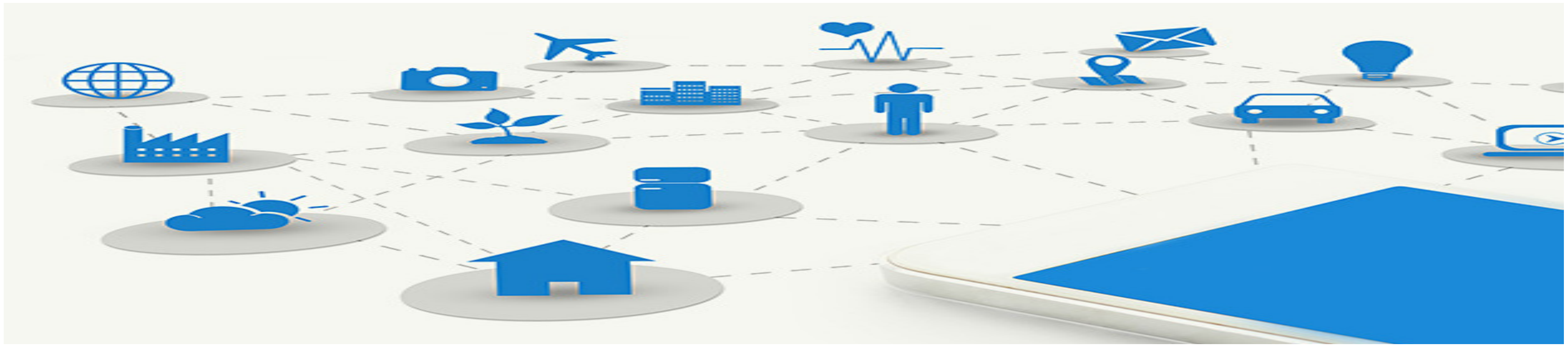
ADMin
Adaptive Monitoring Dissemination for the Internet of Things

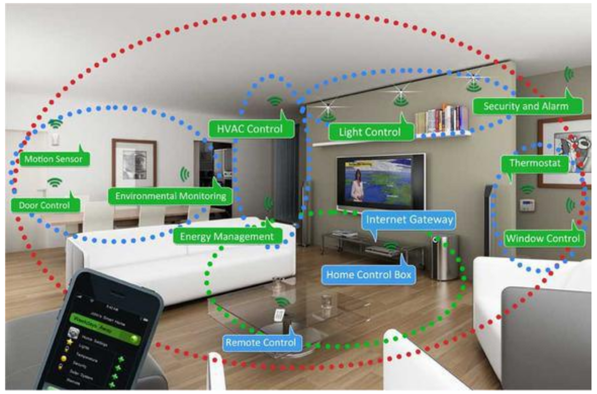


Demetris Trihinas, George Pallis, Marios D. Dikaiakos

{**trihinas**, gpallis, mdd}@cs.ucy.ac.cy

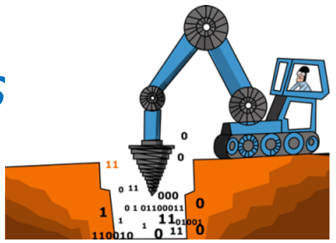
Department of Computer Science
University of Cyprus

The Internet of Things



The physical world is now becoming an information system

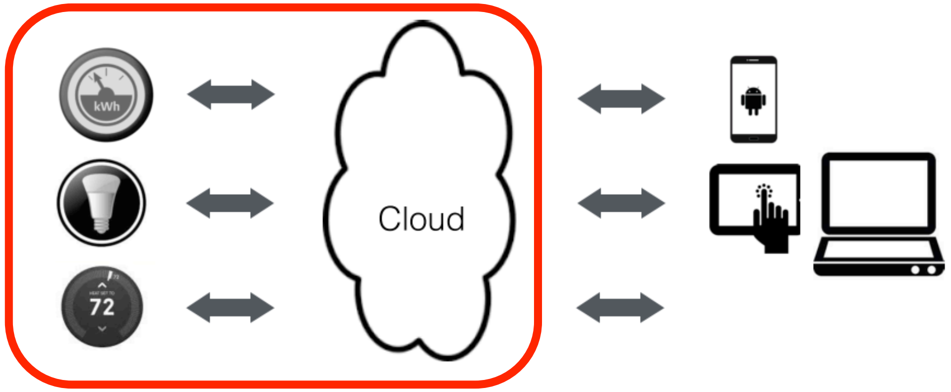
Physical (battery-powered) and network-enabled devices with smart processing capabilities…

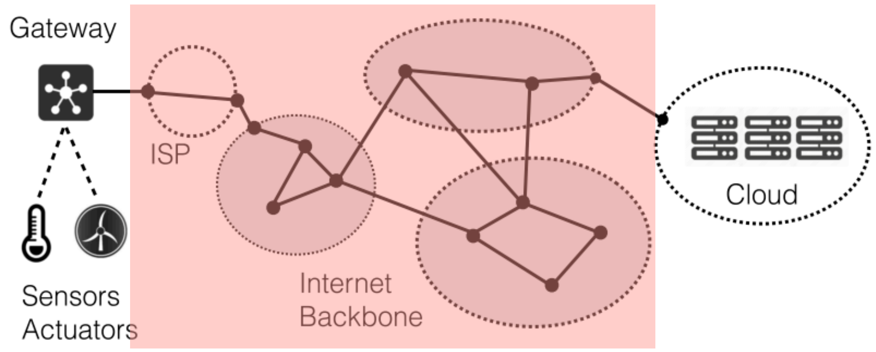


Exchanging continuous data streams with other network-enabled devices, systems and humans…

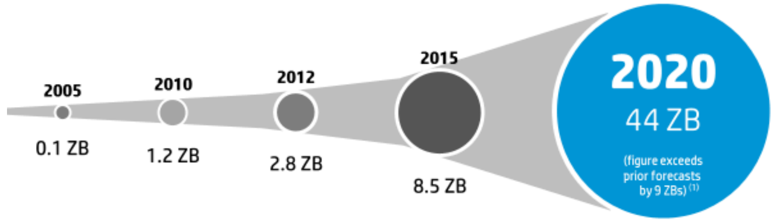
Augmenting IoT with the Cloud

The IoT Developer Perspective





Data explosion outpacing technology [HP, Oct 2016]



[Cisco VNI, 2016]

IP Traffic 2.3ZB by 2020, 58% from IoT and (video) streaming

The Harsh Reality

Network latencies, bandwidth limitations,

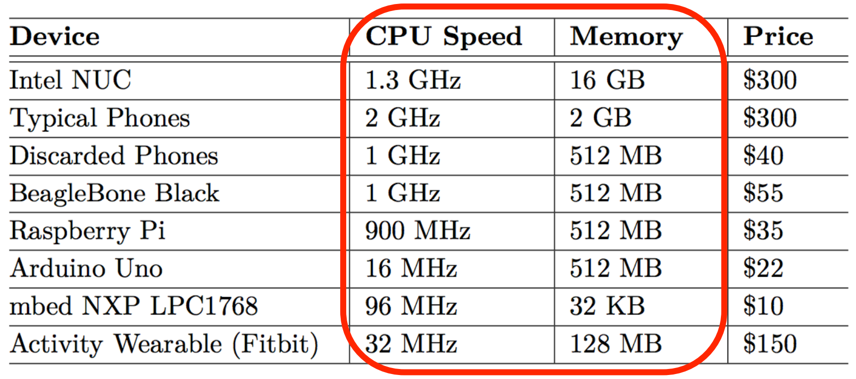
pricing (**in/out cloud traffic is billed**),

constant location awareness

"The cloud is not enough: Saving IoT from the cloud", B. Zhang et al., Usenix HotCloud 2015
"Taking the internet to the next physical level", V. Cerf and M. Senges, IEEE Computer, 2016

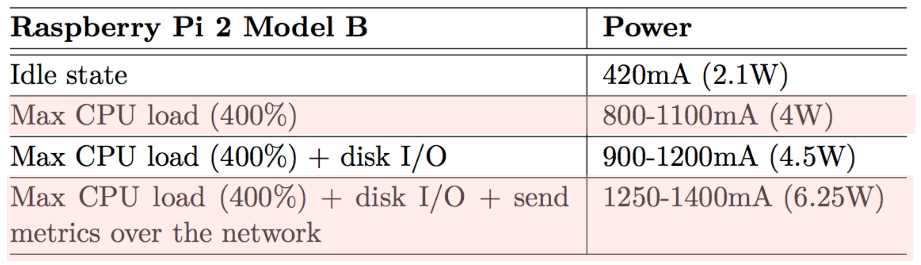
Augmenting Edge Computing with the Cloud

Challenge 1 – Taming data volume and data velocity with limited processing and network capabilities in the IoT/Edge realm



Device	CPU Speed	Memory	Price
Intel NUC	1.3 GHz	16 GB	$300
Typical Phones	2 GHz	2 GB	$300
Discarded Phones	1 GHz	512 MB	$40
BeagleBone Black	1 GHz	512 MB	$55
Raspberry Pi	900 MHz	512 MB	$35
Arduino Uno	16 MHz	512 MB	$22
mbed NXP LPC1768	96 MHz	32 KB	$10
Activity Wearable (Fitbit)	32 MHz	128 MB	$150

Challenge 2 - IoT devices are usually battery-powered which means intense processing leads to less battery-life



Raspberry Pi 2 Model B	Power
Idle state	420mA (2.1W)
Max CPU load (400%)	800-1100mA (4W)
Max CPU load (400%) + disk I/O	900-1200mA (4.5W)
Max CPU load (400%) + disk I/O + send metrics over the network	1250-1400mA (6.25W)

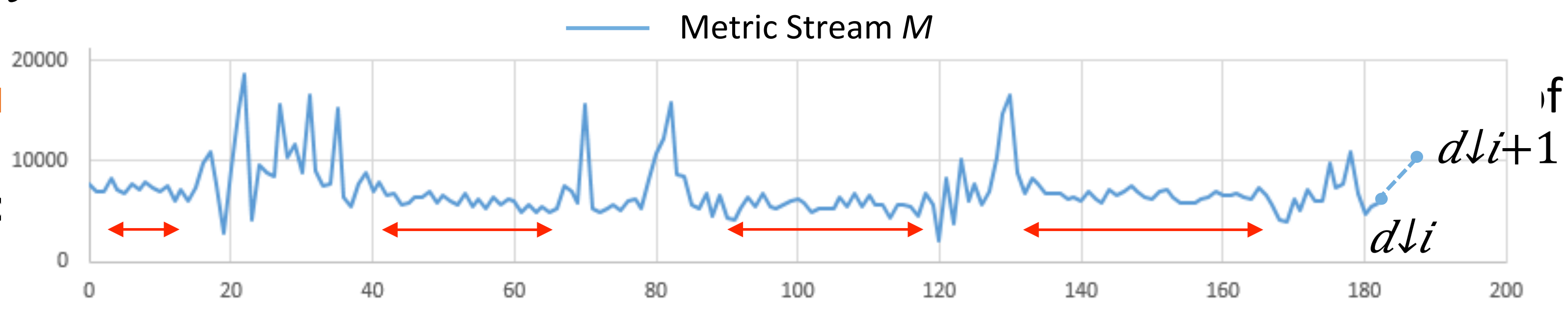
Processing and **data dissemination** are the main energy drains in embedded and mobile devices

