PMACS'19 – Paper 3

Towards a Predictive Energy Model for HPC Runtime Systems Using Supervised Learning

Gence Ozer[1)], Sarthak Garg[1)], Neda Davoudi[1)], Gabrielle Poerwawinata[1)], Matthias Maiterth[3)], Alessio Netti[1,2)] and Daniele Tafani[2)]

1)

Leibniz Supercomputing Centre
of the Bavarian Academy of Sciences and Humanities 2)

3)

LRZ – Research for energy efficiency in production

- Hardware capabilities
- System integration
- Software solutions

- Examples at LRZ:
    - SuperMUC Phase 1: Hot water cooling & energy aware scheduling
    - CoolMUC 2: Adsorption chiller
    - LRZ projects for data center management regarding energy and cooling
    - SuperMUC-NG: Reflected in procurement document and now in production

- Big takeaway from all efforts:

“If you can’t measure it, you can’t improve it” *Peter Drucker*

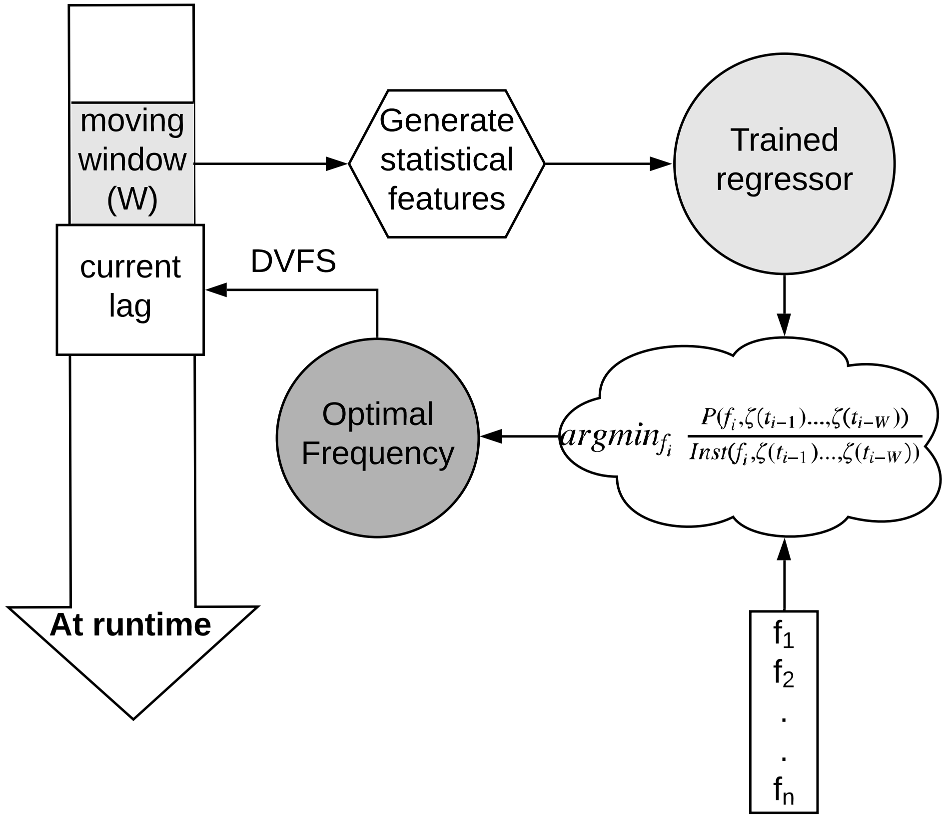
Monitoring energy in large scale systems

- Existing monitoring solutions:
  - Scheduler → Improve scheduling
  - Cluster monitoring → System health
  - User-job monitoring → Improved user support

- New use-cases:
  - Cluster monitoring for
    - System characterization
    - Understanding system utilization due to job characteristics.
  - Job runtime systems:
    - Utilizing application information
    - Predictive online performance tuning

Research question:

"Can we predict optimal frequency in the next time-step using available data."



