University of Glasgow
School of Computing Science
ESSENCE
PERVASIVE & DISTRIBUTED INTELLIGENCE

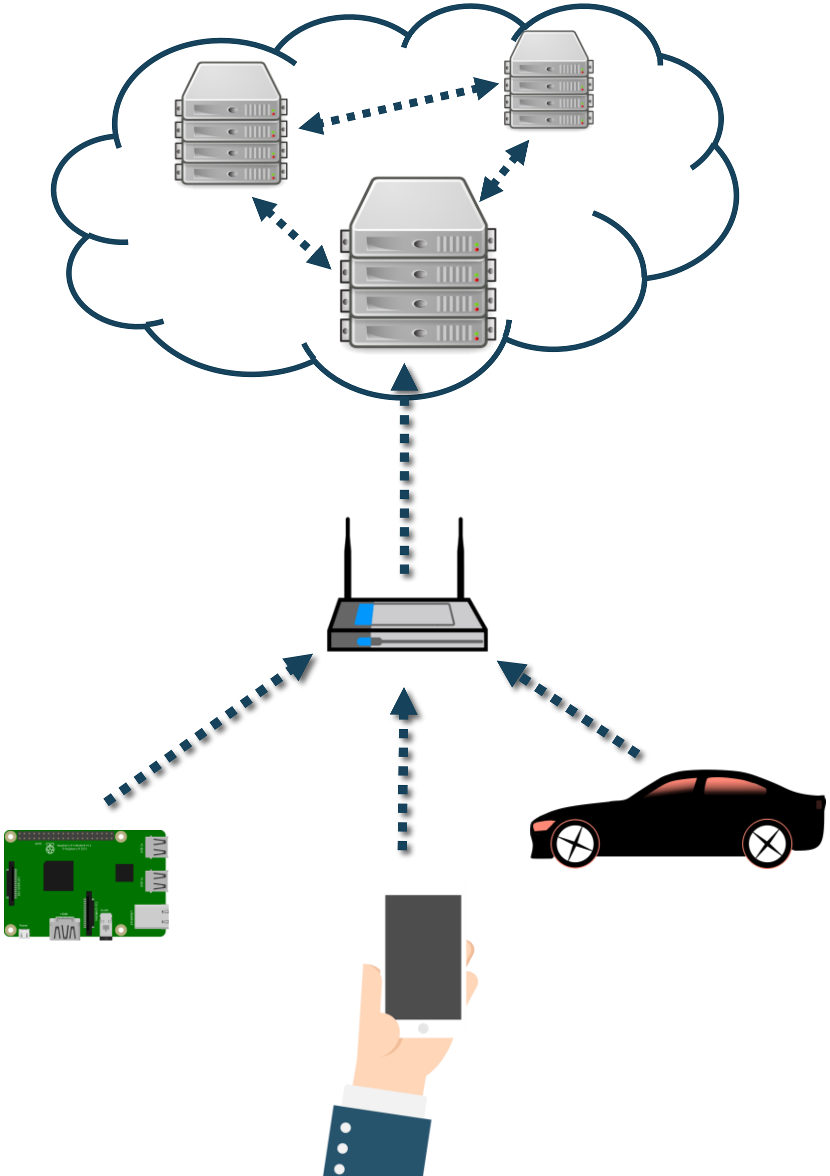
Machine Learning Model Updates in Edge Computing: An Optimal Stopping Theory Approach

Katie Aleksandrova,
Christos Anagnostopoulos, Kostas Kolomvatsos

18th IEEE International Symposium On Parallel And Distributed Computing,
Amsterdam, 5-7 JUNE 2019