"AdaM: Adaptive Monitoring Framework for Sampling and Filtering on IoT Devices", D. Trihinas et al., IEEE BigData 2015

Adaptive Dissemination

Preliminaries

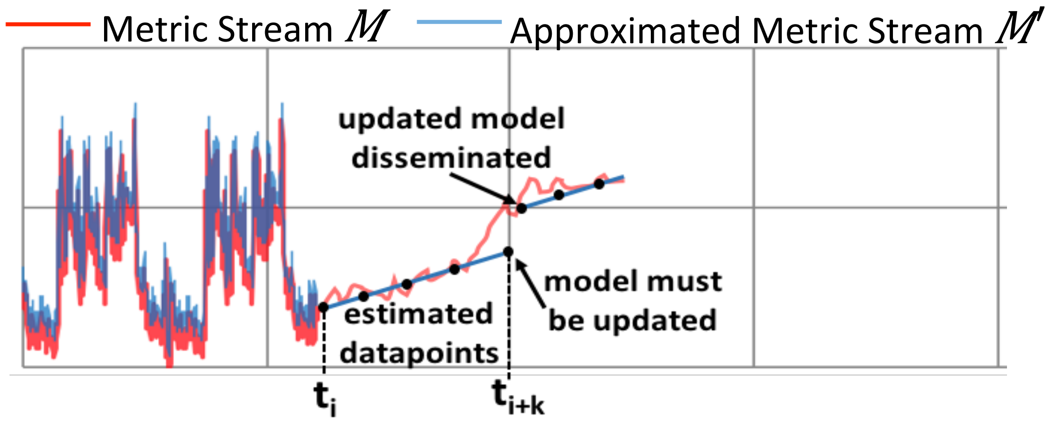
- A **metric stream** $M=\{d_i\}_{i=0}^{n}$ published by a monitoring source is a large stochastic sequence of i.i.d datapoints, denoted as d_i, where $i=0, 1,\ldots,n$ and $n\rightarrow\infty$
- A **datapoint** d_i is a tuple (m_{id}, t_i, v_i) described by a unique identifier for the monitoring source m_{id}, a **timestamp** t_i and a **value** v_i
- **Pe** f a mc disseminated at



Energy is wasted while generating large data volumes at a high velocity

Model-Based Adaptive Dissemination

- Dynamically adapt dissemination rate by **applying approximation techniques** to sensed datapoints to reduce communication overhead

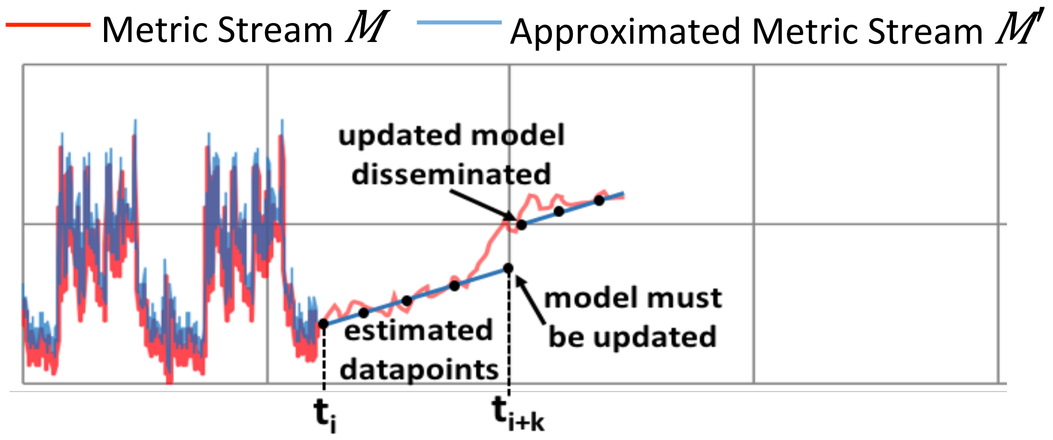


- Monitoring source maintains **runtime estimation model** $\rho(M)$ capturing monitoring stream evolution and variability

- At the $i\uparrow th$ time interval instead of metric values, the **model is disseminated**

- Receiving entities **predict the IoT device state** from given model assuming subsequent k-datapoints can be approximated $d\downarrow i+k|i = f(\rho(M), d\downarrow i)$ **within given**

Model-Based Adaptive Dissemination



- Monitoring source **withholds further dissemination** interacting with receiver only when **shifts in monitoring stream value distribution** render model as inconsistent with the actual IoT device state

decision function?

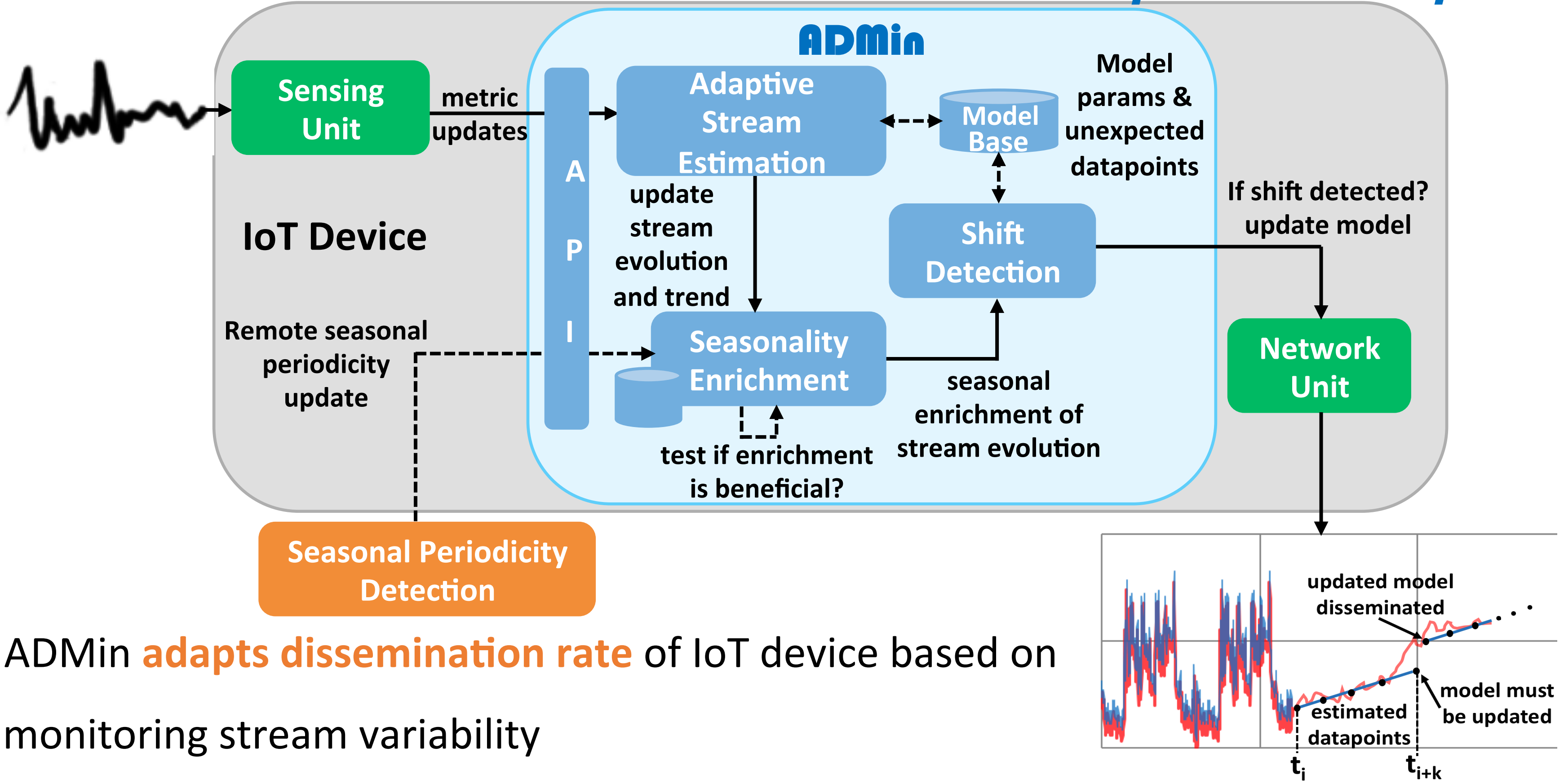
$$g(M, M', t) = \begin{cases} \text{trigger dissemination}, & dist > \eta(\delta) \\ \text{suppress dissemination}, & \text{otherwise} \end{cases}$$

decision criteria?

- If **model parameterization is inconsistent**, at this point, it must be updated