Data collection:

- DCDB
  - Continuous system monitoring
  - System metrics e.g.:
    - BMC controller measurement
    - Kernel information (sysfs)
    - ‘perf’readings
    - ...
  - Centralized collection, for long term usability

- Main paper:
  - Netti et al. SC’19 (check it out at SC!)

- GEOPM
  Global Extensible Open Power Manager
  - Job runtime system to optimize power and performance of applications.
  - Job specific tuning
  - Tracing and reporting capabilities
  - Extensible agent-based:
    - Optimization strategies
    - Hardware interfaces

- Main paper:
  - Eastep et al. ISC’17

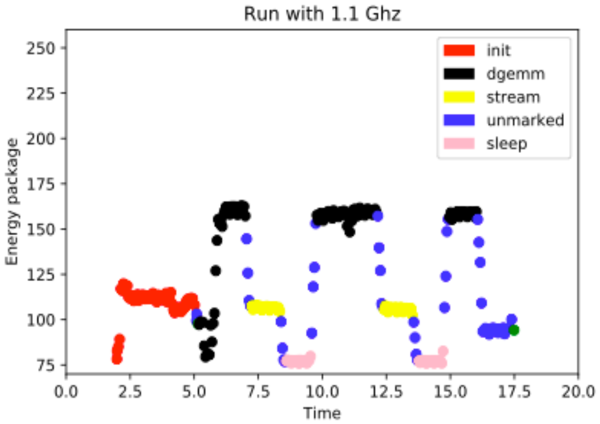
Data Generation – Usability and Improvements

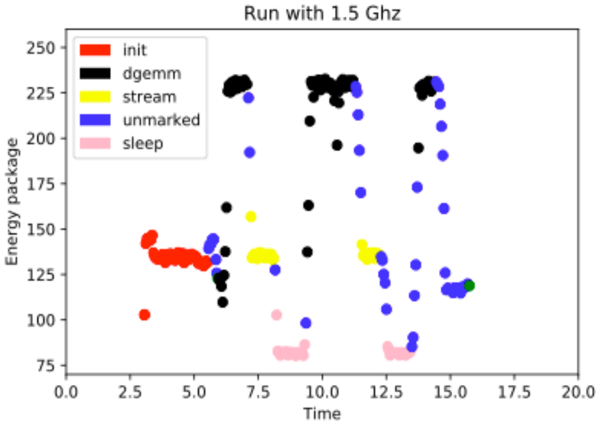
- DCDB:
  - ~400 counters 64bit each 0.1 second: 2.8GB per node per day.
  - CoolMUC3: 148 Nodes, CoolMUC2: 384, SuperMUC: 6480 Nodes
  - →CoolMUC2 would generate 1TB of monitoring data each day
  - + Data collection from sysfs every 2seconds.
  - What differentiates useful data from useless?

- GEOPM:
  - Fewer counters, selected by domain experts. No long term storage.
  - Criteria: High Granularity useful during Runtime: typical 4-8 cntrs dep. on use-case.
    - Granularity 0.01s, high fidelity due to GEOPMs use-case
    - Can decisions be augmented by data sources outside of runtime's scope?

