\end{aligned}
$$

2.3 Architecture of DNN

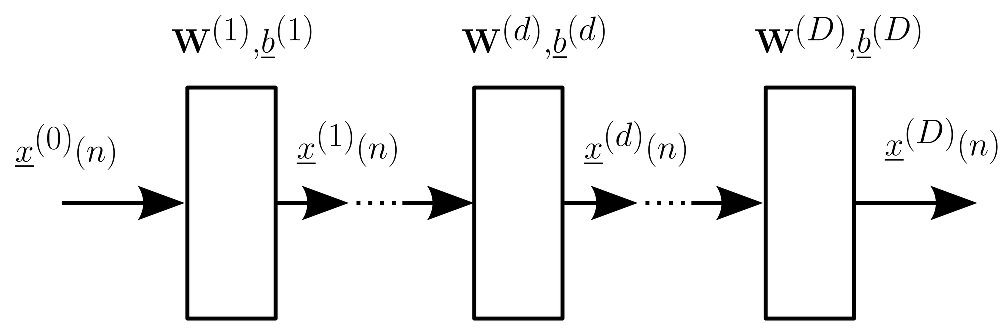


Abbildung 2: *Architecture of the D-layer DNN*

- DNN can model very complex conditional density functions
  - Input: current observation plus context in block of length B

$$\underline{x}^{(0)}(n) = [\underline{x}^T(n-B), \ldots, \underline{x}^T(n+B)]^T$$

  - Mapping:

$$
\begin{aligned}
\underline{a}^{(d)}(n) &= \mathbf{W}^{(d)} \underline{x}^{(d-1)}(n) + \underline{b}^{(d)},\ 1 \le d \le D \\
\underline{x}^{(d)}(n) &= \Phi^{(d)}(\underline{a}^{(d)}(n))
\end{aligned}
$$

  - Output:

$$
\begin{aligned}
\underline{x}^{(D)}(n) &= [f_1(\underline{x}^{(0)}(n)), \ldots, f_{M_k}(\underline{x}^{(0)}(n))]^T \\
f_i(\underline{x}^{(0)}(n)) &= P(s_k(n)=i|\underline{x}^{(0)}(n))
\end{aligned}
$$

- The DNN is trained to approximate

$$\tilde{p}_{s_k(n)} \sim \frac{P(s_k(n)|\underline{x}^{(0)}(n))}{P(s_k(n))}$$

2.4 Inference and signal extraction

- Infer the state sequence from the aggregate signal, using the state posteriors estimated by DNN

$$P(\{s_k\}|\{\underline{x}\}, \underline{\lambda}_k) = \frac{1}{Z} \underline{\pi}^k_{s_k(1)} \tilde{p}^k_{s_k(1)} \prod_{n=2}^{N} \tilde{p}^k_{s_k(n)} \mathbf{A}_{s_k(n), s_k(n-1)}$$

- Reconstruct the power consumed by load k from the state sequence

$$\hat{\underline{x}}_k(n) = \underline{\mu}^k_{s_k(n)}$$

2.5 Supervised training

- For a given training sequence $\underline{x}^t_k$ of load k, maximize

$$p(\{\underline{x}^t_k\}|\underline{\lambda}_k) = \sum_{\{s_k\}} \underline{\pi}^k_{s_k(1)} p^k_{s_k(1)} \prod_{n=2}^{N} p^k_{s_k(n)} \mathbf{A}_{s_k(n), s_k(n-1)}$$

  over parameters $\underline{\lambda}_k$ of the DNN-HMM
- Infer the most likely state sequence $\{\hat{s}^t_k\}$ for given $\{\underline{x}^t_k\}$ and all possible $\{\underline{x}^t\}$
- Optimize over the DNN parameters in order to minimize

$$
\begin{aligned}
J(\{\underline{x}^t_k\}, \{\underline{x}^t\}) &= -\sum_{n=1}^{N} \log\left(\sum_{i=1}^{M_k} \mathbf{A}_{i, \hat{s}^t_k(n-1)} \tilde{p}^k_i p^k_i \mathbf{A}_{\hat{s}^t_k(n+1), i}\right) \\
&= -\log \prod_{n=1}^{N} p(\underline{x}^t_k(n), \underline{x}^t(n)|\hat{s}^t_k(n-1), \hat{s}^t_k(n+1))
\end{aligned}
$$

3. Experiments and results

- Extract the major loads fridge (FR), dishwasher (DW), microwave (MW) and kitchen outlets (KO)
- Reference Energy Disaggregation Dataset (REDD) containing 18 loads used

$$E_k = \frac{1}{F_s} \sum_{n=1}^{N} x_k(n)$$

$$\hat{E}_k = \frac{1}{F_s} \sum_{n=1}^{N} \hat{x}_k(n)$$

$$\mathrm{NAD} = \frac{\sum_{n=1}^{N} |\hat{x}_k(n) - x_k(n)|}{\sum_{n=1}^{N} |x_k(n)|}$$

