ITU KALEIDOSCOPE
ONLINE2020

7-11 December 2020

Automation of Computational Resources Control of Cyber-Physical Systems with Machine Learning

Ved P. Kafle
National Institute of Information and Communications Technology
Tokyo, Japan

Session: 2

Design principles, architecture and protocols for the digital transformation

Paper: S2.2

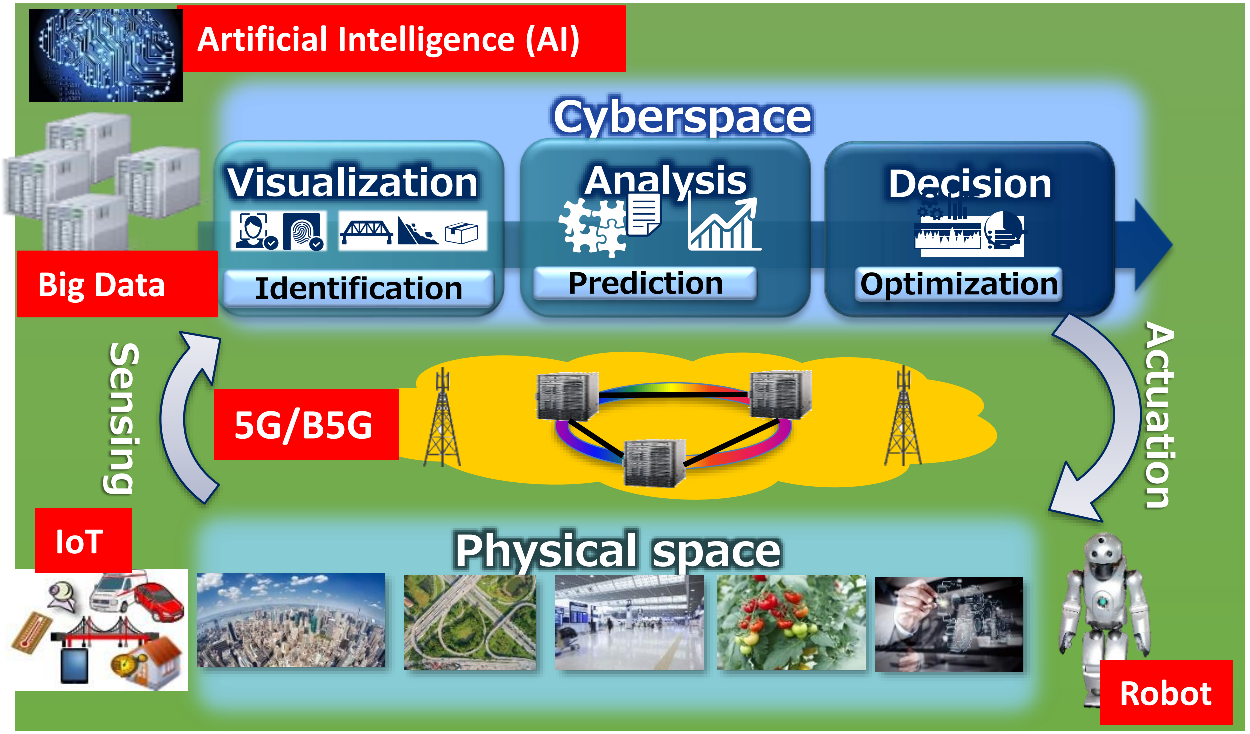
Profile: https://www.itu.int/en/ITU-T/academia/kaleidoscope/2020/Pages/Ved-P-Kafle.aspx

Content

- Cyber-physical system (CPS) overview
- System model and problem domain
- Related work
- Offline training of machine learning models
- Deployment and online retraining
- Experimental setup and results
- Conclusion and standardization prospective