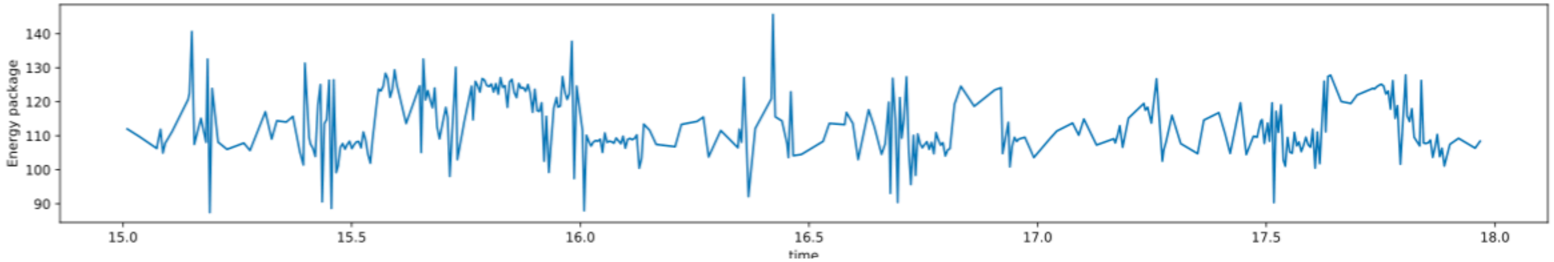
Preliminary Work: Taking a first look at the data

- Energy & Frequency for Benchmarks

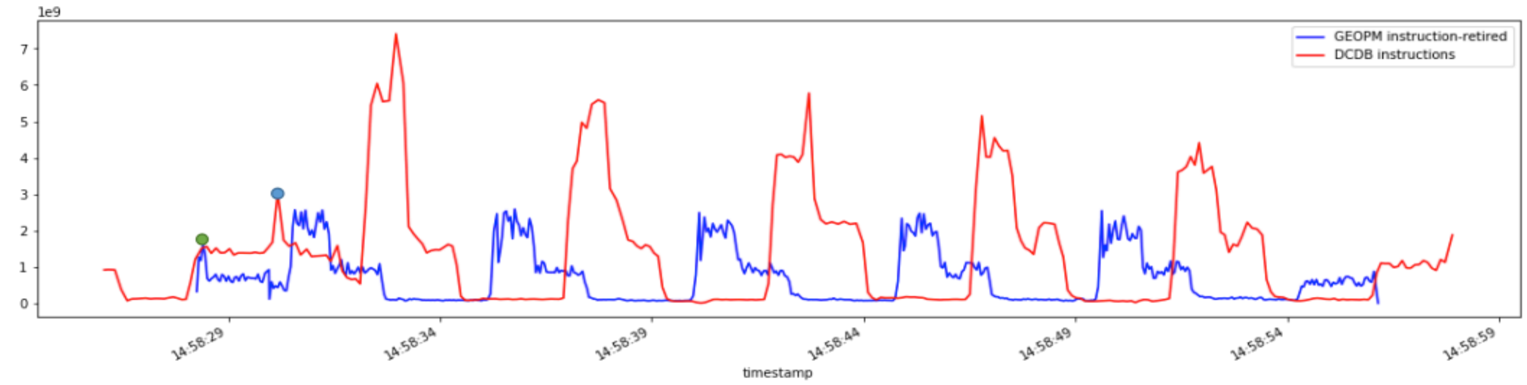




- Energy from LRZ user applications (Gadget)