Problem Statement

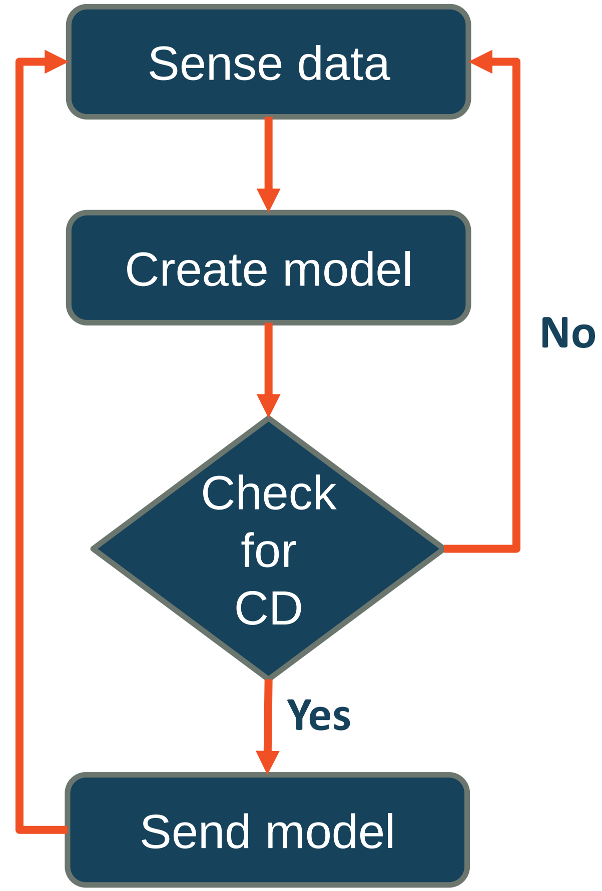
- **Internet of Things system:**
  - Edge Sensors
  - Neighbourhood Edge Gateways
  - Data Centres (Cloud)
- **What we Are doing:**
  - Sense multivariate contextual data at the Edge
  - Transfer data to the Cloud for analytics
  - Have accurate and up-to-date knowledge in the Cloud
- **What we Don’t want:**
  - Computational overhead at the Data Centres
  - Communication Overhead
  - High Network bandwidth



Problem Statement

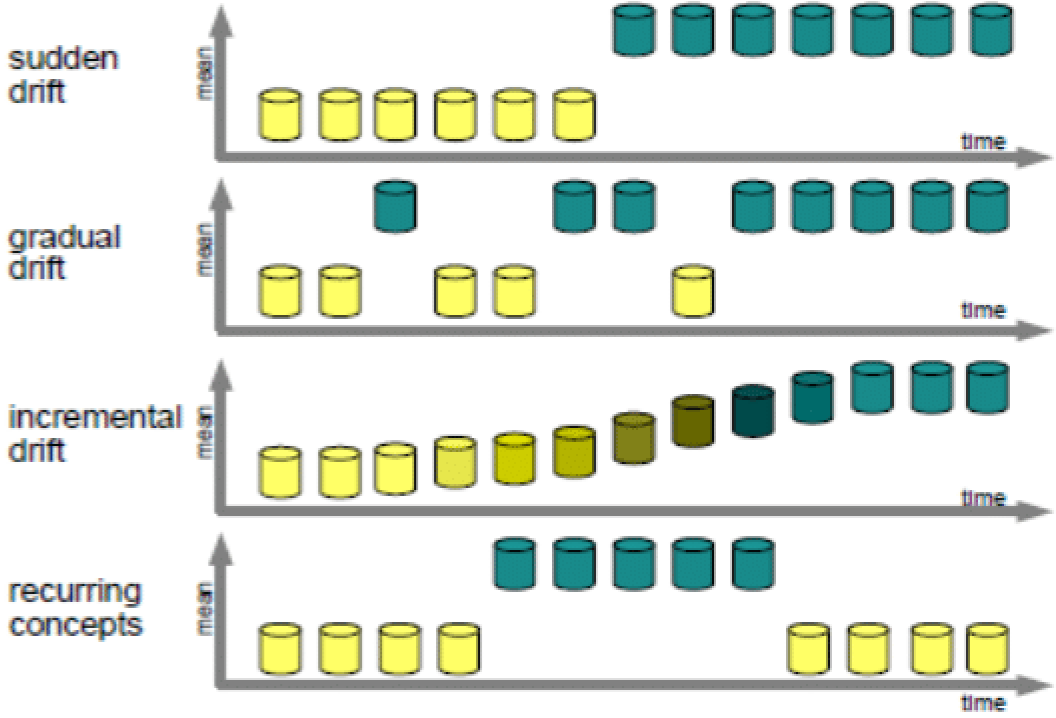
- **What we Can do:**
  - **Gather** some of the sensed data in the sensor
  - **Create** a ML model from that data
  - **Communicate** the ML model
  - **Wait** until a ML model Concept Drift (CD) has occurred
  - **Communicate** an updated ML model

