The Effect of Data Granularity on Load Data Compression

Andreas Unterweger[1], Dominik Engel[1] and Martin Ringwelski[2]

[1] Salzburg University of Applied Sciences,
Josef Ressel Center for User-Centric Smart Grid Privacy, Security and Control
[2] Technische Universität Hamburg-Harburg, Institut für Telematik

November 12, 2015

Motivation I

- Smart metering leads to a lot of load data
  - Millions of households (40 mio. in Germany)
  - 15-minute granularity typical

  $\rightarrow$ 25 TiB of load data per day (in Germany)
- Load data has to be transmitted
  - Low-bandwidth comm. links are problematic (e.g., PLC)

  $\rightarrow$ **Compression** to the rescue!
  - Side effect: Saves storage costs as well

Motivation II

- Can load data even be compressed?
  - Literature promises 90% reduction (compression ratio 10:1) and more
  - Commonly used (TUD, MIT) vs. real-world data

  → **Can real-world load data be compressed considerably?**
- Impact of data granularity
  - 1-second granularity required for some applications
  - 1- to 15-minute granularity typical

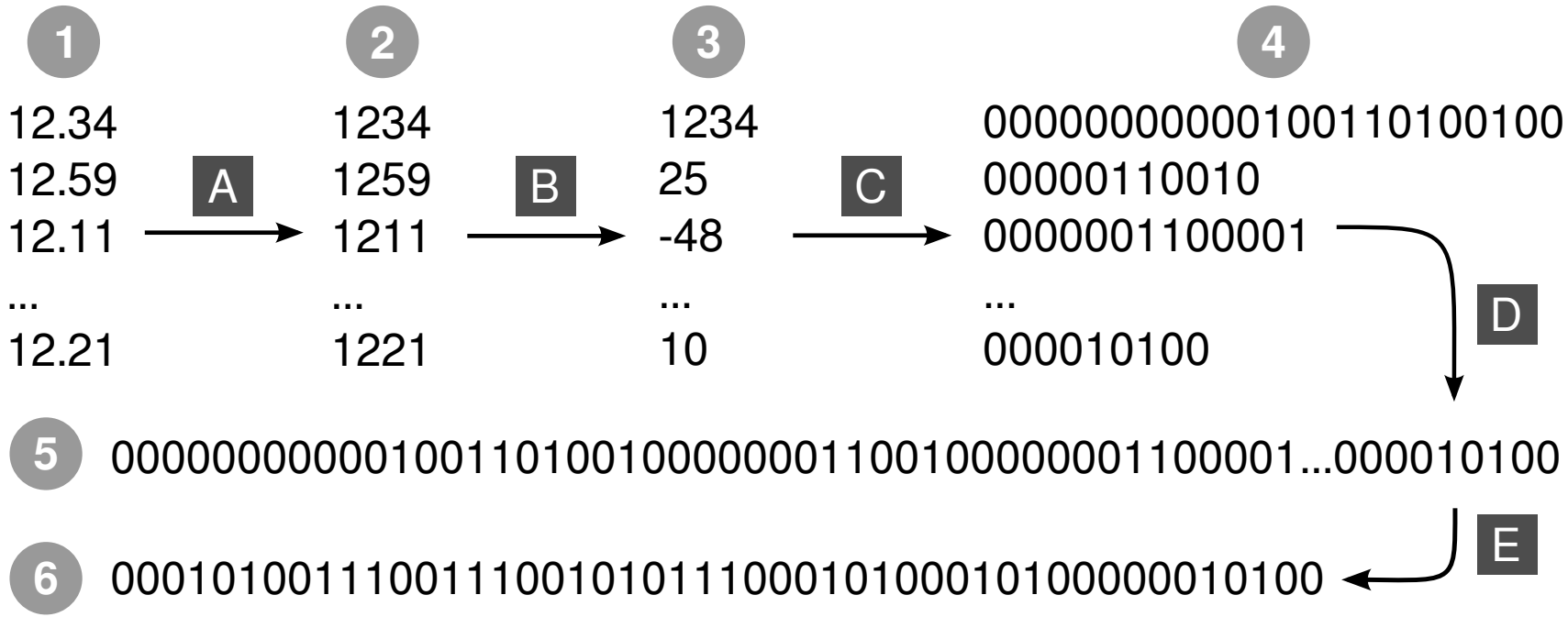
  → **How much does data granularity impact compression?**