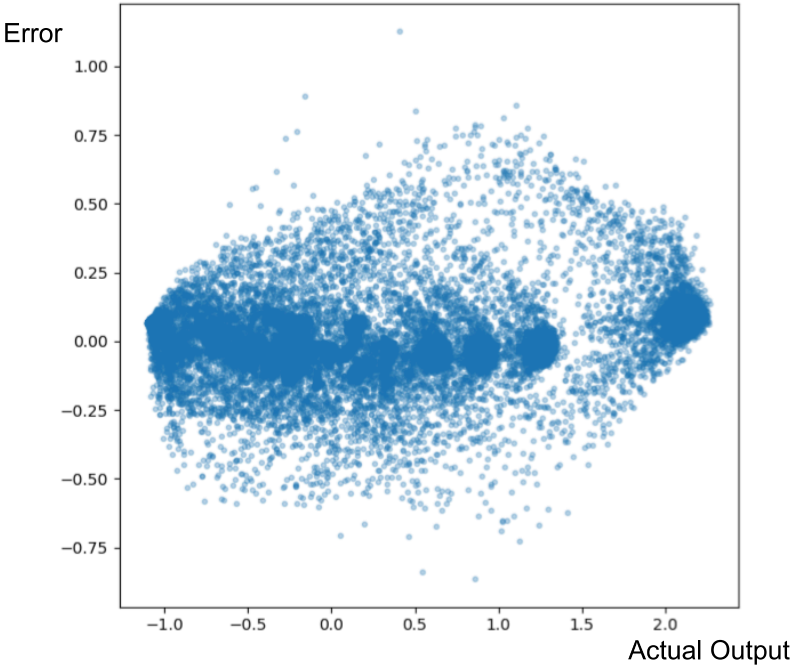


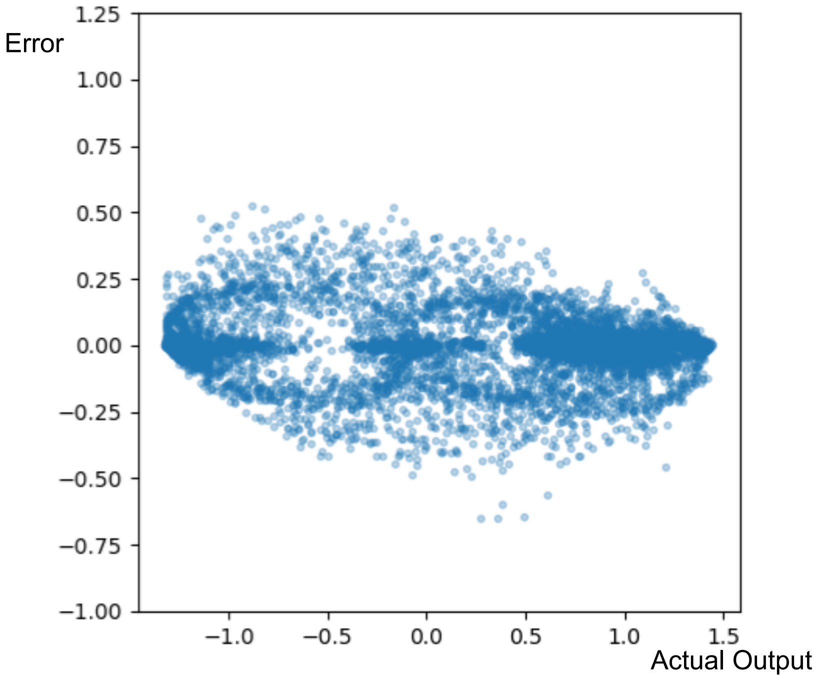
- Alignment of data sources

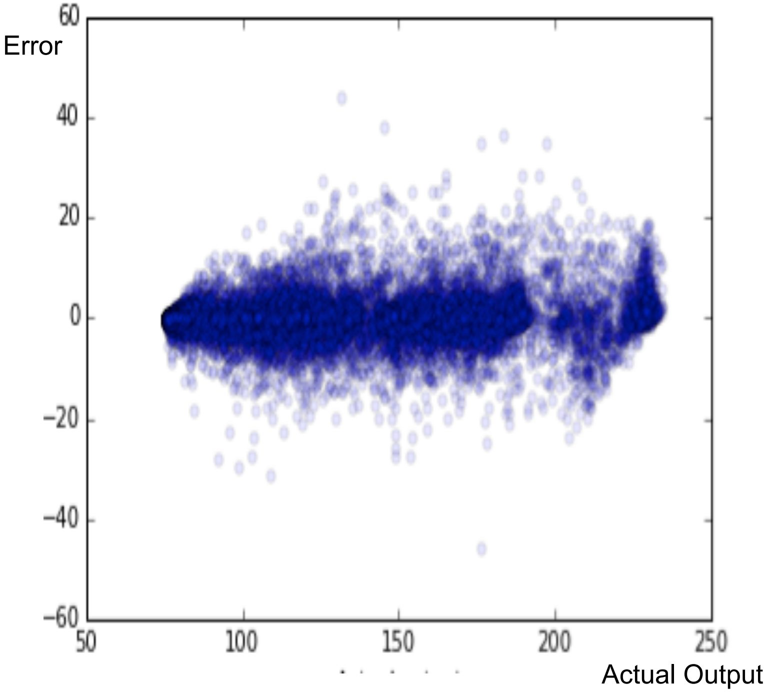


Preliminary Work: Selection of ML technique

- Ridge linear regression
- Support vector regression
- Random forest regression



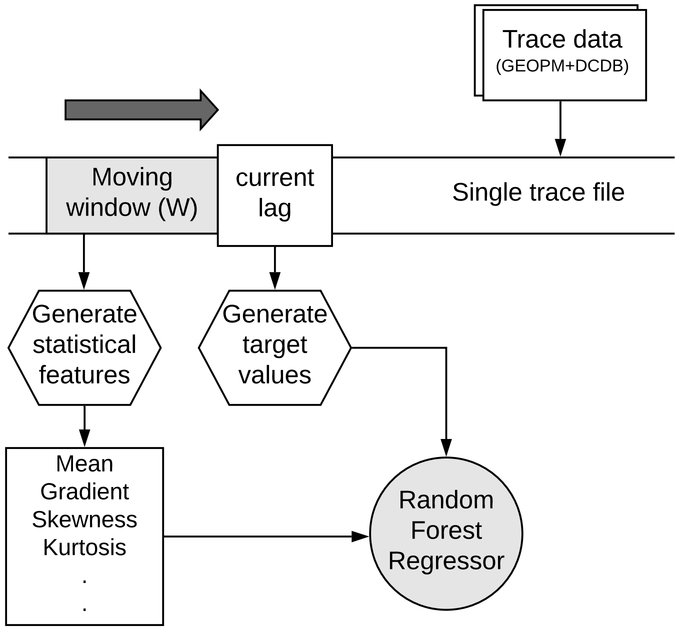




Mean relative error:

0.181 | 0.074 | 0.024