Cyber-physical system overview

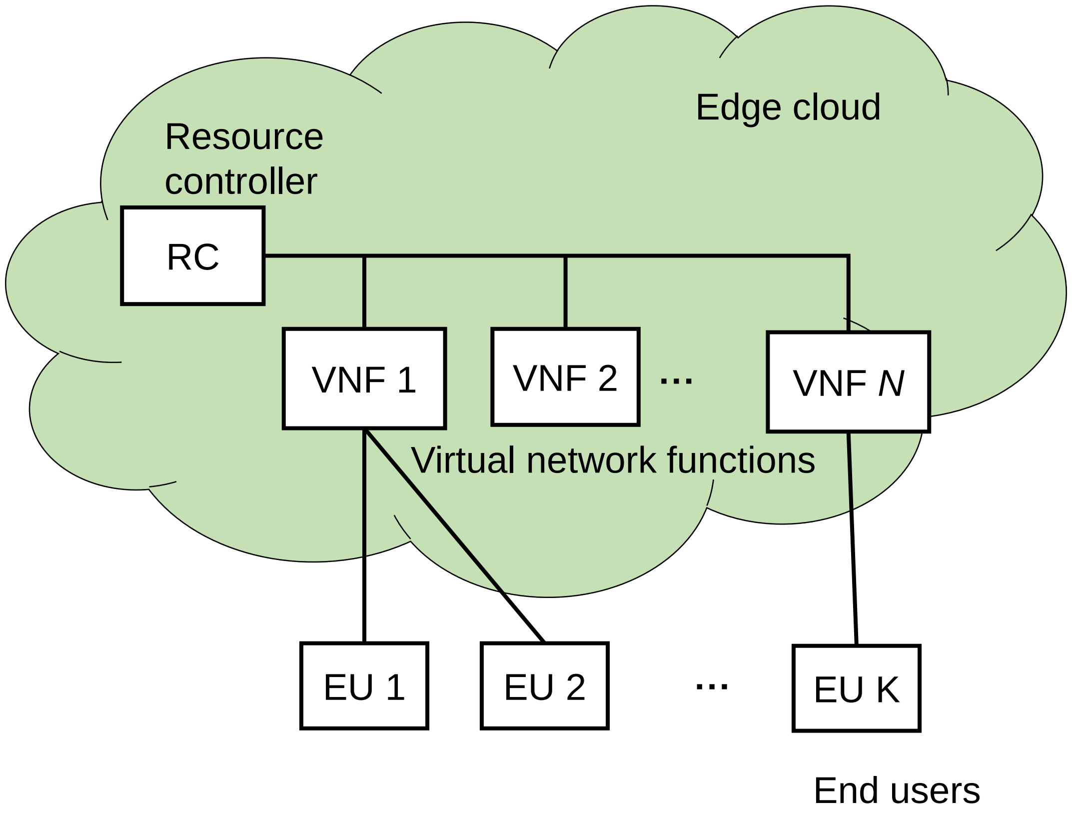
- Enabler of Society 5.0, Industry 4.0
- Component technologies
  - AI, IoT, Big data
  - 5G/B5G networks
  - Robotics
- Ultra low latency applications growing
  - Autonomous driving
  - Factory automation
  - Remote surgery
- Require computing facility (i.e., cyber system) closer to users
  - Edge cloud



System model and problem domain

- Edge cloud
  - Computational and storage resources deployed closer to end users (EU)
  - Cyber applications deployed in the form of virtual network functions (VNFs)
  - Possesses limited resources, but is need to satisfy low latency computation requirements
- Require resource monitoring and dynamic control (adjustment) mechanism