Reference algorithms (uncompressed)

- IEC 62056-21
  - ASCII coding of base 10 representation
  - 123.45 becomes `00110001 00110010 00110011 00101110 00110100 00110101`
  - Additional separating character (one byte) between values required
- A-XDR (also referred to as IEC 61334-6)
  - Fixed-length fixed-point base 2 representation
  - 123.45 becomes `00110000 00111001` (12345)
  - No delimiter required

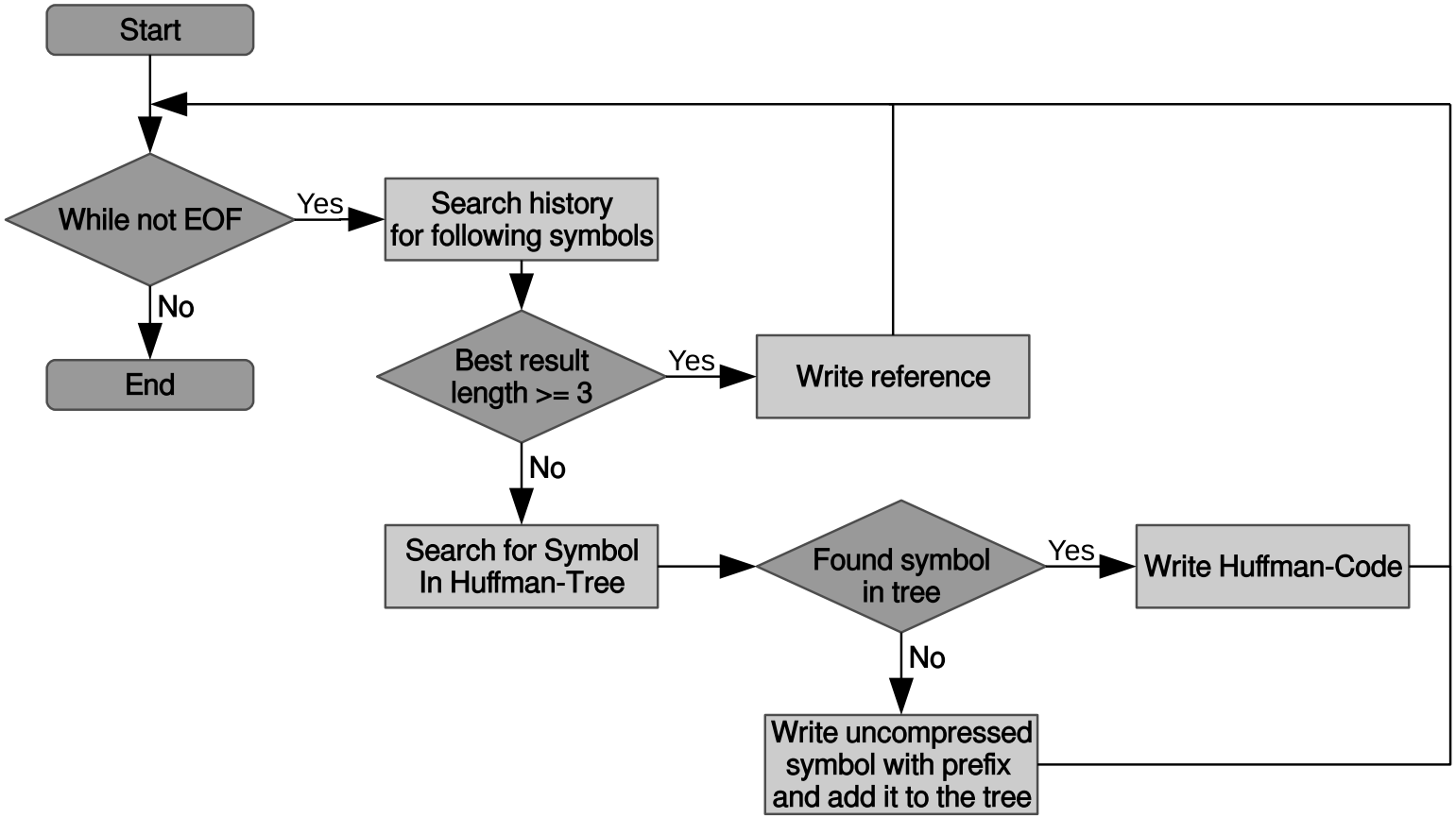
Compression algorithms I

- DEGA coding (Unterweger and Engel, 2015)