Forecasting optimal Frequency for the next time-slice



- Initial offline approach:
  - Selection of ML technique (done)
  - Training of regressor
  - Weighting of data by importance

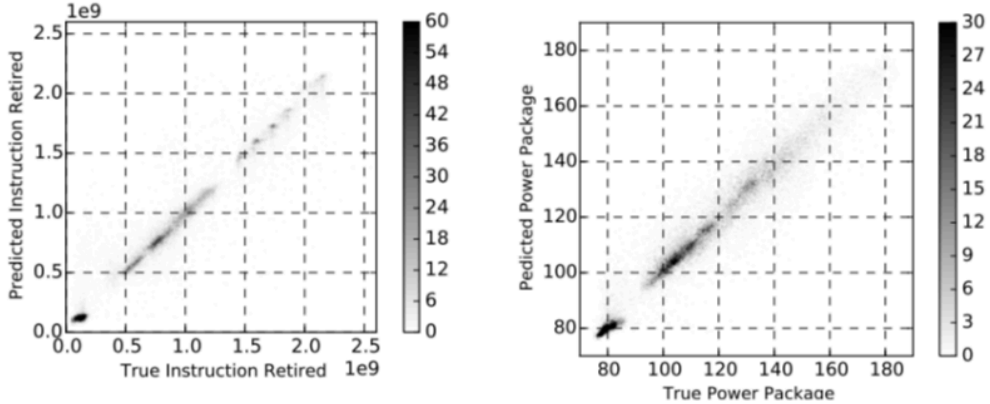
- Goal: $$MinEnergy(i) = min_{f_i} \frac{P(f_i, \zeta(t_{i-1}), \zeta(t_{i-2}), ..., \zeta(t_{i-W}))}{Inst(f_i, \zeta(t_{i-1}), \zeta(t_{i-2}), ..., \zeta(t_{i-W}))}$$