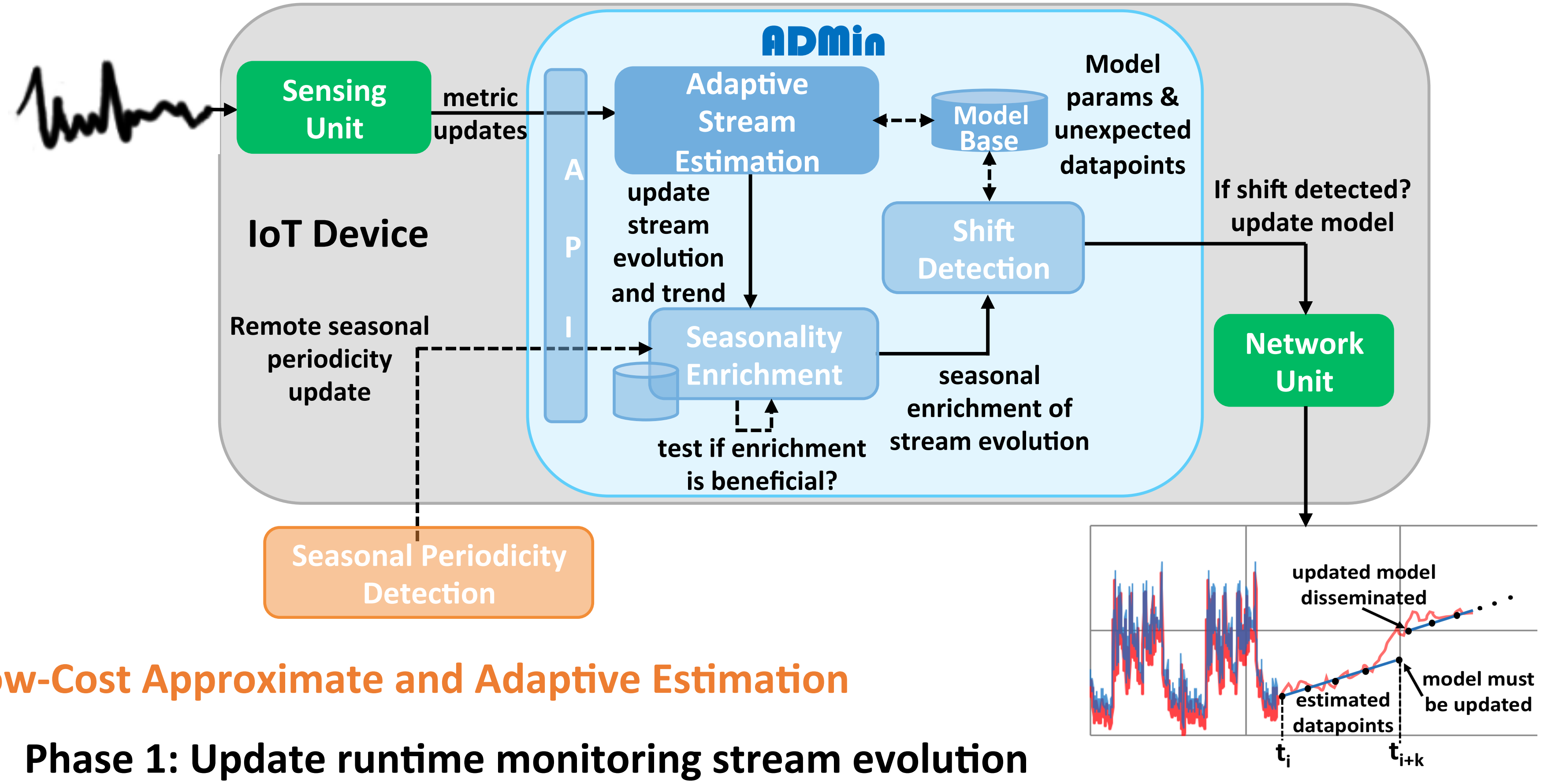
The ADMin Framework

Disseminate model updates not datapoints…


ADMin
IoT Device
Sensing Unit
metric updates
A P I
Adaptive Stream Estimation
Model Base
Model params & unexpected datapoints
update stream evolution and trend
Shift Detection
If shift detected? update model
Network Unit
Seasonality Enrichment
seasonal enrichment of stream evolution
test if enrichment is beneficial?
Remote seasonal periodicity update
Seasonal Periodicity Detection
updated model disseminated
model must be updated
estimated datapoints
t_i
t_{i+k}

- ADMin **adapts dissemination rate** of IoT device based on monitoring stream variability
- Reduces on device **energy consumption** and **volume** and **velocity** of data disseminated in streaming networks
- Balance between efficiency and accuracy with **low-cost estimation process O(1)**

The ADMin Framework


ADMin
Sensing Unit
metric updates
A P I
Adaptive Stream Estimation
Model Base
Model params & unexpected datapoints
update stream evolution and trend
Shift Detection
If shift detected? update model
IoT Device
Remote seasonal periodicity update
Seasonality Enrichment
Network Unit
test if enrichment is beneficial?
seasonal enrichment of stream evolution
Seasonal Periodicity Detection
updated model disseminated
model must be updated
estimated datapoints
t_i
t_{i+k}

Low-Cost Approximate and Adaptive Estimation

- **Phase 1: Update runtime monitoring stream evolution**
- **Phase 2: Detect gradual trends in monitoring stream**

Low-Cost Approximate Stream Estimation Model

Phase 1: Update runtime monitoring stream evolution $\rho(\mathrm{M})$

- **Probabilistic** Exponential Weighted Moving Average (**PEWMA**) to estimate v_{i+1} from μ_i and the standard deviation σ_{i+1} :

$$v_{i+1} = \mu_i = \alpha \mu_{i-1} + (1-\alpha) v_i$$

Looks like an exponential moving average, right?

But weighting $a = \alpha(1 - \beta \mathrm{P}_i)$ is **probabilistically** applied!

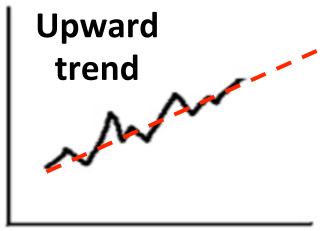
- Datapoints labelled as **“expected”** if estimation lands in prediction intervals determined from user **confidence** guarantees or **“unexpected”** otherwise

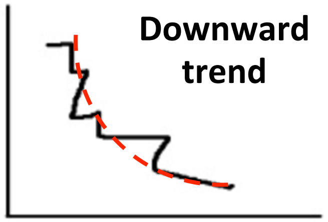
Low-Cost Approximate Stream Estimation Model

Phase 2: Detect over time **gradual trends** in monitoring stream to **reduce "lagging"** effects in monitoring stream evolution estimation

- **Holt's Trend Method** used to bring moving average to appropriate value base

$$X_i = \begin{cases} v_i - v_{i-1}, & i = 2 \\ \gamma\ (\mu_i - \mu_{i-1}) + (1-\gamma)\ X_{i-1}, & i > 2 \end{cases}$$

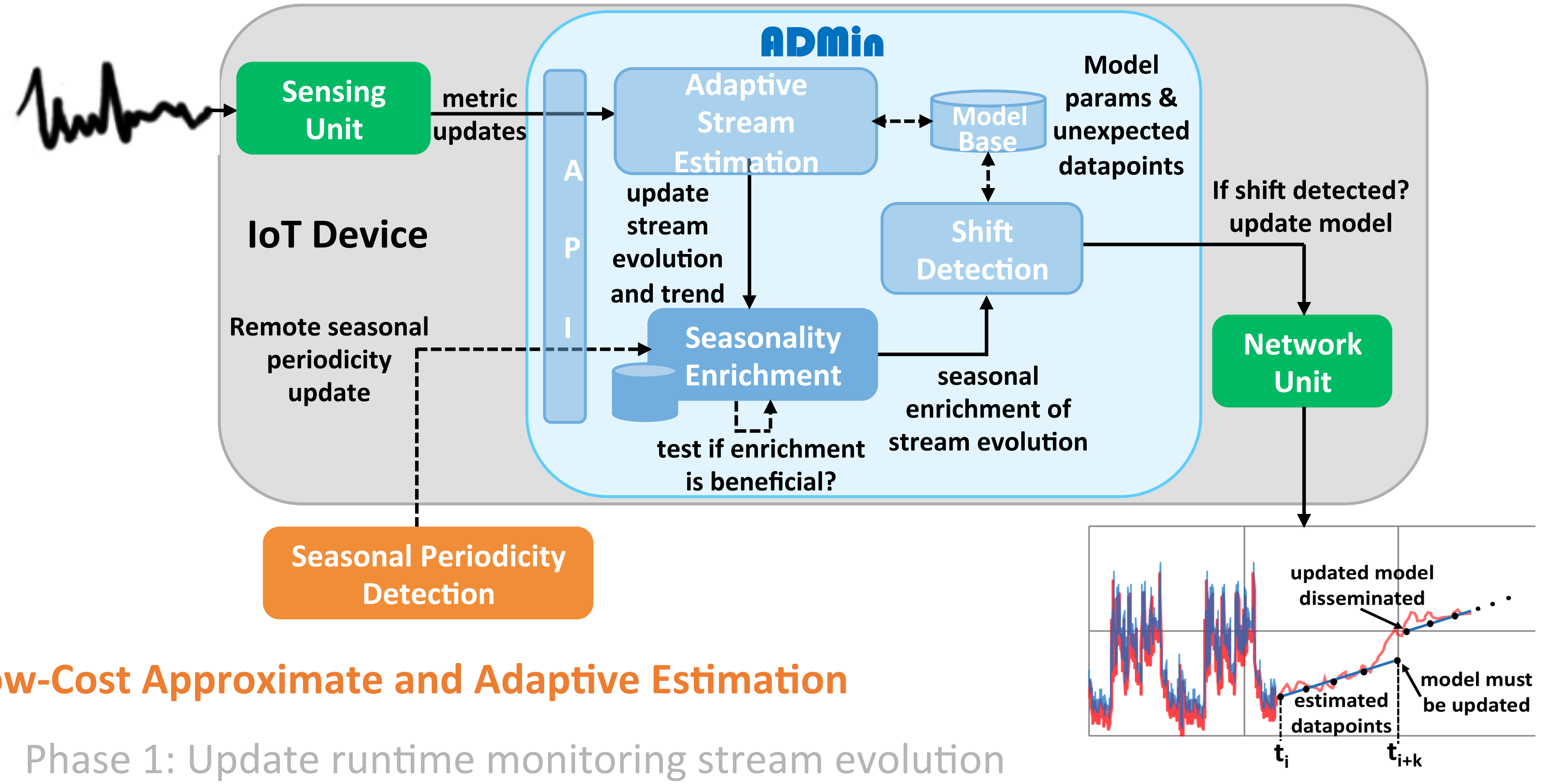




- Improve **forecasting** from 1-step ahead (moving average) predictions to ***k*-datapoint** values

$$v_{i+k|i} \leftarrow \mu_i + k X_i$$

The ADMin Framework


ADMin
Sensing Unit
metric updates
IoT Device
A P I
Adaptive Stream Estimation
Model Base
Model params & unexpected datapoints
update stream evolution and trend
Shift Detection
If shift detected? update model
Network Unit
Remote seasonal periodicity update
Seasonality Enrichment
seasonal enrichment of stream evolution
test if enrichment is beneficial?
Seasonal Periodicity Detection
updated model disseminated
model must be updated
estimated datapoints
t_i
t_{i+k}

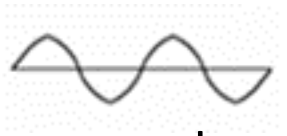
Low-Cost Approximate and Adaptive Estimation

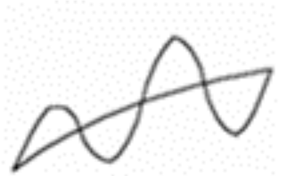
- Phase 1: Update runtime monitoring stream evolution
- Phase 2: Detect gradual trends in monitoring stream
- **Phase 3: Seasonality enrichment**

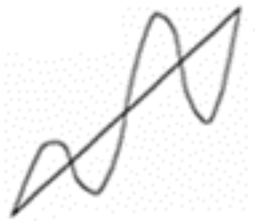
Low-Cost Approximate Stream Estimation Model