Target of this work

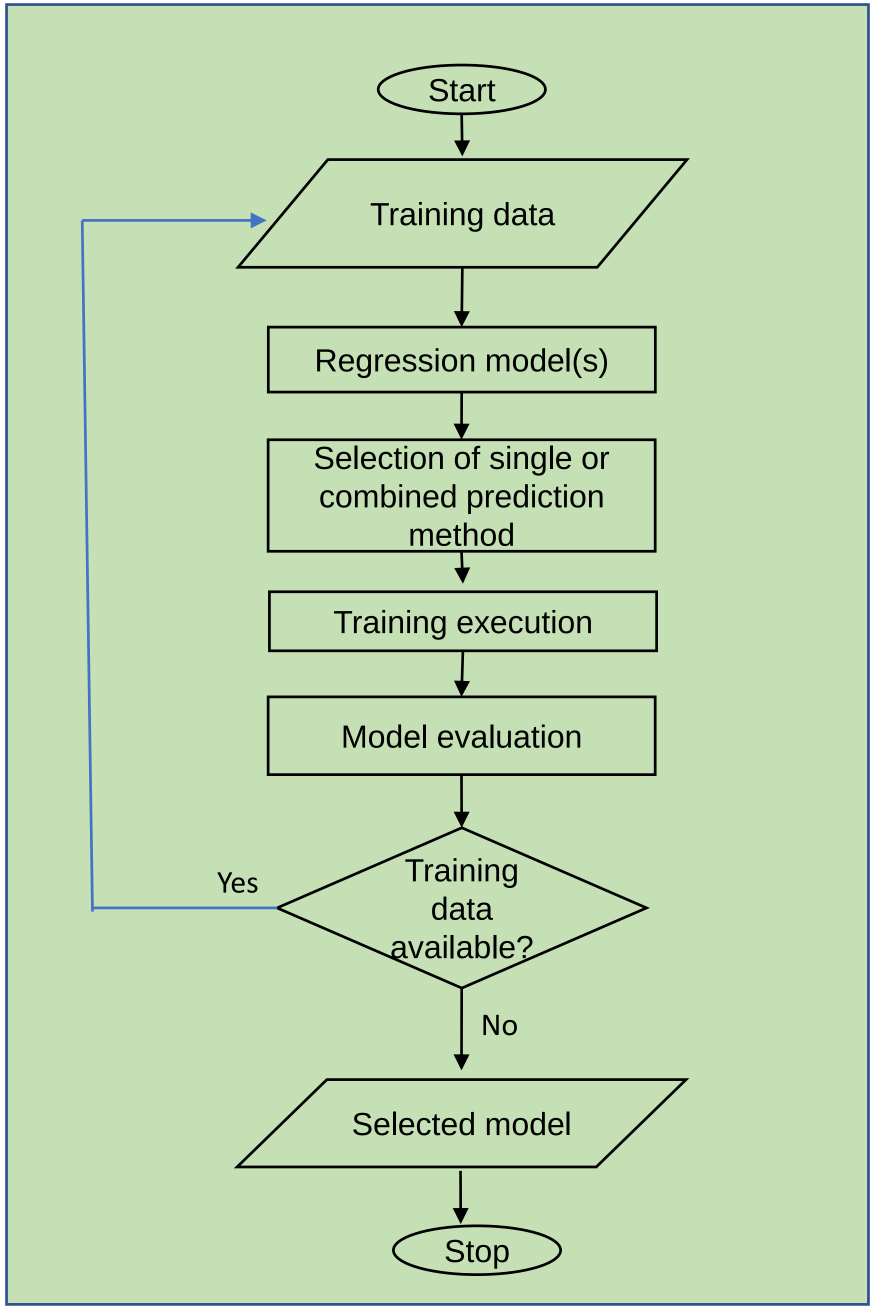


Related work

- Mechanisms for resource adjustment according to workload prediction
  - Threshold rule-based (reactive)
  - Machine learning-based (predictive)
- Commonly used machine learning (ML) models
  - Gaussian process [6]
  - Auto-regression [7]
  - Supervised learning [8] (require human involvement in training data preparation)
  - Reinforcement learning [11] (no human involvement in training, better prediction accuracy in unseen data inputs, but slow in convergence)
- This work
  - Multiple regression models
    - Extremely-randomized trees regression (ETR)
    - Gradient boosting regression (GBR)
  - Achieve better prediction accuracy, higher resource utilization and agile control

Offline training of regression models

- Training data preparation
  - Data collection by operating the target system with simulated workload
    - Data = {workload, resources status, latency, …}
    - Collected at the highest possible frequency without hampering performance (e.g., 1s intervals)
- Offline training
  - Train regression models by training data and tuning hyperparameters
  - Rank models based on their prediction accuracy and training time consumption
  - Select the most accurate model to use in system