Compression algorithms II

- LZMH coding (Ringwelski et al., 2012)



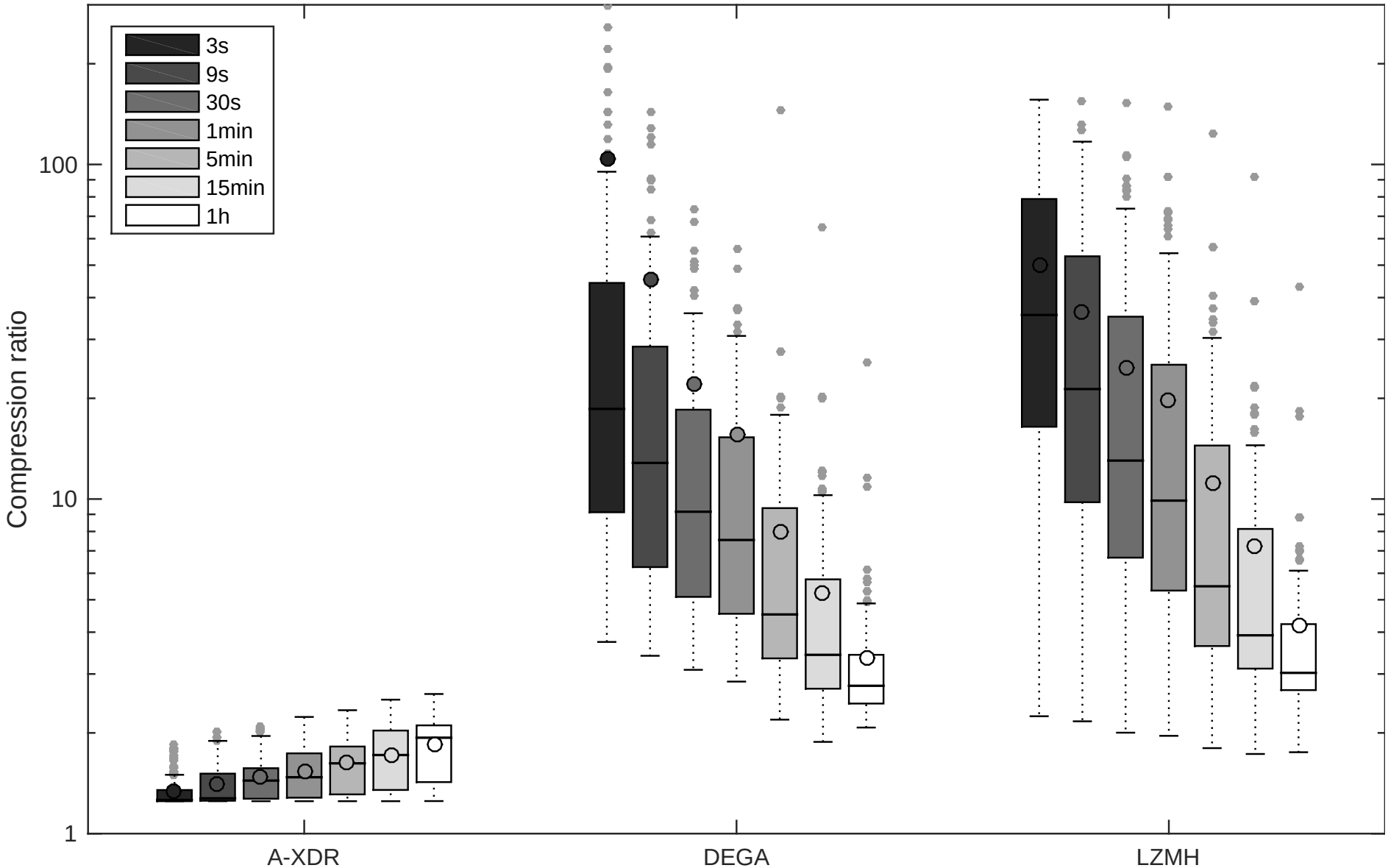
Load data sets

- MIT REDD
  - 6 houses with between 11 and 26 channels (116 total)
  - Average apparent power readings in Watts with two decimal places
  - 3-second granularity (mains have 1-second granularity)
  - Coverage of between 2.7 and 25.8 days
- SAG (provided by our partner Salzburg AG)
  - 508 households and industrial plants (mains only)
  - Accumulated energy readings in kWh with three decimal places
  - 15-minute granularity
  - Coverage of one year (365 days)