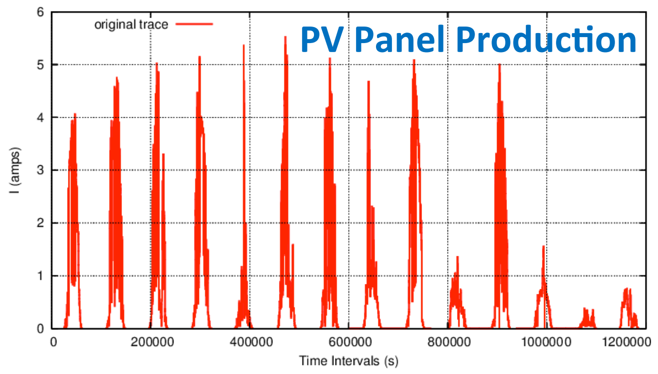
Phase 3: Test if **seasonality enrichment is beneficial** to estimation process and update low-cost approximate model accordingly

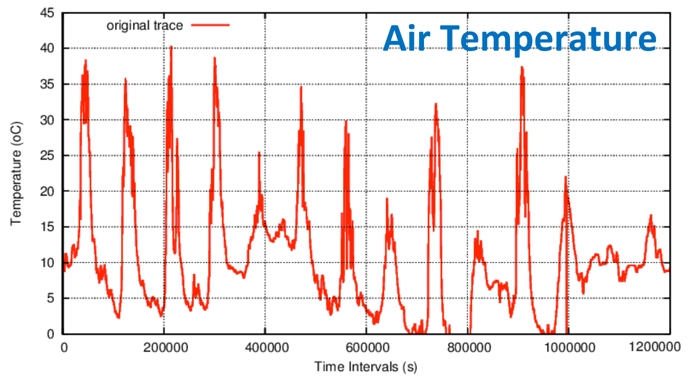
- Tendency of the metric stream to exhibit **behavior that repeats itself** every *L* periods (e.g., hourly)
- Seasonal effects highly evident in **IoT data** (e.g., human biosignals, environmental data)


seasonal


Seasonal with damped trend


Seasonal with exponential trend





Low-Cost Approximate Stream Estimation Model

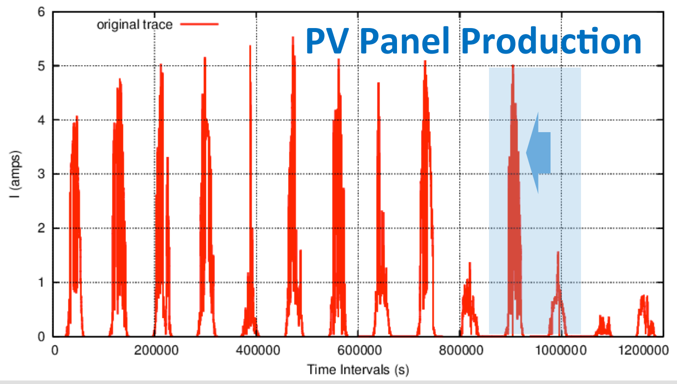
- **Holt Winter's Method** used to estimate **seasonal contribution**

$$S_i = \begin{cases} 0, & i < L \\ \omega\ (v_i - \mu_i - X_i) + (1-\omega)\ (v_i - S_{i-L}), & i > L \end{cases}$$

- Forecasting k-subsequent datapoints with trend and seasonality

$$v_{i+k|i} \leftarrow \mu_i + k X_i + S_i$$

- However... **perfect seasonal behavior is rarely observed** in real-life systems



Considering day before hourly average (S_{i-L}) will lead to **overestimation**

Low-Cost Approximate Stream Estimation Model

- Online testing (T-Test) to determine if **seasonal contribution beneficial** to estimation or not

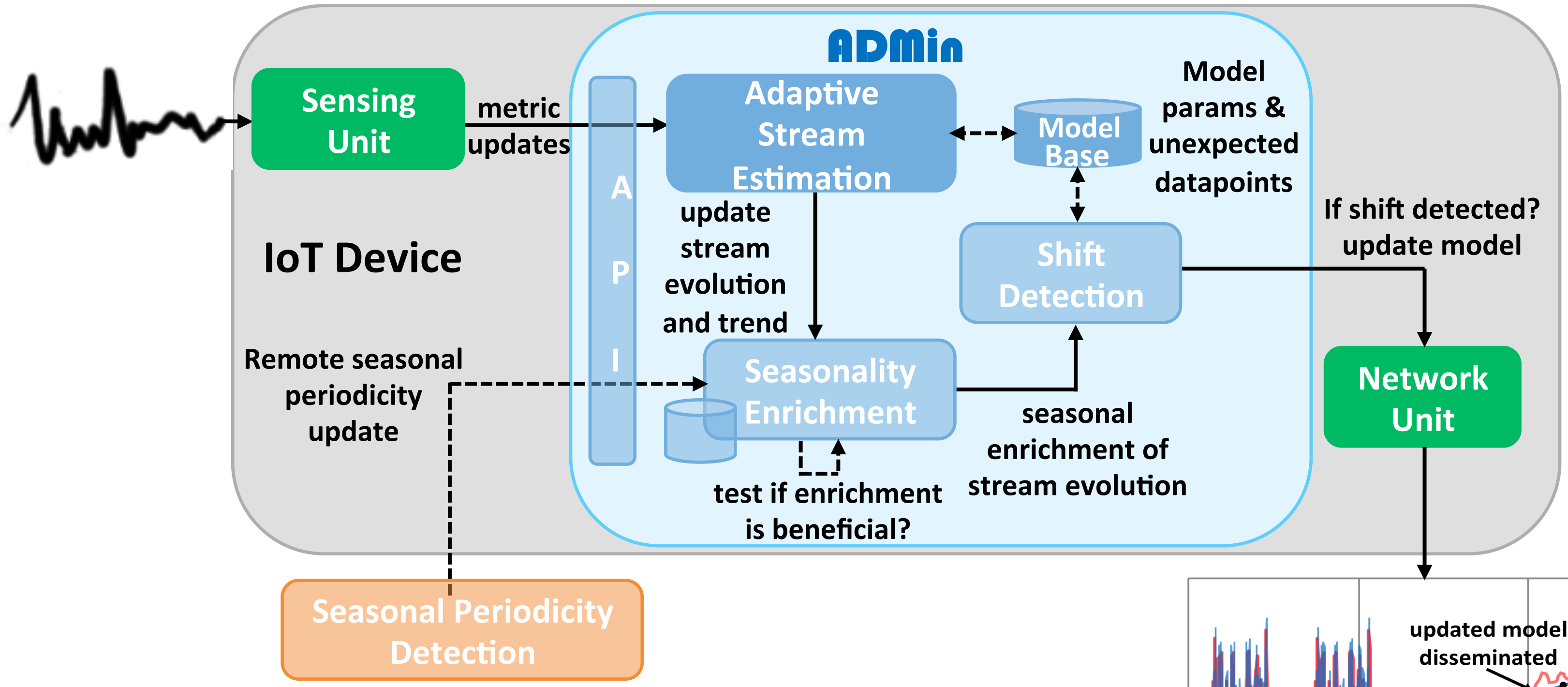
$$v_{i+k|i} \leftarrow \mu_{i} + k X_{i} \cancel{+ S_{i}}$$

- Detecting **optimal seasonality cycle** (*L*) is an open research challenge especially when different cycles exist in monitoring stream

- Approximate runtime seasonal periodicity detection
    - **ComCube Framework** (Matsubara et al., WWW, 2016): lightweight tensor-based and parameter-free framework for **near-optimal seasonal periodicity** detection

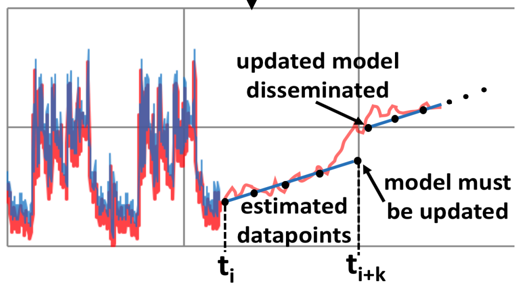
"Non-linear mining of competing local activities ", Y. Matsubara, Y. Sakurai and C. Faloutsos, **WWW 2016**

The ADMin Framework


ADMin
Sensing Unit
metric updates
A P I
Adaptive Stream Estimation
Model Base
Model params & unexpected datapoints
update stream evolution and trend
Shift Detection
If shift detected? update model
IoT Device
Remote seasonal periodicity update
Seasonality Enrichment
Network Unit
seasonal enrichment of stream evolution
test if enrichment is beneficial?
Seasonal Periodicity Detection

Low-Cost Approximate and Adaptive Estimation

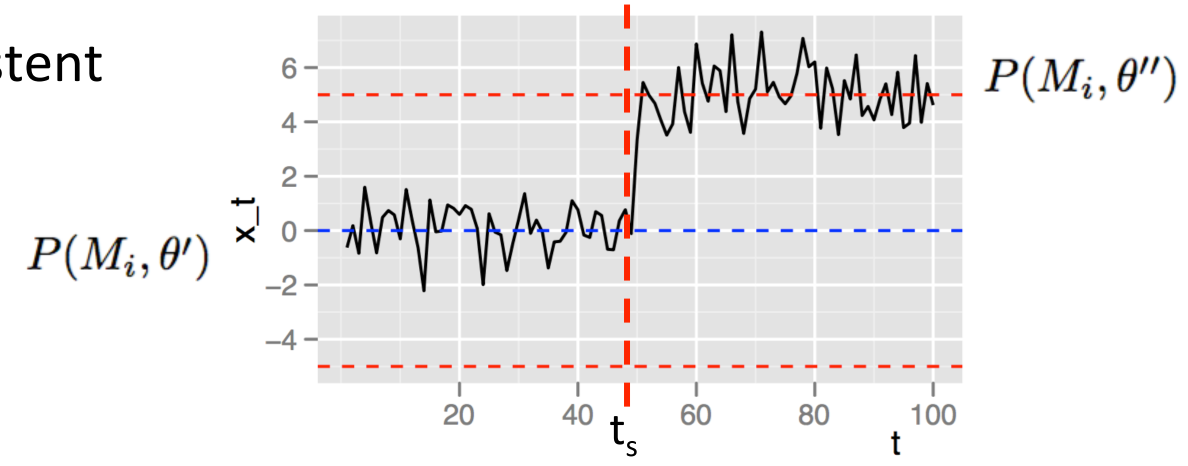
- Phase 1: Update runtime monitoring stream evolution
- Phase 2: Detect gradual trends in monitoring stream
- Phase 3: Seasonality enrichment



Detect Shifts in Monitoring Stream Evolution

Detecting Shifts in a Monitoring Stream

- Cumulative Sum (CUSUM) log-likelihood test to **detect shifts in monitoring stream value distribution** which render estimation model as inconsistent



1. log-likelihood ratio

$$c_i = \ln \frac{P(M_i, \theta'')}{P(M_i, \theta')}$$

after shift

prior shift

2. Detecting shifts in mean

$c_i = \ln P(M_i, \mu'')/P(M_i, \mu')$

$\mu_i'' = \mu_i' + \varepsilon$

where ε magnitude of change

However, ε not known beforehand

but… **estimation model provides** us with an **approximate $\boldsymbol{\varepsilon}$**