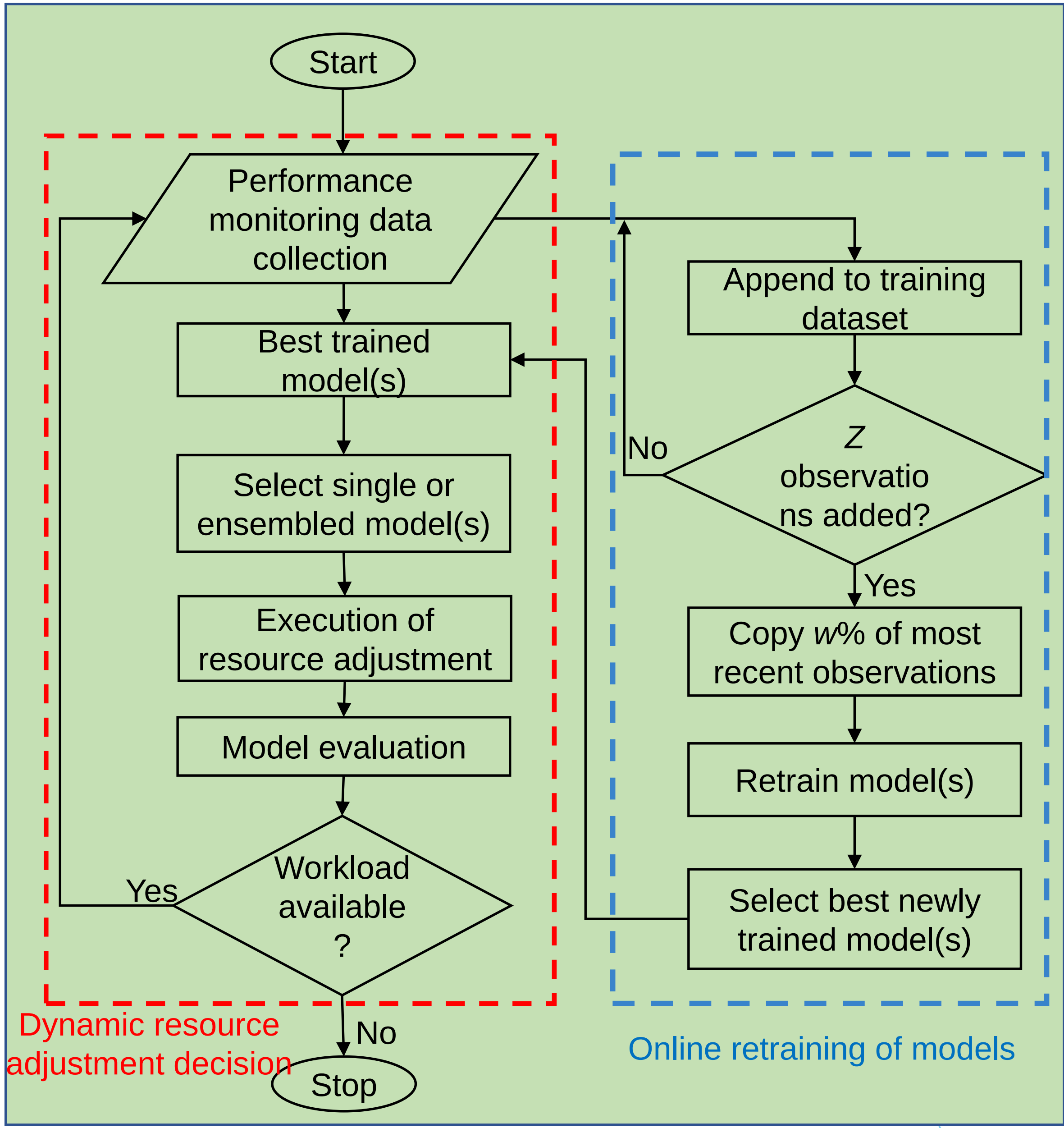
Offline training of regression models

Model deployment and online retraining

- Model deployment
  - Deployed the most accurate model
  - Predicted workload and system status
  - Resource adjustment decision, $y = f(x_i)$,