- **What we Will achieve:**
  - **Less communication** in the network
  - **Lower bandwidth** requirement
  - Data is delivered to the Datacentre **partially analysed**
  - Data is **anonymised** by preserving the raw context at the sensor level



What is a Concept Drift?

- *Def.* **'A changing context which induces a change in the target concepts'** **(Widmer & Kubat, 1996)**



(Lemaire et. al., 2015)

Handling Concept Drift: Cumulative Sum (CuSum)

- **Absolute Error Difference** between current ML model and previously delivered ML model on the most up-to-date data:
  - $\Delta e = |e - e'|$

- **Good Distribution** and the **Bad distribution of Δe**

- Estimate the Probability Density Functions: P_{good} **and** P_{bad}

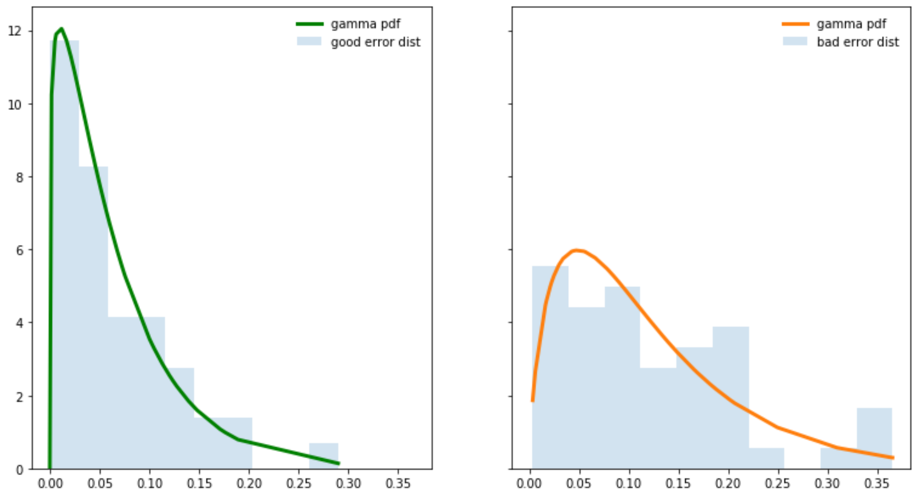


Fig. Good Distribution vs. Bad Distribution

CuSum: Invented by E. Page, Uni of Cambridge, 1954

Handling Concept Drift: Cumulative Sum (CuSum)

- For each new $\Delta \boldsymbol{e}$, calculate the Log-Likelihood Ratio:
  - $\boldsymbol{l_t} = \boldsymbol{L_{\Delta e}} = \boldsymbol{ln} \frac{\boldsymbol{P_{\Delta e \mid bad}}}{\boldsymbol{P_{\Delta e \mid good}}}$
- Sum up the log-likelihood ratios up to time t:
  - $\boldsymbol{S[t]} = \sum_{k=0}^{t} \boldsymbol{l_k}$
- Decision Value for Concept Drift detection:
  - $\boldsymbol{g} = \boldsymbol{S[t]} - \boldsymbol{min_{0 \leq k \leq t-1}(S[k])}$
- **ML Model Update Criterion:** $\boldsymbol{g > h}$, $(\boldsymbol{h} \leftarrow \boldsymbol{threshold})$

From CuSum to Optimal Stopping Theory

- **What does Optimal Stopping Theory deal with?**
  - How to estimate the best time to **stop** a process and gain the **highest reward** or suffer the least penalty?