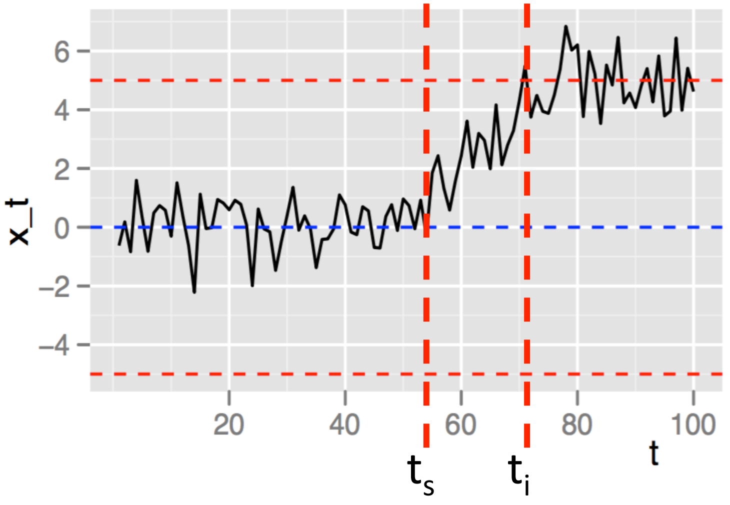
Detecting Shifts in a Monitoring Stream

3. Online CUSUM and Decision Function

$$G_{i,,\{low,\ high\}} = \{G_{i-1,\{low,\ high\}} + c_i\}^+$$

$$\boxed{if\ G_{i,\{low,\ high\}} > h \rightarrow shift\ detected}$$

h is measured in standard deviation units



4. Actual Shift Time

$$C_{i,\{low,\ high\}} = C_{i-1,\{low,\ high\}} + c_i$$

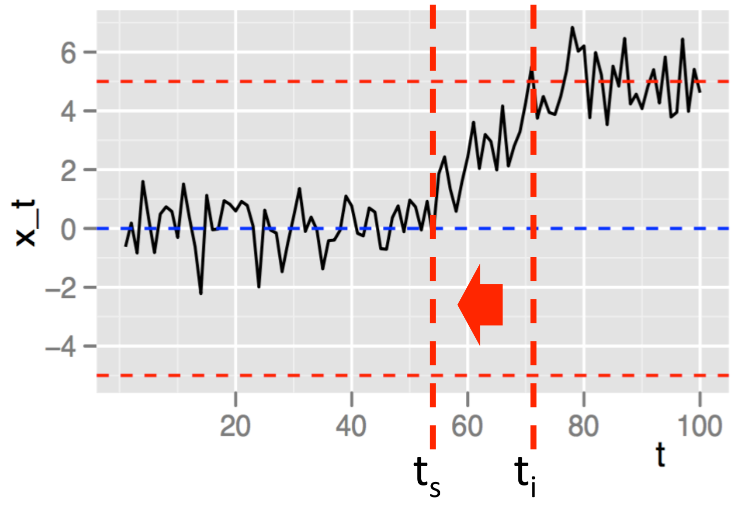
$$t_s = \arg\min_{j \le s \le i} (C_{s-1})$$

Challenge 1: Linear time… **Can t_i be used instead of t_s?**

the time the CUSUM detects the shifts

t_i may greatly from t_s differ in cases of **gradual trends**

Detecting Shifts in a Monitoring Stream



Trend and **seasonality knowledge** provide model with **greater accuracy** ($\varepsilon \rightarrow_{\top} \varepsilon$) and to adapt to unexpected, abrupt and and volatile changes is monitoring stream

Challenge 2: CUSUM threshold *h* static and **sensitive** when stream variability is low ($\sigma \rightarrow 0$) thus triggering… **false alarms**

Adapt CUSUM sensitivity by adapting *h* after dissemination triggered and restrict *h* with h_{min}

$$h_i = \max\{h_{min},\ h(\delta_i, \sigma_i)\}$$

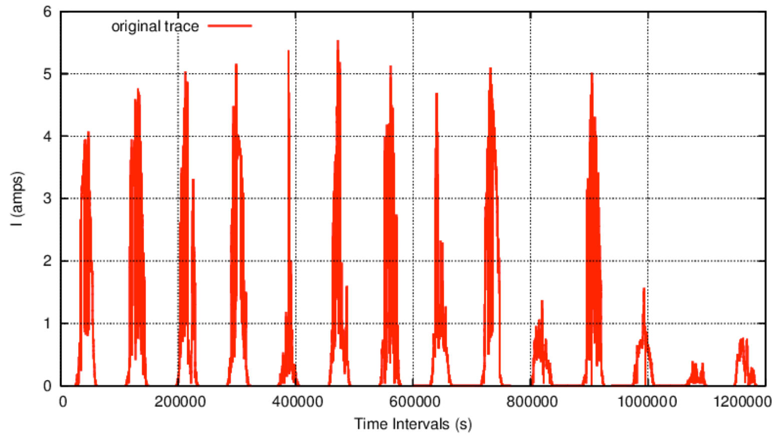
Evaluation

ADMin Evaluation

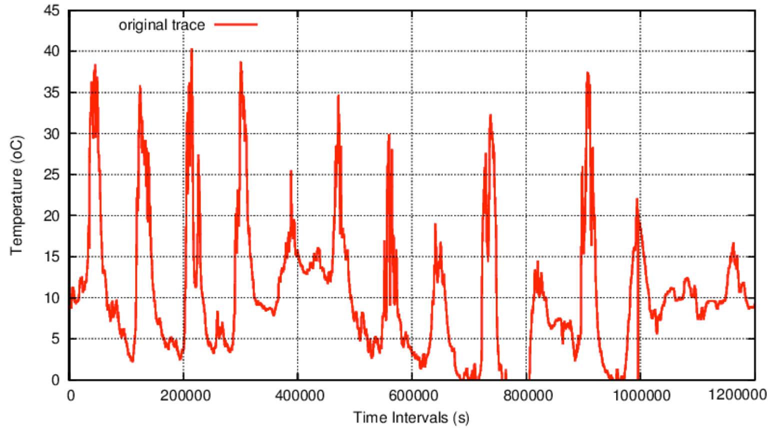
- **ADMin** in 3 configurations
  - No seasonality enrichment (**ADMin**)
  - Static seasonality enrichment - previous day hourly average (**ADMin_S1**)
  - Dynamic seasonality enrichment - ComCube integration (**ADMin_S2**)
- Under comparison frameworks
  - **LANCE** [Werner et al., ACM SenSys 2011]
  - **G-SIP** [Gaura et al., IEEE Trans. on Sensors 2013]
  - **ADWIN** [Bifet et al., SIAM 2010]

All three under-comparison framework parameters configured to output best results

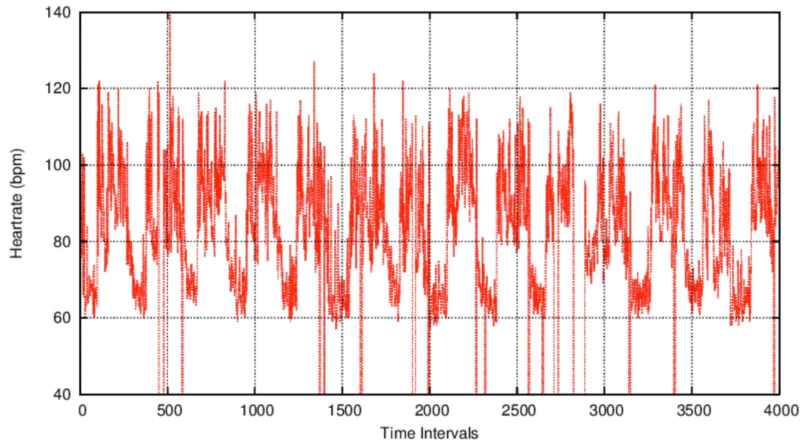
Real-World Datasets



Photovoltaic Panel Current (I_{DC}) Production
Periodicity: 1 second, Duration: 2 weeks (Jan 2015)



Weather Station Air Temperature (°C)
Periodicity: 1 second, Duration: 2 weeks (Jan 2015)



Wearable Human Heartrate (bpm)
Periodicity: 1 min, Duration: 1 month (Mar 2016)

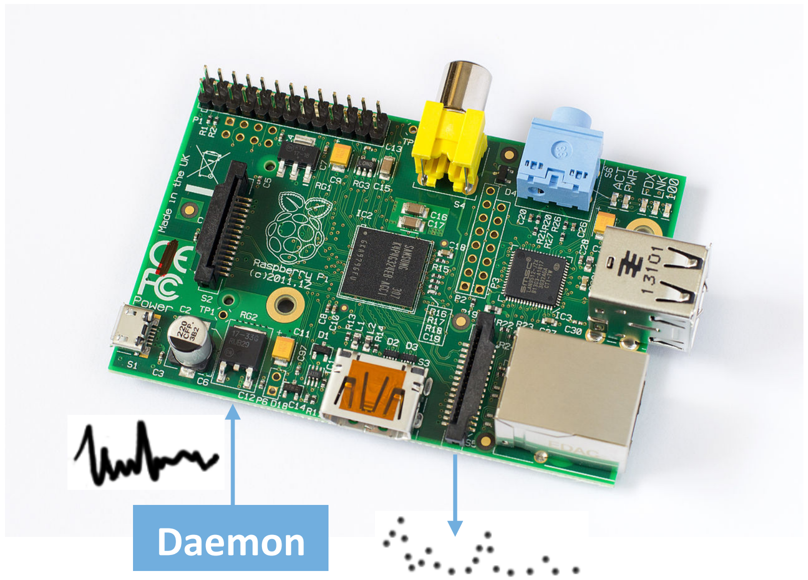
Fitbit Data Extractor

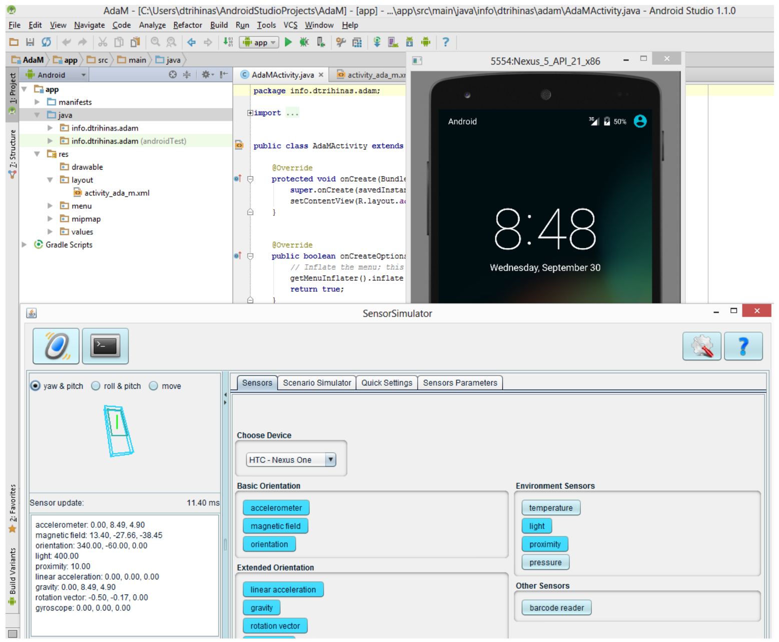
Open-sourced to extract YOUR own RAW data from fitbit