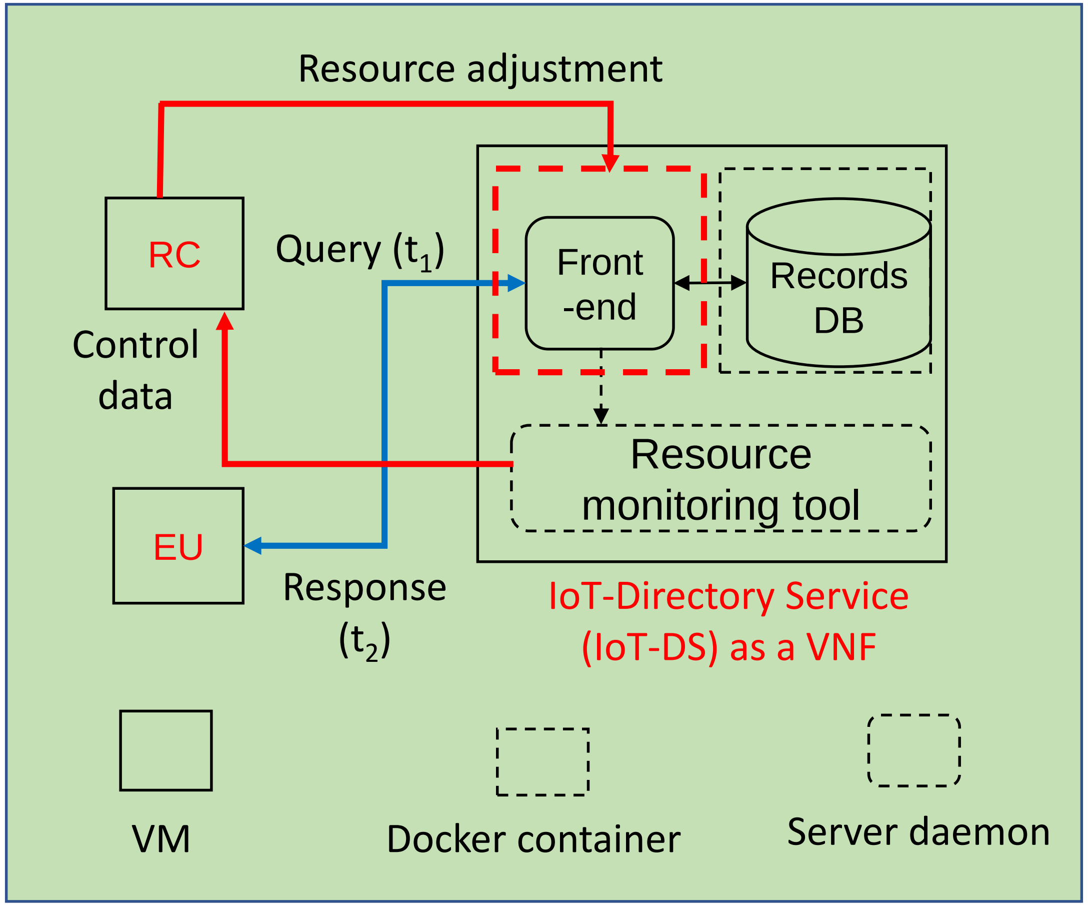
    where y = new amount of required resource; xi = current system parameters (workload, resource utilization, performance latency, …)
  - Evaluated prediction errors by using performance feedbacks

- Online retraining for improving accuracy
  - Retraining models by data obtained from running system
  - Best among newly trained models selected and updated in system