- **Popular Examples:**
  - The Secretary problem
  - The Blackjack Card game
  - The House Selling problem
  - …

- **Our problem: Delay** sending a ML model update as much as possible until a **change** in the distribution of the error difference has occurred.

Optimal Stopping Theory in Practice

- Cumulative Sum principle on the **Absolute Error Difference** not allowed to exceed a Prediction Quality Tolerance $\boldsymbol{\Theta}$

  - **Error Difference:** $\Delta \boldsymbol{e}_t = |\boldsymbol{e}_t - \boldsymbol{e}'_t|$ **with CDF** $\mathrm{F}_{\Delta e}$
  - **Cumulative Sum:** $\boldsymbol{S}_t = \sum_{k=0}^{t} \Delta \boldsymbol{e}_k$

- **Problem:** Maximize the Reward Function

$$\boldsymbol{V}_t = \begin{cases} \boldsymbol{t}, & \boldsymbol{S}_t \leq \boldsymbol{\Theta}; \quad postopone\ model\ delivery\ (continue) \\ -\boldsymbol{B}, & \boldsymbol{S}_t > \boldsymbol{\Theta}; \quad penaly\ (stop) \end{cases}$$

- **Theorem:** If the currently reward is higher than the conditional expected future reward, send an updated model. The reward is maximized at the first time t:

$$\boldsymbol{V}_t \geq \mathbb{E}[\boldsymbol{V}_{t+1} | \mathbb{F}_t] \Leftrightarrow \boldsymbol{F}_{\Delta e}(\boldsymbol{\Theta} - \boldsymbol{S}_t) \leq \frac{\boldsymbol{t} + \boldsymbol{B}}{\boldsymbol{t} + \boldsymbol{1} + \boldsymbol{B}}$$

Other Model Update Policies

- Median-based Policy
  - ML Model Update Criterion:

    $\Delta e_t > \alpha * median(\Delta e_1, \ldots, \Delta e_{t-1})$, α in (0,1)

- Accuracy-based Policy
  - ML Model Update Criterion: (old) $\mathrm{e_t} > e_t'$ (new)

- Random-based Policy
  - ML Model Update with probability: p
  - p is the *empirically* estimated probability sending at the best time

Performance Evaluation

- **GNFUV: Unmanned Surface Vehicles Sensor Dataset**

  (Harth & Anagnostopoulos, 2018)

  - **data**: (humidity, temperature) from 4 USVs
  - used with **Linear Regression**

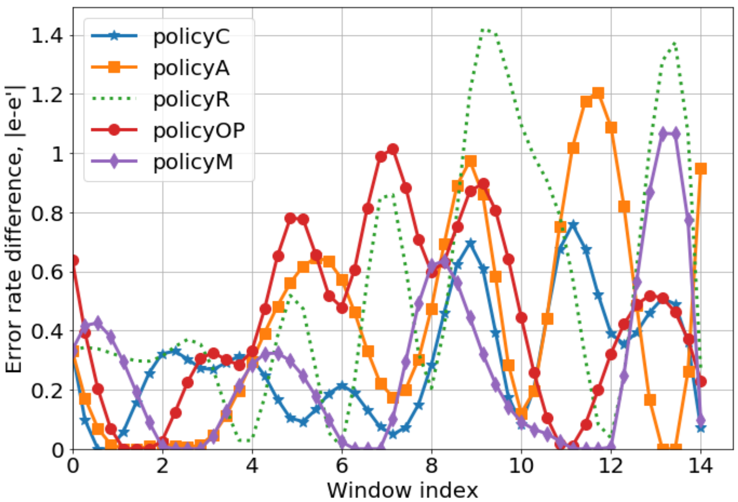
- **Gas Sensors for Home Activity Monitoring Dataset**