(steps, heartrate, calories, active minutes, distance)

https://github.com/dtrihinas/FitbitDataExtractor

“Big Data” Testbeds





```
package info.dtrihinas.adam;

import ...

public class AdaMActivity extends

    @Override
    protected void onCreate(Bundle
        super.onCreate(savedInstar
        setContentView(R.layout.ac
    }

    @Override
    public boolean onCreateOptions
        // Inflate the menu; this
        getMenuInflater().inflate
        return true;
    }
```

```
Sensor update:                11.40 ms

accelerometer: 0.00, 8.49, 4.90
magnetic field: 13.40, -27.66, -38.45
orientation: 340.00, -60.00, 0.00
light: 400.00
proximity: 10.00
linear acceleration: 0.00, 0.00, 0.00
gravity: 0.00, 8.49, 4.90
rotation vector: -0.50, -0.17, 0.00
gyroscope: 0.00, 0.00, 0.00
```

Raspberry Pi 1st Gen. Model B
512MB RAM, Single Core 700MHz

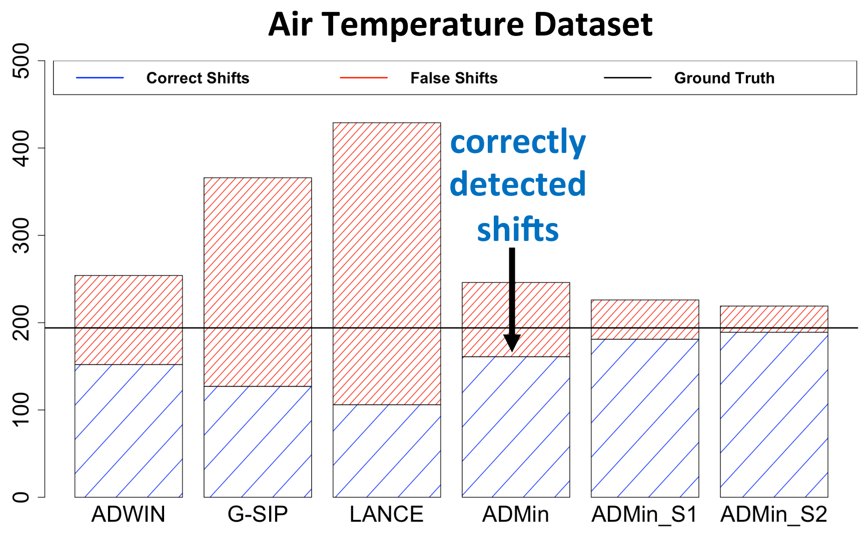
Daemon emulates **PV** and **Temperature** trace behavior while feeding samples to each algorithm

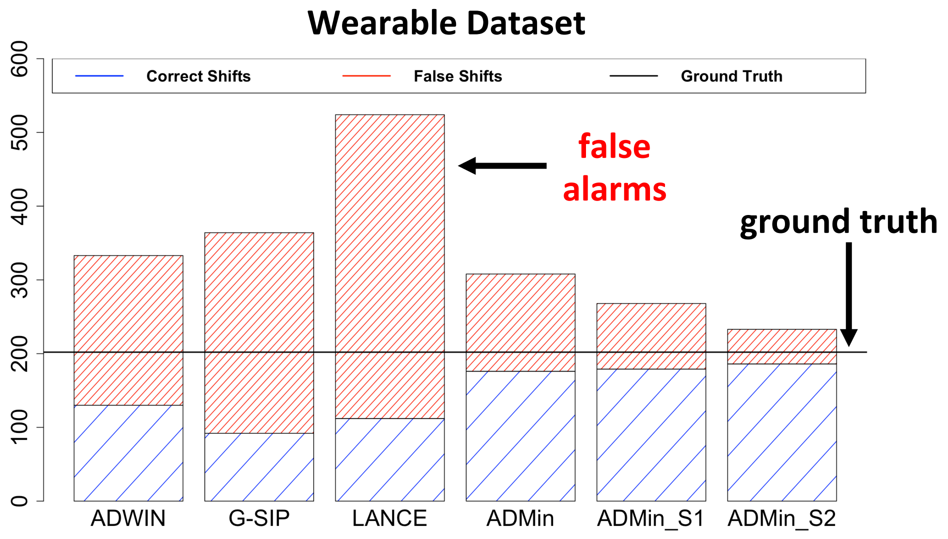
Android Emulator + SensorSimulator
128MB RAM, Single Core ARM 32MHz

SensorSimulator script emulates **heartrate** readings by feeding datapoints to **Android Wear** emulator for processing

Shift Detection Accuracy Evaluation

- Comparison of number of monitoring stream disseminations triggered for shifts that actually occurred (**true positives**) and number of false alarms (**false positives**)
- **Ground truth** pre-determined offline by PELT algorithm [Killick, 2012]





ADMin features **high accuracy (>90%)** and **low false alarm ratio (<10%)** which is drastically reduced when incorporating **seasonality knowledge by at least 47%** compared to the other approaches

Shift Detection Delay Evaluation

- Shift detection delay is the difference in time to when a shift is detected by a technique compared to the actual time of occurrence

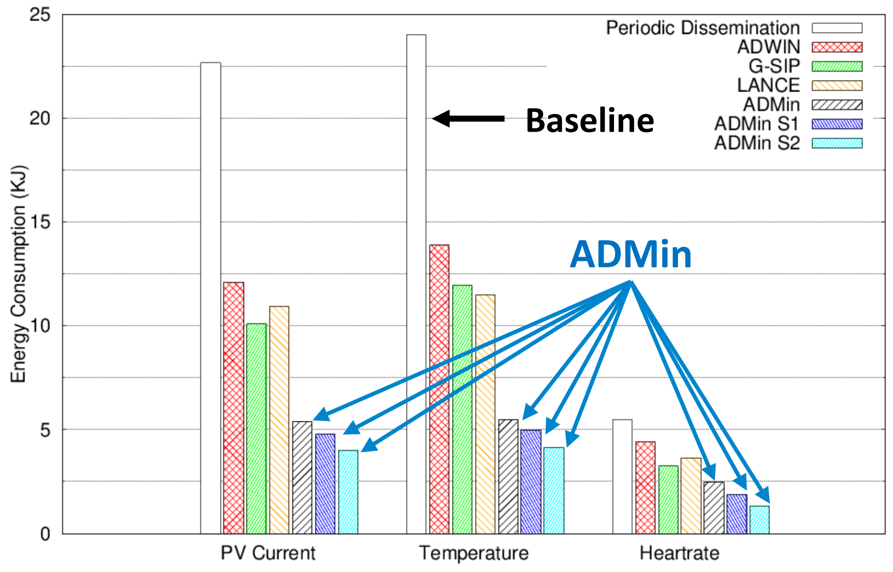
Framework	PV Current (Time Intervals)	Temperature (Time Intervals)	Heartrate (Time Intervals)
ADWIN	9.34 ± 3.47	9.94 ± 3.84	10.39 ± 3.96
G-SIP	10.02 ± 3.96	11.76 ± 4.16	14.17 ± 4.93
LANCE	10.78 ± 4.12	12.63 ± 3.92	15.97 ± 4.12
ADMin	6.04 ± 2.19	7.12 ± 1.97	8.03 ± 2.78
ADMin_S1	3.13 ± 2.03	5.11 ± 2.10	6.22 ± 2.83
ADMin_S2	**2.62 ± 1.94**	**3.23 ± 2.26**	**4.73 ± 2.43**

ADMin outperforms other techniques by at least 29%

When incorporating **trend** and **seasonality knowledge** even for datasets with irregular seasonal behavior **ADMin reduces shift detection time by at least 67%** compared to the other techniques

Overhead Evaluation

- Other than accuracy evaluation **no other study undergoes overhead evaluation!**
- **Periodic dissemination baseline** added with 10 time interval aggregation window

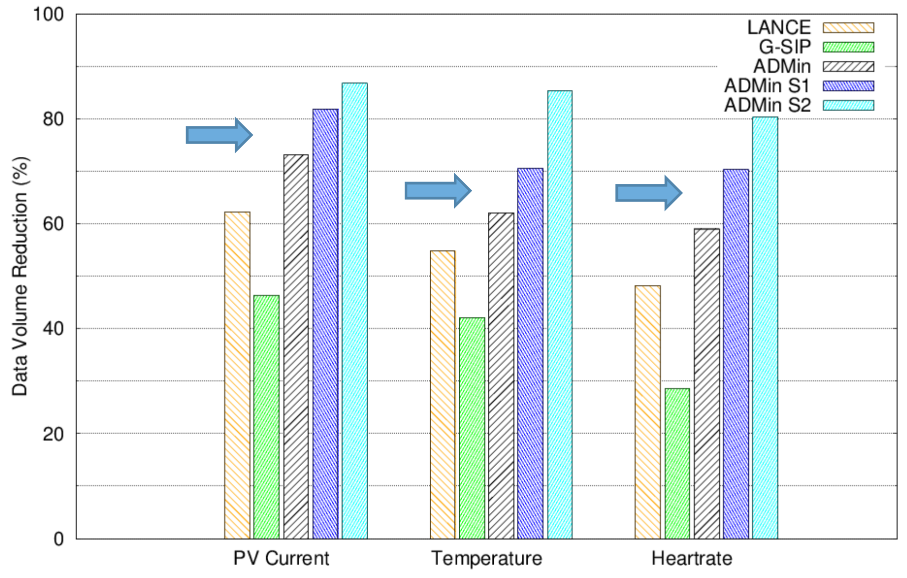


Energy Consumption

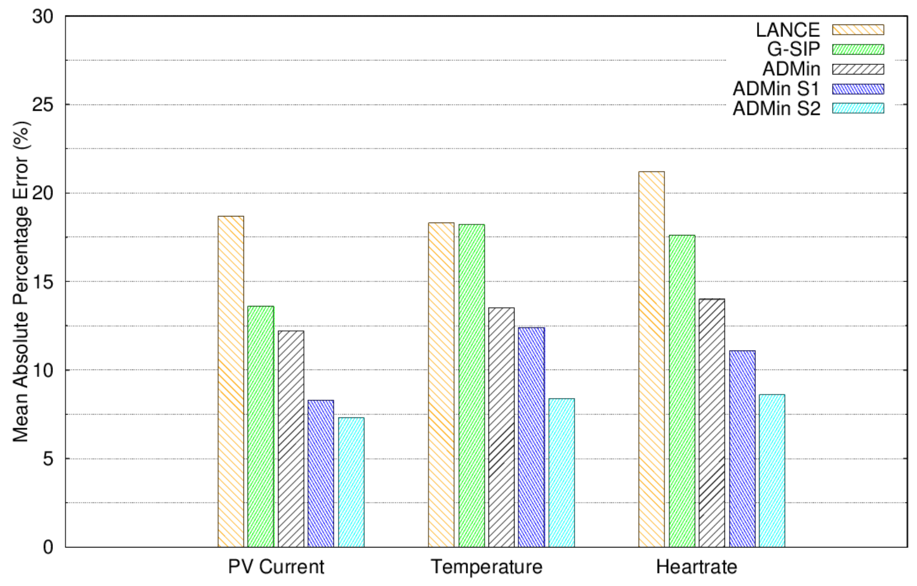
ADMin reduces **energy consumption by at least 76%** and when incorporating **seasonality knowledge by at least 83%**

Overhead Evaluation

- What does the monitoring stream receiver see!