- Energy manipulated by changing control Mechanism of DVFS → Frequency

$$f_i^{opt} = argmin_{f_i} \frac{P(f_i, \zeta(t_{i-1}), \zeta(t_{i-2}), ..., \zeta(t_{i-W}))}{Inst(f_i, \zeta(t_{i-1}), \zeta(t_{i-2}), ..., \zeta(t_{i-W}))}$$

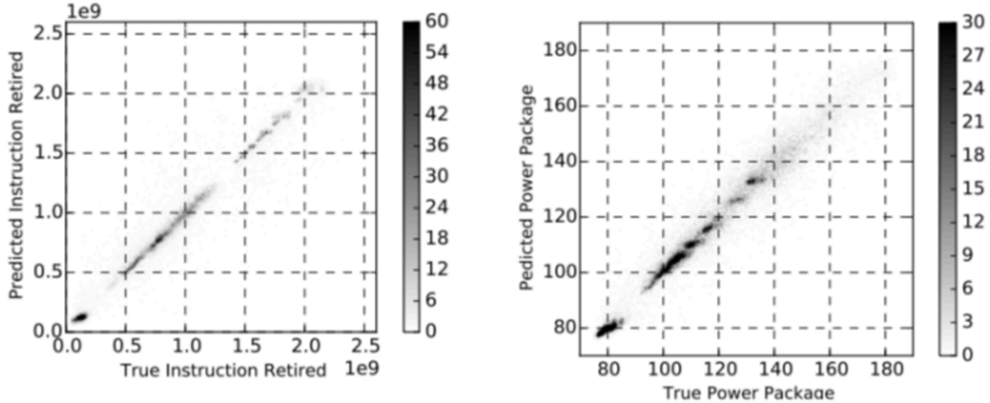
Evaluating model performance

- Training and comparison of using
  - Only GEOPM
  - GEOPM + DCDB



(a) GEOPM data.

	GEOPM	GEOPM + DCDB
Number of features	81	417
Overall training error	0.039	0.024
Overall validation error	0.091	0.060
Validation error (Power Package)	0.030	0.022
Validation error (Instruction Retired)	0.153	0.097