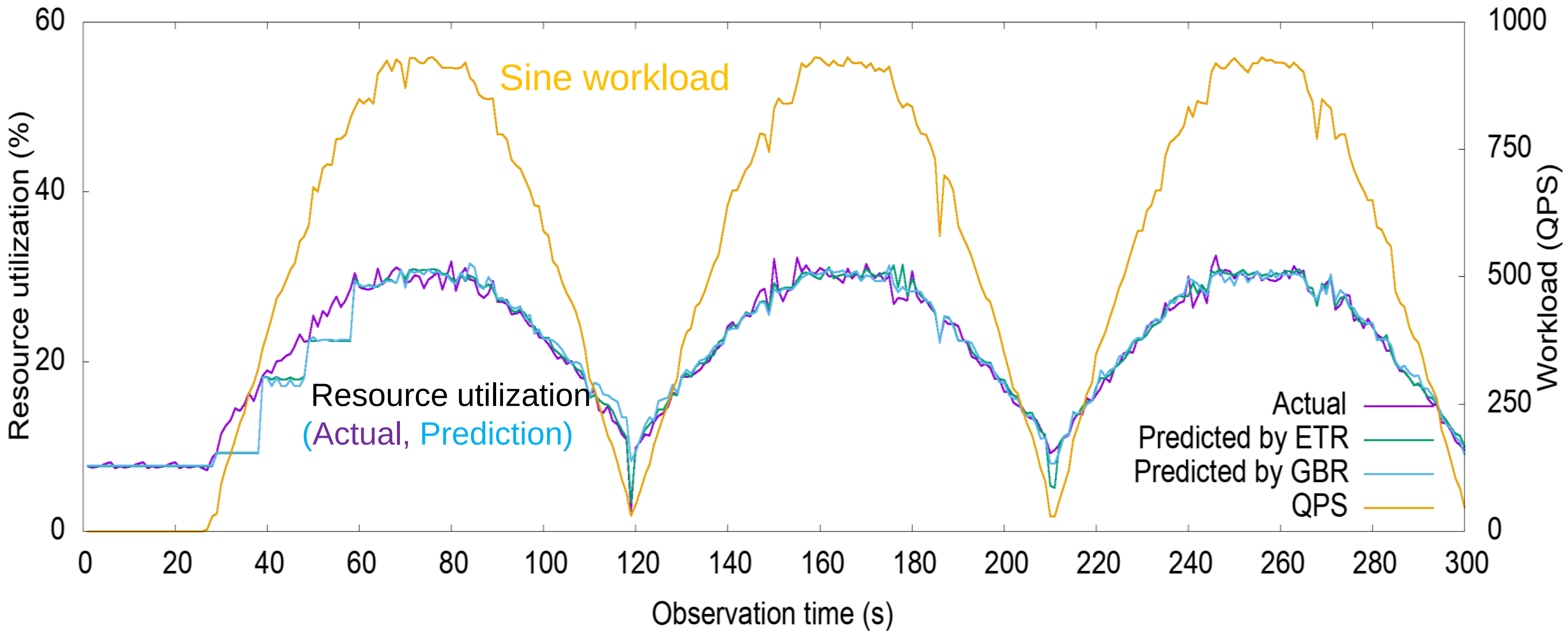
Performance evaluation: Experimental setup

- Implemented in virtual machines (VMs)
- Resource controller (RC)
  - ML models (written in Python), training and testing database; resource control commands generator
- End-user device (EU)
  - Workload generating in various patterns (Poisson, Sine curve, etc.) and sending to VNF
- IoT-directory service (IoT-DS) as a VNF
  - Implemented in Docker container
  - Comprising front-end and back-end (IoT records database with 100K records
  - Monitoring front-end for resource allocation, utilization, workload, etc.
  - Dynamic adjustment of allocated CPU cycles of front-end by Docker commands



IoT DS architecture is based on Rec. ITU-T Y.3074

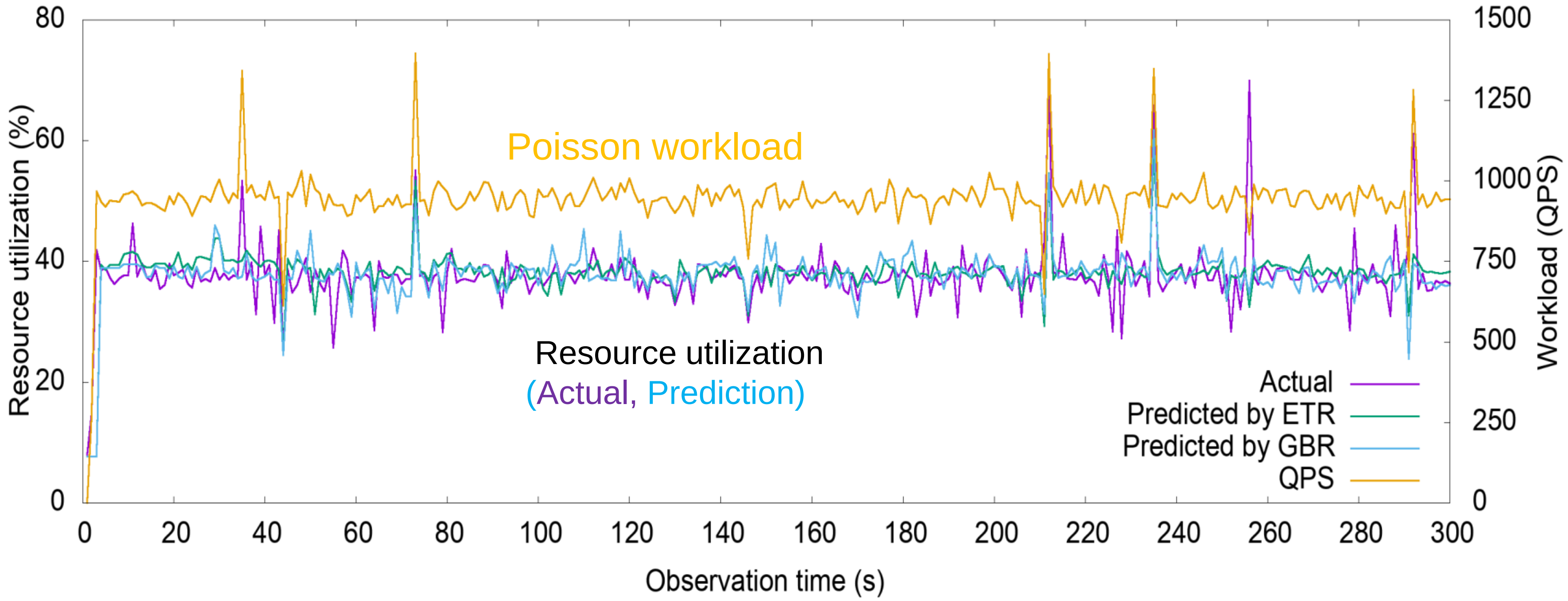
Performance evaluation: Results (1/4)