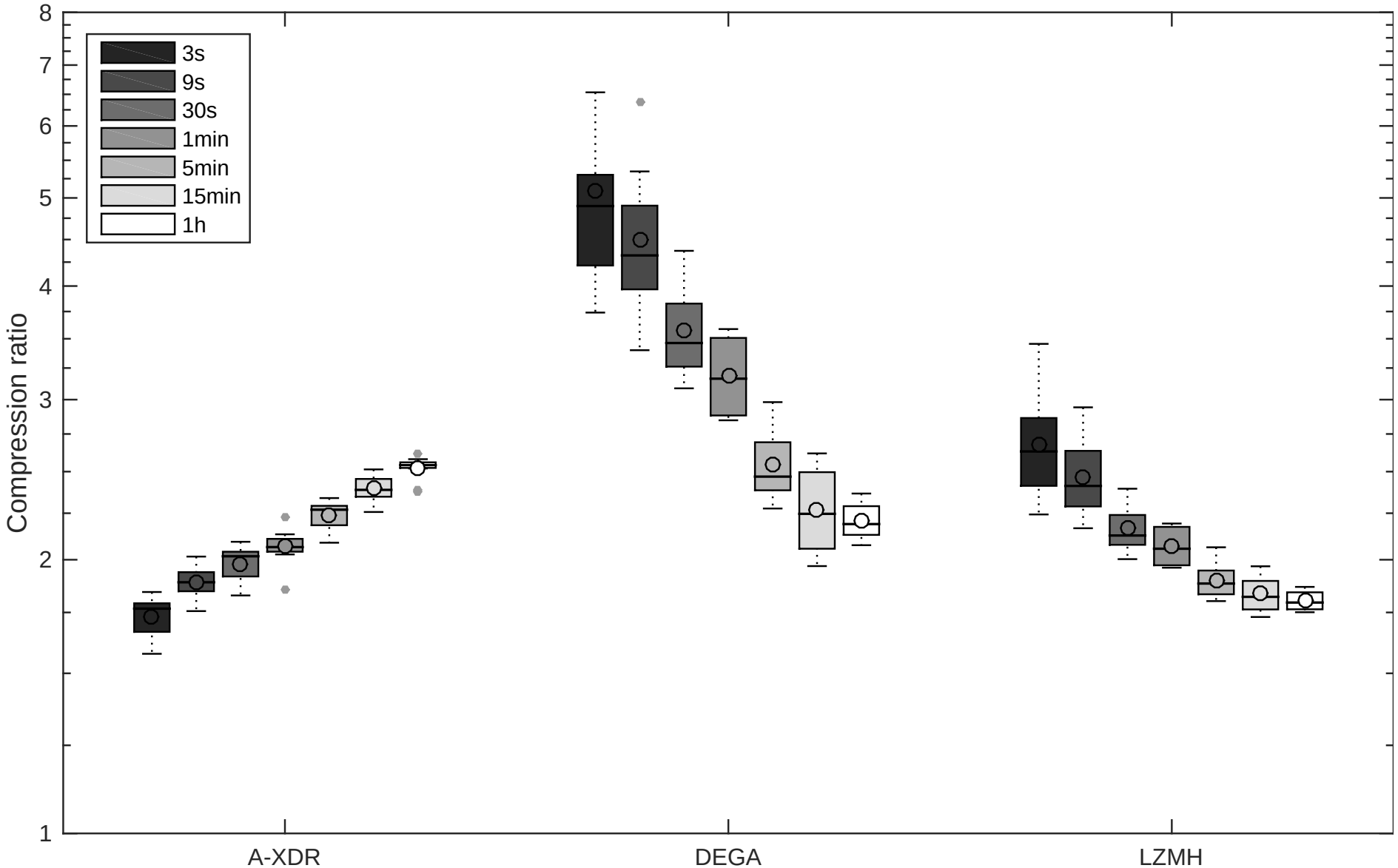
Evaluation

- Granularity levels
  - Simulation of coarser granularity levels through summation
  - MIT REDD data: 3 s, 9 s, 30 s, 1 min., 5 min., 15 min., 1 h
  - SAG data: 15 min., 1h
- Compression algorithms
  - IEC 62056-21 (input; converted if necessary)
  - A-XDR vs. DEGA vs. LZMH
- Implementation
  - Reimplementation of all algorithms for comparable I/O performance
  - C programming language; no special optimizations for any algorithm
  - Environment: 64-bit Ubuntu 14.04 on an Intel Xeon W3503 CPU