Data Volume Reduction

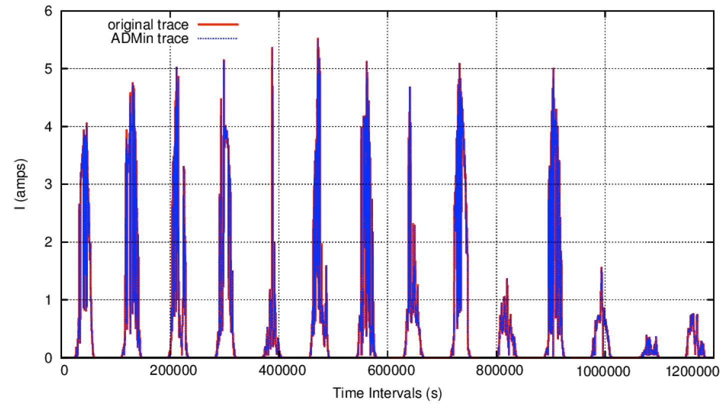


Mean Absolute Percentage Error

ADMin **reduces data volume by at least 60%** while maintaining **accuracy always above 86%**

With **seasonality knowledge data volume is reduced by at least 71%** while **accuracy is always above 91%**

So… does ADMin work?



Photovoltaic Panel Current (I_{DC}) Production

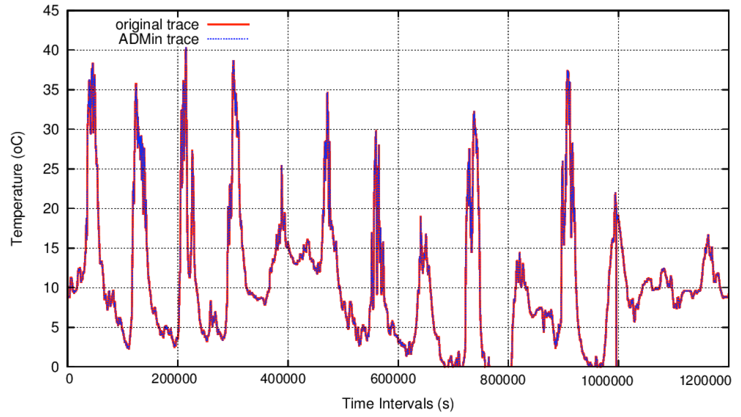
2 Weeks of data collected every 1 second

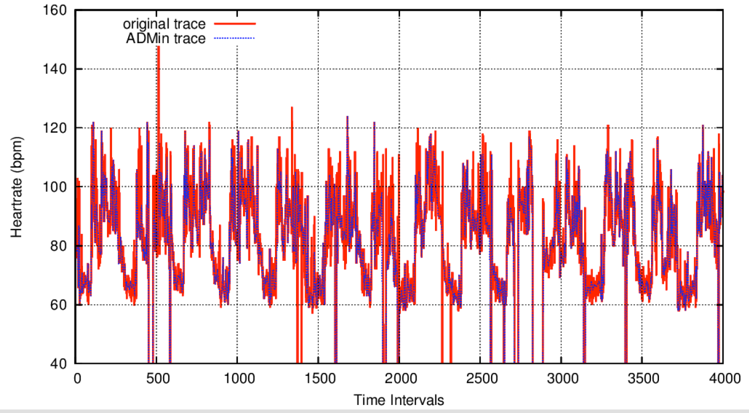
Data reduction: 87% -- Accuracy: 93%

Weather Station Air Temperature (°C)

2 Weeks of data collected every 1 second

Data reduction: 85% -- Accuracy: 92%





Wearable Human Heartrate (bpm)

1month of data collected every 1 minute

Data reduction: 80% -- Accuracy: 90%

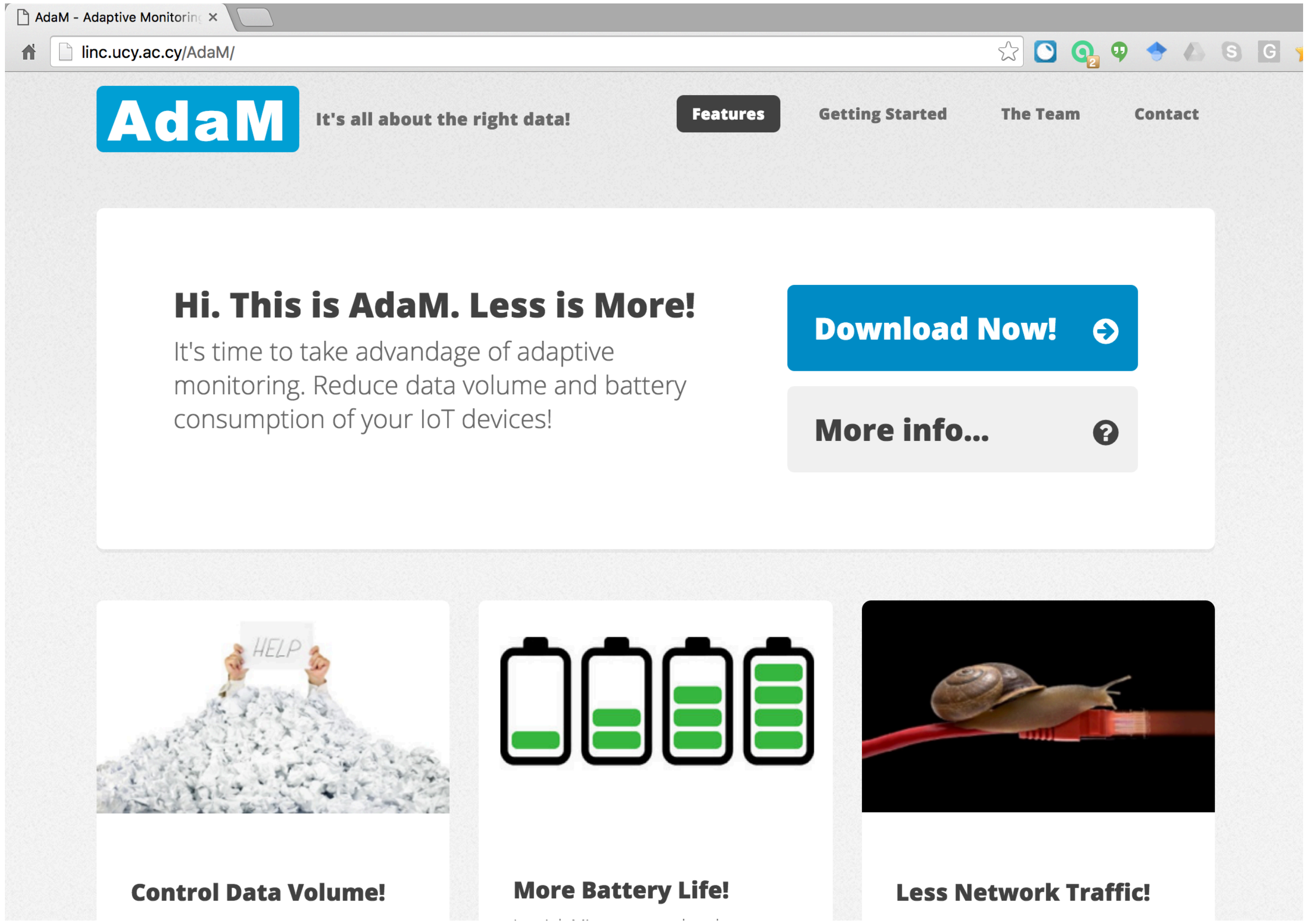
Acknowledgments

UNICORN

Co-funded by the H2020 framework of the European Commission

ADMin is now part of the ADAptive Monitoring framework (AdaM)

http://linc.ucy.ac.cy/AdaM

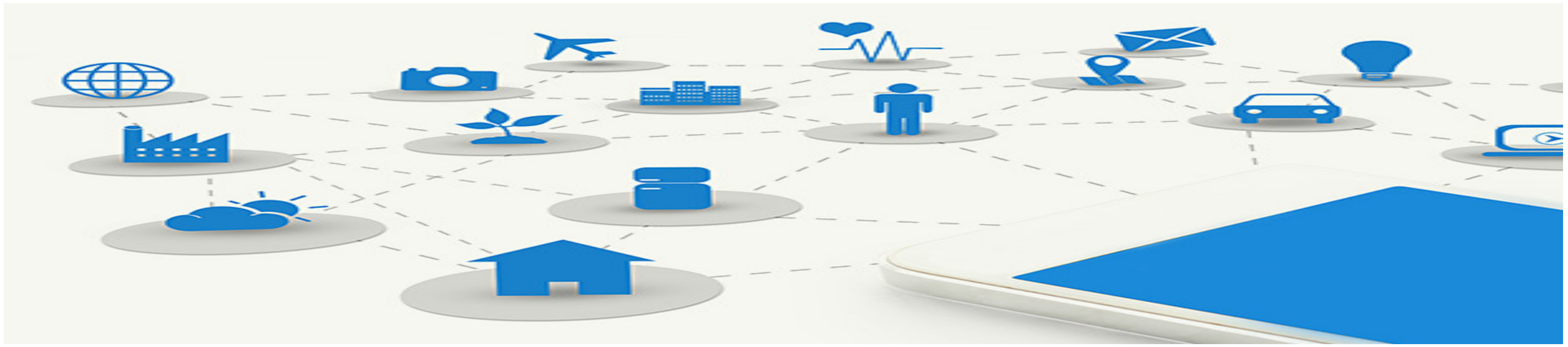

AdaM - Adaptive Monitorin
linc.ucy.ac.cy/AdaM/
AdaM It's all about the right data!
Features Getting Started The Team Contact

Hi. This is AdaM. Less is More!
It's time to take advandage of adaptive monitoring. Reduce data volume and battery consumption of your IoT devices!