Comparison of actual and predicted resource utilization for Sine workload pattern

- Initial (re)training stage (20-60s): prediction < actual
- After retraining, accuracy increases, prediction slightly > actual

Performance evaluation: Results (2/4)



Comparison of actual and predicted resource utilization for Poisson workload

- As workload variation is less,
  prediction almost equal to or marginally higher than actual
  after the first round of training (at around 10 s)

Performance evaluation: Results (3/4)

- Measurement of errors:
  - Mean Absolute Error (MAE)

$$\text{MAE} = \frac{1}{X}\sum_{i=1}^{X} abs(a_i - p_i) \times 100\ \%$$

a_i and p_i = actual utilization and predicted utilization, respectively,
X = total number of observations.

  - Delta

$$\Delta = \frac{\sum_{i=1}^{X}(a_i - p_i)}{\sum_{i=1}^{X} a_i} \times 100\%$$

+ve value = underfitting;
-ve value = overfitting

	Sine workload		Poisson workload	
	MAE (%)	Δ(%)	MAE (%)	Δ(%)
ETR	0.89%	0.96%	2.66%	- 1.16%
GBR	1.02%	0.59%	2.85%	- 0.47%

- Sine wave workload is easy to predict, thus smaller MAE
- Poisson workload has a slightly larger MAE with prediction > actual (thus –ve Δ)

Performance evaluation: Results (4/4)

Comparison of resource saving and performance satisfaction

	Average of 5 observation	
Algorithms	CPU Allocation	Latency violation (cases of >8ms)
Conventional [10] as baseline	1	11
GBR	0.781	5.6
ETR	0.81	7.6

- Compared to Conventional threshold-rule based algorithm, this work with GBR and ETR reduced
  - CPU resource demand by 21.9% and 19%.
  - Latency requirement violations by 49.0% and 30.9%, respectively.

Conclusion

- Presented a machine learning based mechanism for the prediction of system workload and resource utilization and dynamically adjusting resources

- Experimental results demonstrated its effectiveness to meet QoS requirements with lesser amount of resources

- Future work:
    - Develop algorithm for the automatic selection of training data size and intervals
    - Extend the mechanism to simultaneously adjust CPU, memory and bandwidth
    - Contribution to standardization

Standardization perspective

- Related ITU-T Recommendations (already published):
  - ITU-T Y.3074 (Directory service architecture for storing huge amount of IoT records)
  - ITU-T Y.3172 (Architectural framework for machine learning in networks)
  - ITU-T Y.3174 (Framework for data handling to enable machine learning in future networks)

- Related ITU-T Recommendation drafts (work-in-progress in Study Group 13):
  (Authors contributing from the outcome of this research work)

  - Y.ML-IMT2020-RAFR (network resource and failure management)
  - Y.ML-IMT2020-serv-prov (network service provisioning)

ITU KALEIDOSCOPE

ONLINE 2020

Thank you!