  (Huerta et. al., 2016)

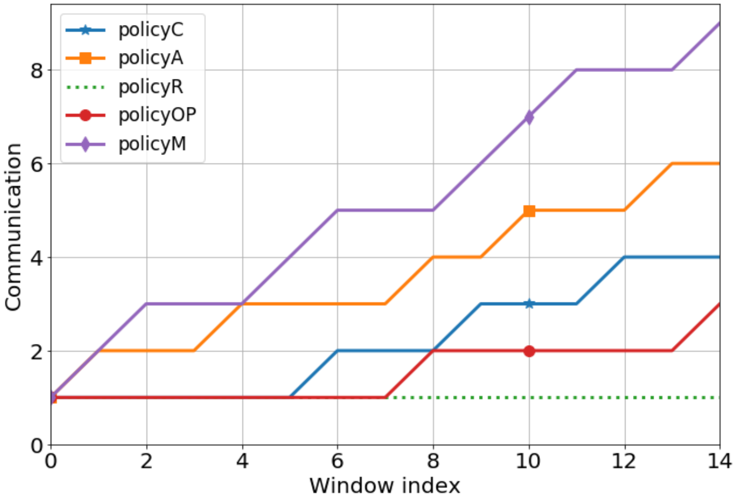
  - **data**: (humidity, temperature) from 8 metal-oxide sensors
  - used with **Support Vector Regression** (RBF kernel)
  - included artificial incremental concept drift in the data

Linear Regression Model

- The absolute error difference for the Optimal Policy (OP) does not drastically deviate from the other policies.



- OP saves on average **5 times** more communication



ANOVA Test: Linear Regression

- Statistical significant difference b/w policies:
  - ‘waiting time’ (ML model update postponing)

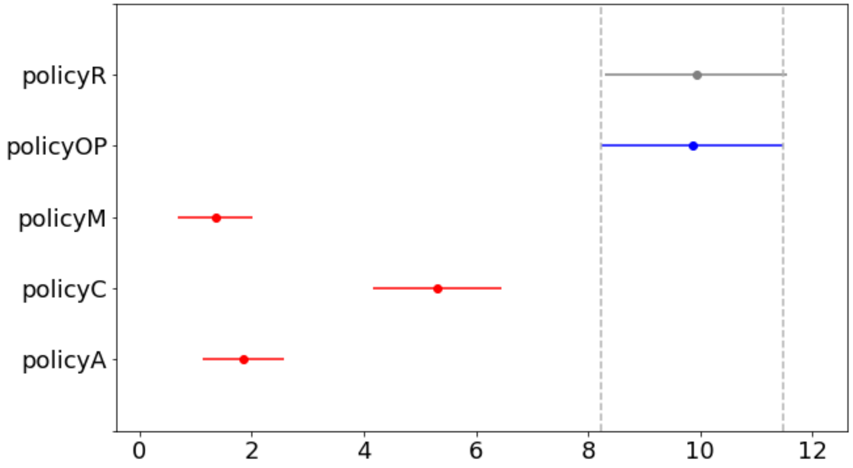
ANOVA p-value for waiting time		
sensor pi3	1.248e-30	<= 0.05
sensor pi4	7.893e-14	<= 0.05

  - ‘absolute error difference’ (ML model discrepancy w.r.t. predictability)