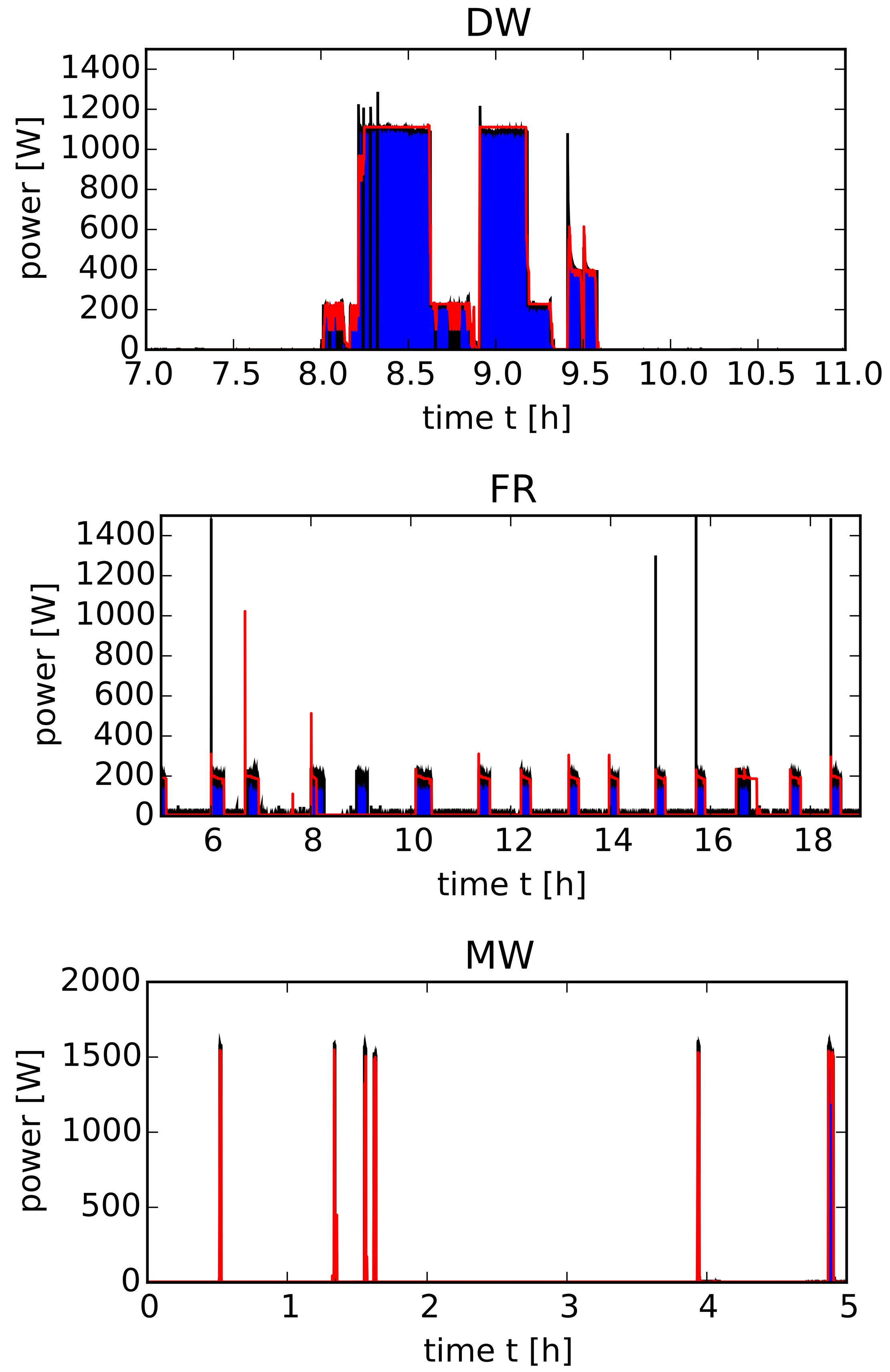


Abbildung 3: *Ground truth (filled blue) and extracted (red) signals.*

Appl.	E_t	$\hat{E}_t$	NAD	Gain
FR	4.30	4.26	0.14	10.5
DW	0.93	0.94	0.08	21.1
MW	1.66	1.49	0.27	13.1
KO	0.35	0.34	0.22	21.0

Tabelle 1: *Power based metrics*

ICASSP 2016, March 20-25, Shanghai, China