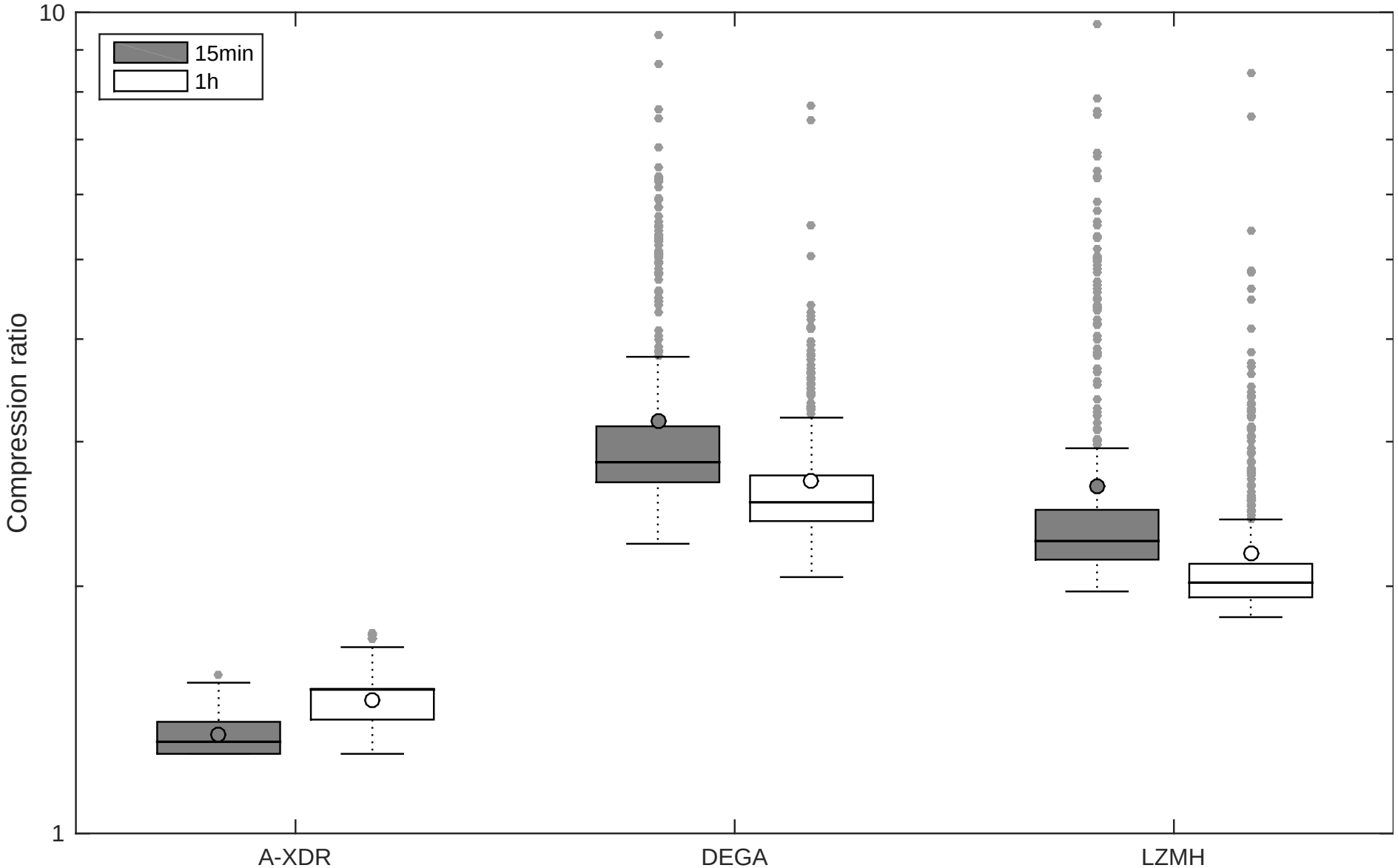
Results: MIT REDD data set



Results: MIT REDD data set (mains only)