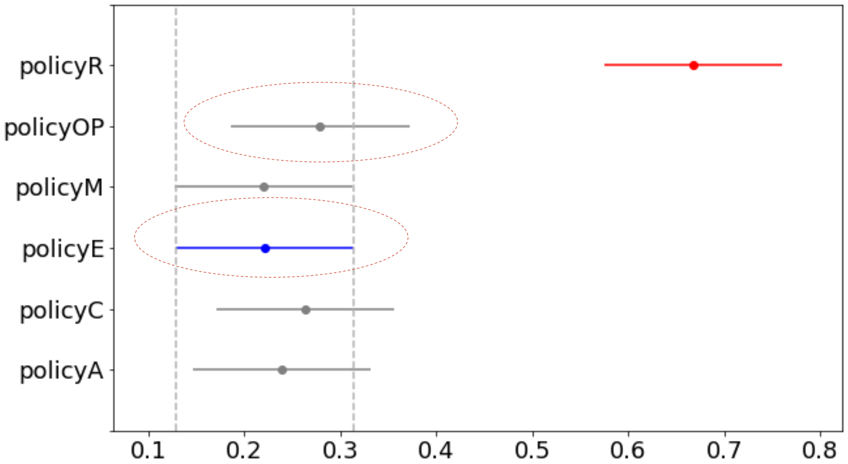
ANOVA p-value for abs error		
sensor pi3	1.244e-13	<= 0.05
sensor pi4	2.723e-17	<= 0.05

Tukey's HSD Test: Linear Regression

The waiting time for OP has a higher mean and the difference is **statistically significant**.

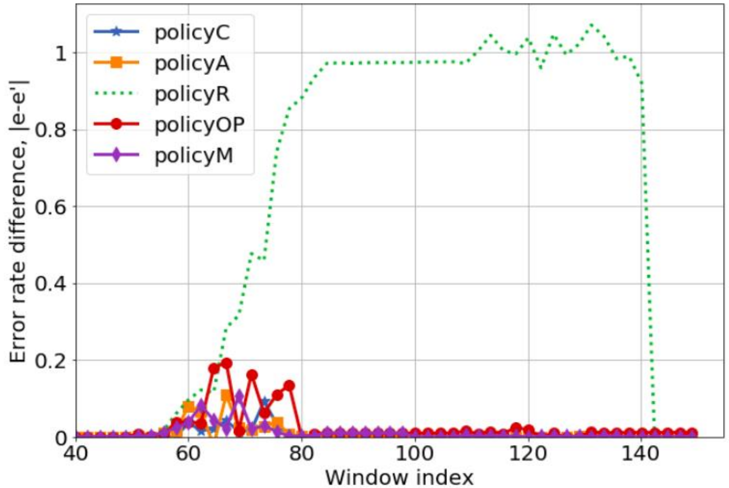


The difference in the absolute error difference between OP and the policy that sends model updates constantly is **not statistically significant**.

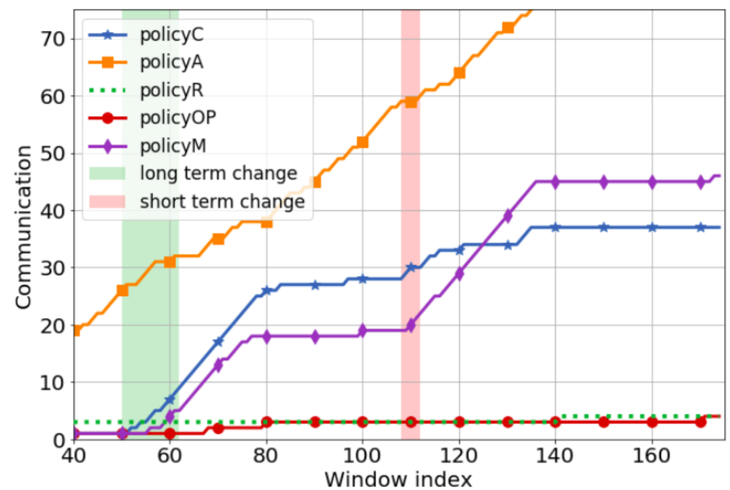


Support Vector Regression Model

- The absolute error difference for the OP deviates the most from the Accurate Policy