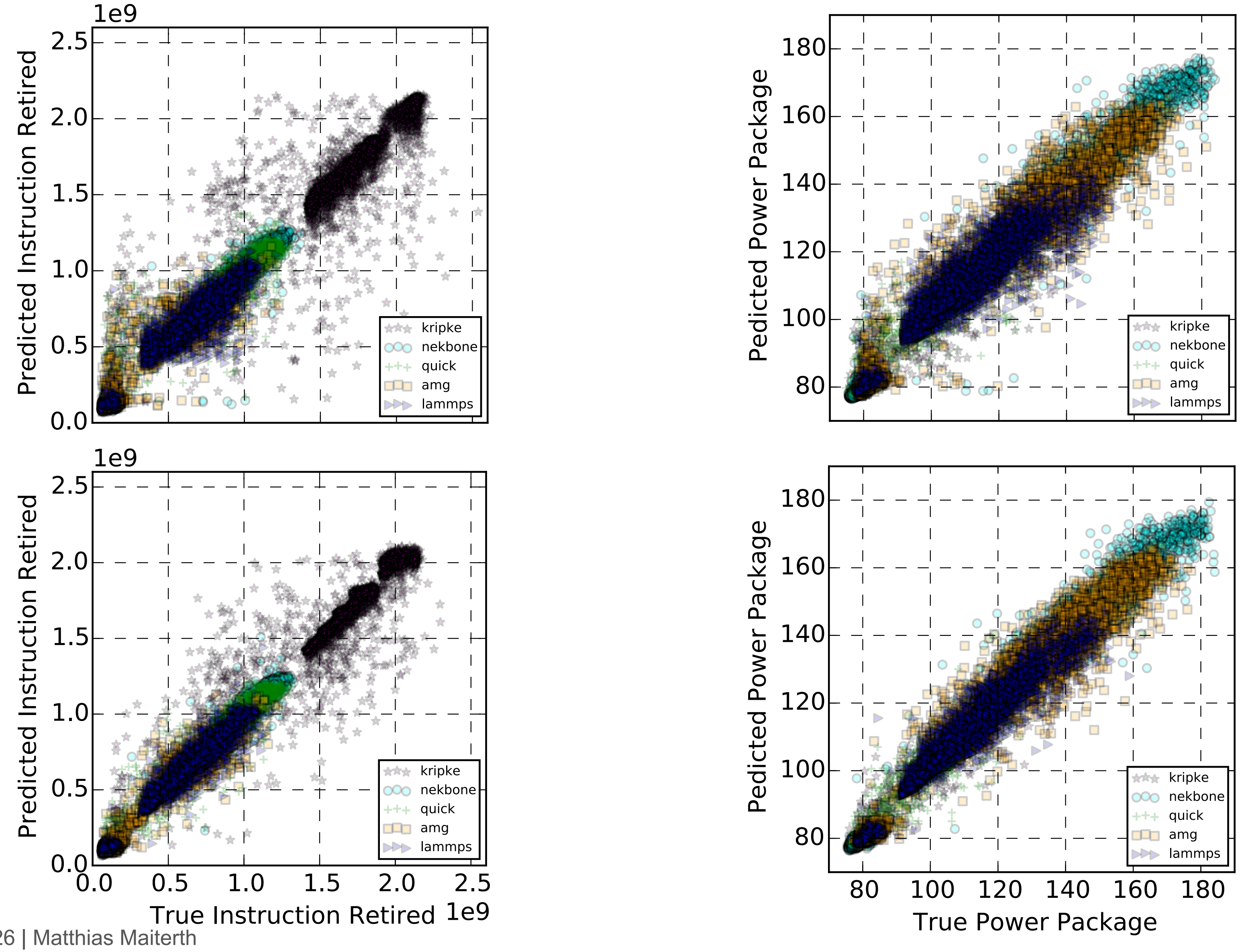


(b) GEOPM+DCDB data.

Prediction of instruction retired and power package with different data sources

Evaluating Model performance: By application

- GEOPM only
- GEOPM + DCDB



Derived feature importance of Random Forest Regressor

Evaluating Model performance

GEOPM		GEOPM+DCDB	
Score	**Name**	**Score**	**Name**
0.208	geopm inst-retired mean exp weighted	0.376	geopm inst-retired mean exp weighted
0.171	geopm cycles thread kurtosis	0.144	dcdb hfi0temp grad exp weighted
0.071	geopm cycles reference quantile 0.25	0.121	dcdb col idle grad exp weighted
0.060	geopm frequency	0.098	dcdb hfi0temp diff sum
0.048	geopm energy dram quantile 0.25	0.055	dcdb references quantile 0.5
0.047	geopm energy pkg quantile 0.75	0.052	dcdb energy quantile 0.75
0.045	geopm power pkg quantile 0.75	0.042	dcdb hfi1temp grad exp weighted
0.044	geopm power pkg quantile 0.5	0.040	dcdb intr quantile 0.25
0.040	geopm power pkg kurtosis	0.022	dcdb col idle diff sum
0.038	geopm inst-retired quantile 0.5	0.014	geopm frequency

Conclusions:

- Development of Machine Learning Model
  - Prediction of CPU power and Instructions Retired
  - With the goal of optimal frequency selection.

- Evaluation of combining data-sources:
  - DCDB & GEOPM

- Model shows universality and good accuracy:

  → Next Step: Integration of ML model in runtime

Gence Ozer
Sarthak Garg
Neda Davoudi
Gabrielle Poerwawinata
Alessio Netti
Daniele Tafani
Matthias Maiterth
matthias.maiterth@intel.com