Download Now!
More info...

Control Data Volume!
More Battery Life!
Less Network Traffic!

ADMin
Adaptive Monitoring Dissemination for the Internet of Things



Demetris Trihinas, George Pallis, Marios D. Dikaiakos

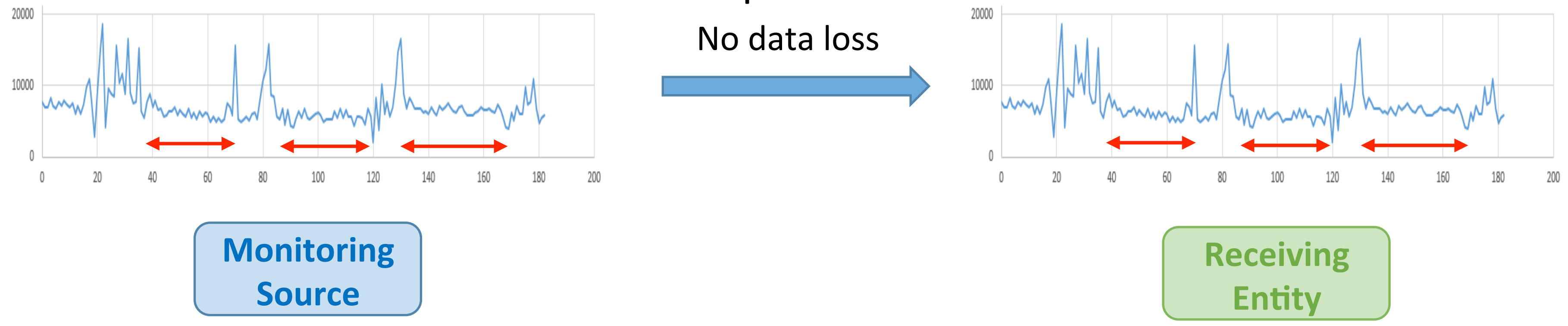
{**trihinas**, gpallis, mdd}@cs.ucy.ac.cy

Department of Computer Science
University of Cyprus

Backup Slides

Periodic Dissemination

- The process of triggering the network controller of a monitored source every T time units such that the i^{th} datapoint is disseminated at time $t_i = i \cdot T$



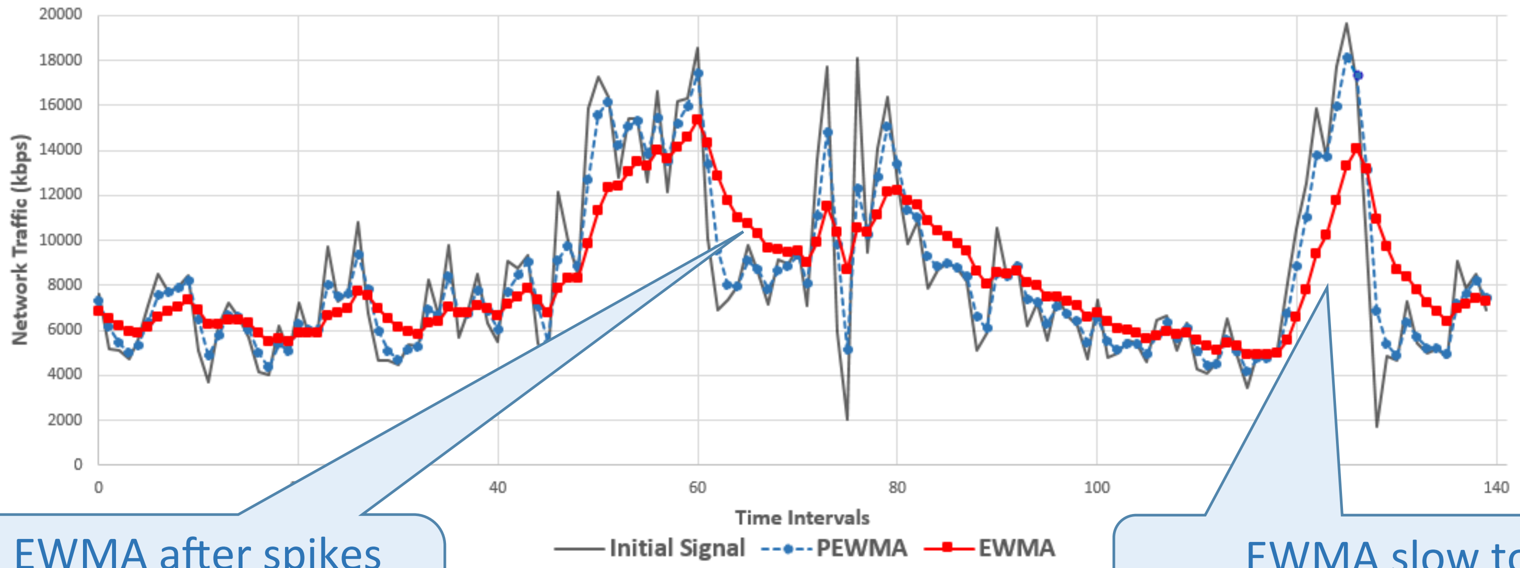
- Cost of Monitoring Dissemination ($\beta_s << \mu_s$)

? $\{ \; \mu_s + \beta_s \cdot \chi \; \}$ datapoint d (t,v)

per message cost   per χ byte cost   -> message compression

Low-Cost Approximate Stream Estimation Model

Why use a Probabilistic EWMA?



EWMA after spikes overestimates subsequent values

EWMA slow to acknowledge spikes

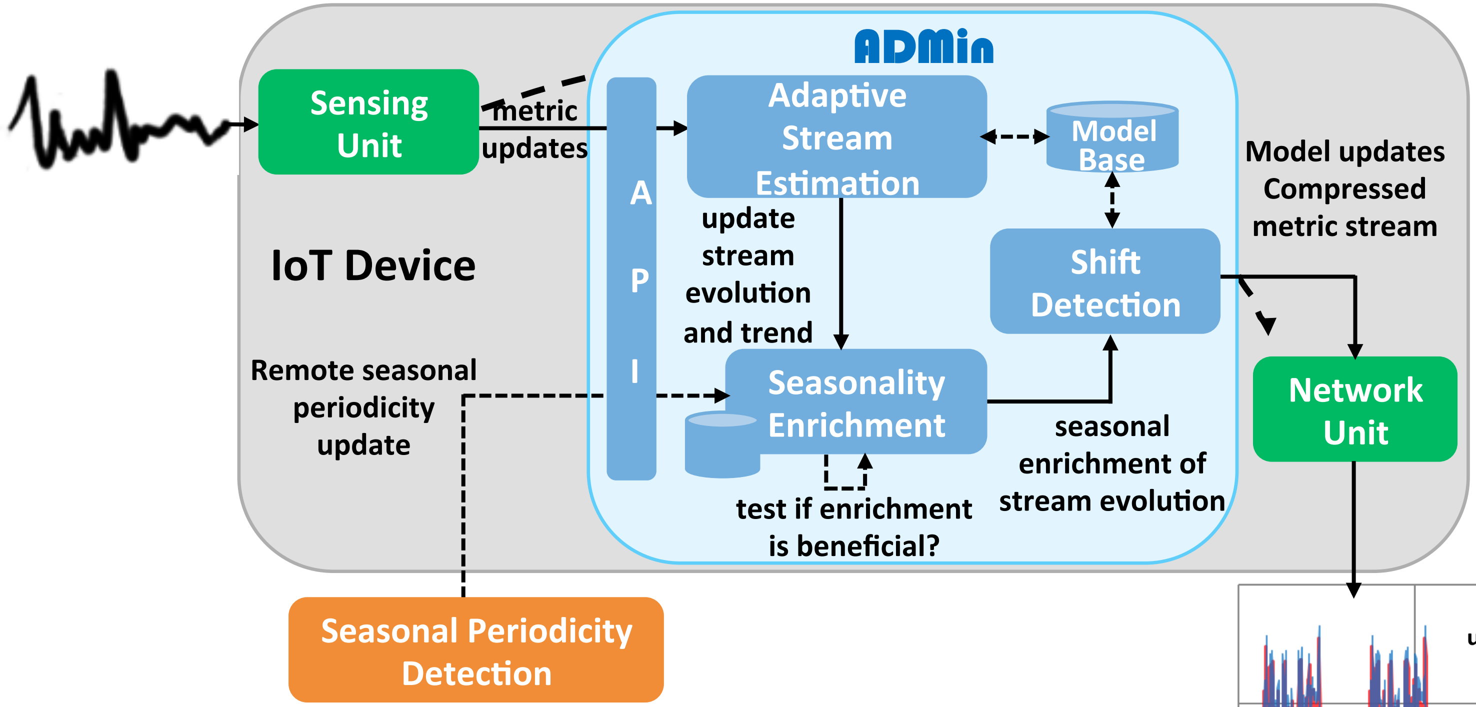
With **probabilistic reasoning** each datapoint will contribute to the estimation process depending on it's p-value

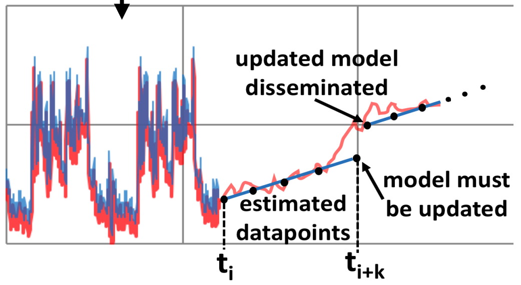
Comparison Towards State-of-the-Art Frameworks

- **LANCE** [Werner et al., ACM SenSys 2011] disseminates **summaries of windowed data** (weighted avg) with receiver deciding if data is useful when **summary violates a user-defined policy** (e.g., user given confidence intervals)
- **G-SIP** [Gaura et al., IEEE Trans. on Sensors 2013] disseminates updates only when **datapoint value rate of change cannot be predicted** from previous value knowledge (EWMA) within given user-defined accuracy guarantees
- **ADWIN** [Bifet et al., SIAM 2010] uses a linear **Naive Bayes predictor** as its estimation model along with two sliding windows to **detect shifts in model** based on user given confidence intervals

All three framework parameters configured to output best results

The ADMin Framework


ADMin
Sensing Unit
metric updates
API
Adaptive Stream Estimation
Model Base
update stream evolution and trend
Shift Detection
Model updates Compressed metric stream
IoT Device
Remote seasonal periodicity update
Seasonality Enrichment
test if enrichment is beneficial?
seasonal enrichment of stream evolution
Network Unit
Seasonal Periodicity Detection



ADMin - disseminate model updates not datapoints…

- **Adapts dissemination rate** of IoT device based on monitoring stream variability
- Reduces on device **energy consumption** and **volume** and **velocity** of data generated in streaming networks