- OP waits on average **30 times** longer than the other policies



ANOVA Test: Support Vector Regression

- Statistical significant difference b/w policies:
  - ‘waiting time’

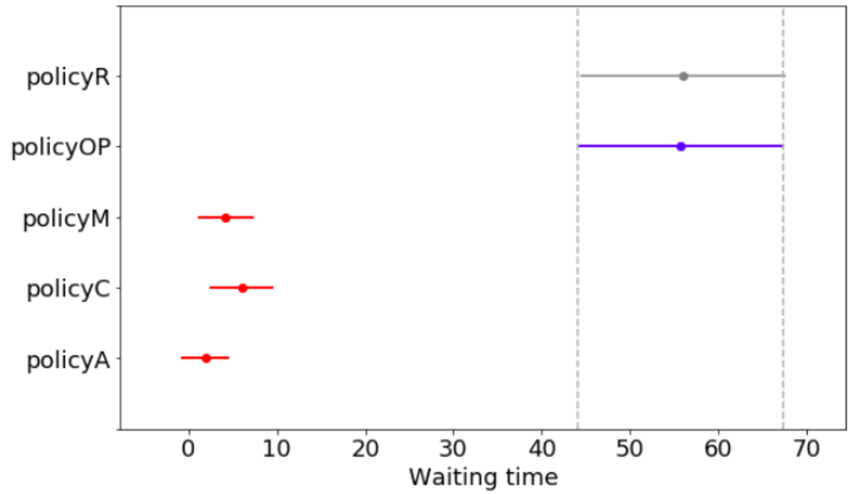
ANOVA p-value for waiting time		
sensor R3	7.52e-28	<= 0.05
sensor R5	9.96e-27	<= 0.05

  - ‘absolute error difference’

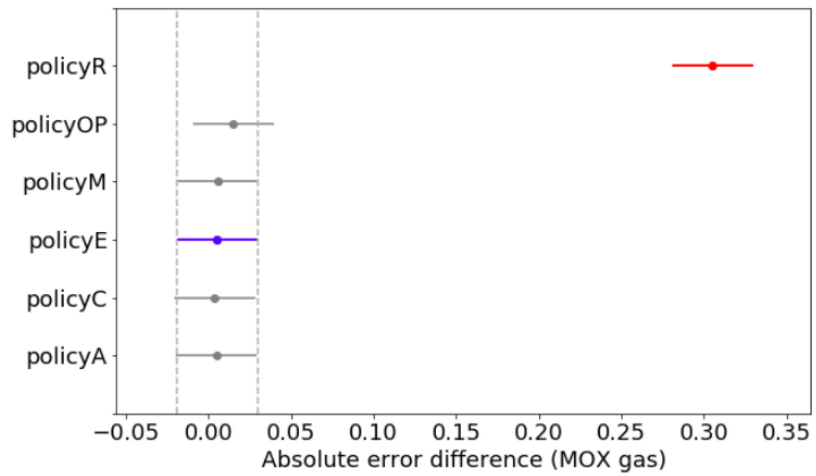
ANOVA p-value for abs error		
sensor R3	2.56e-90	<= 0.05
sensor R5	2.79e-03	<= 0.05

Tukey's HSD Test: Support Vector Regression

The <u>waiting time</u> for OP has a higher mean and the difference is **statistically significant**.



OP and the other policies have **no statistically significant** difference in the <u>absolute error difference</u>.



Conclusions

Policy type	High quality prediction models	Lower quality prediction model
CuSum	high communication	☺
Accuracy-based	high communication	high communication
Optimal Policy	☺	☺
Median-based	high communication	high communication

University of Glasgow
School of Computing Science

ESSENCE
PERVASIVE & DISTRIBUTED INTELLIGENCE

Thank you!

Katie Aleksandrova