Results: SAG data set

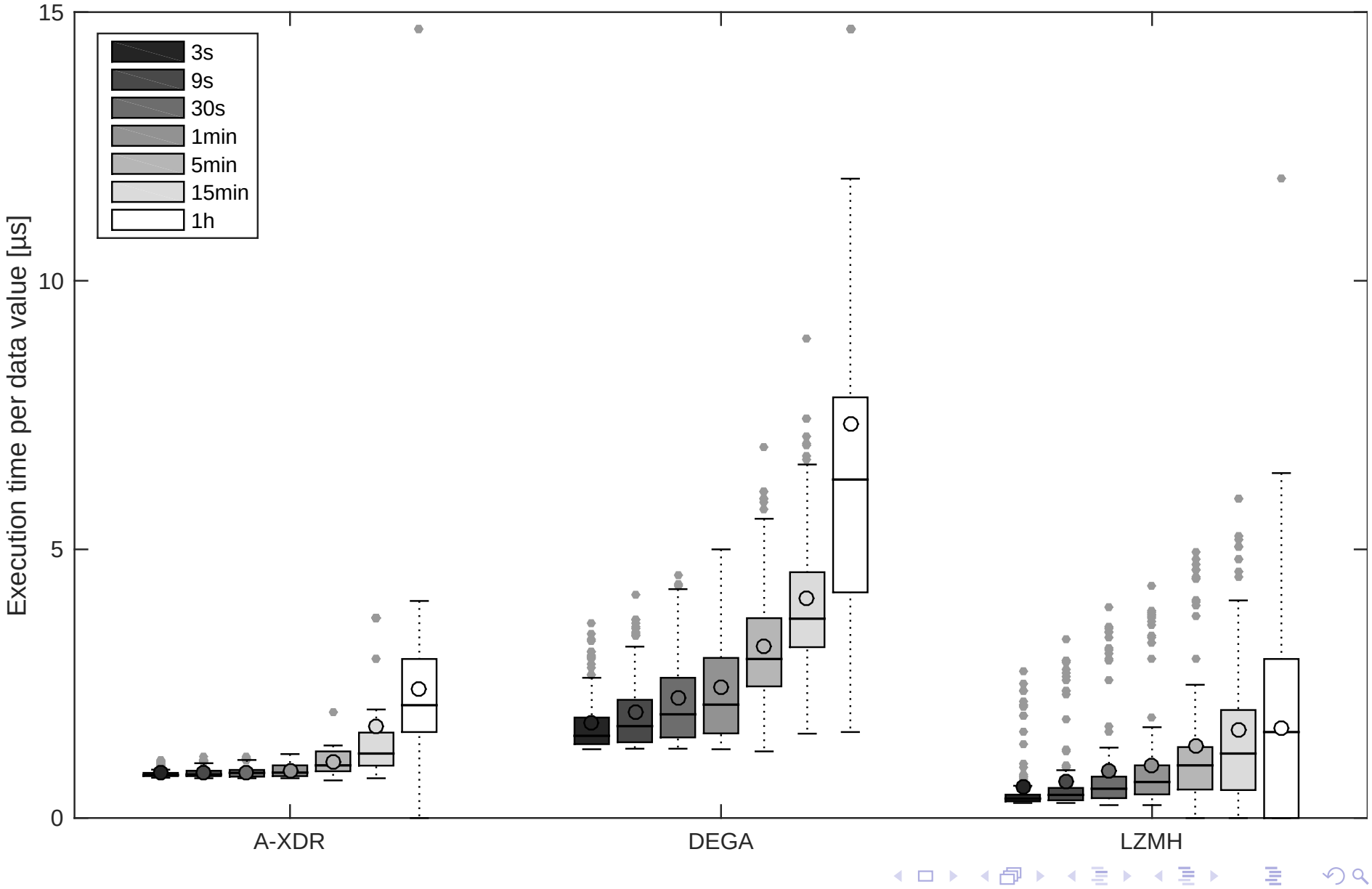


Summary

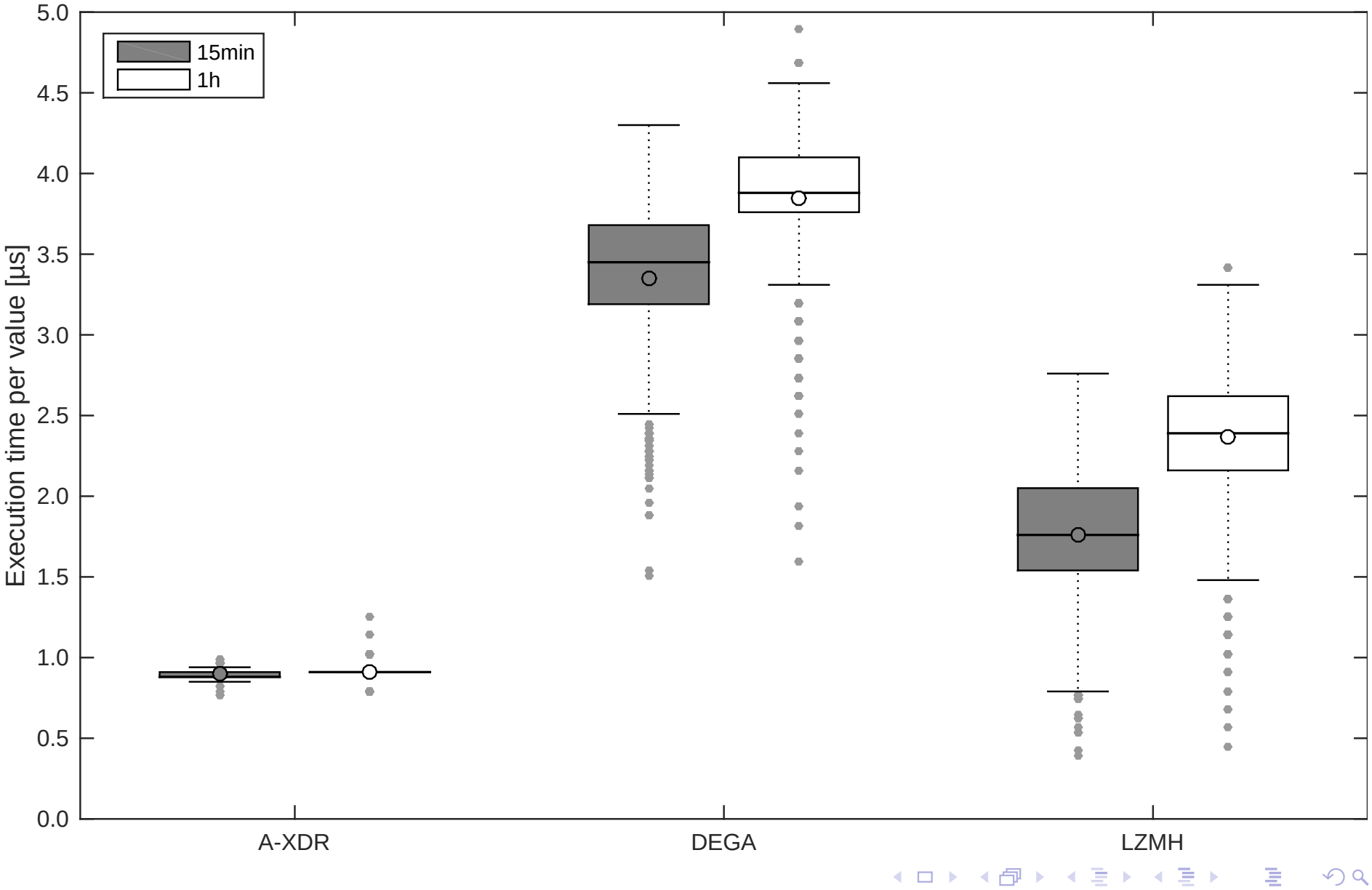
- The good news
  - Real-world load data is compressible
  - Channel-based compression allows for high compression ratios
  - Finer granularity levels allow for higher compression ratios
- The bad news
  - Real-world load data compression is barely worth it
  - Mains cannot be compressed to the same extent as separate channels
  - Practical (coarser) granularity levels render compression (nearly) futile

- But what about **time**?

Results: MIT REDD data set



Results: SAG data set



Conclusion

- Compressing load data is rarely worth the effort
  - Reasonable at fine granularity levels (e.g., 3 s)
  - Effective when individual channels are available
  - → Promises from the literature only true for impractical configurations
  - Compression does not cost a lot of time, but is not free either
- If you still want to compress
  - Use DEGA for mains and coarser granularity levels
  - Use LZMH for individual channels and finer granularity levels
  - Expect DEGA to be slower than LZMH
  - Consider the benefits of uncompressed coding

Thank you for your attention!

Questions?