Keep it Simple, Sustainable!
When Is ML Necessary in Cloud Resource Management?

Thaleia Dimitra Doudali

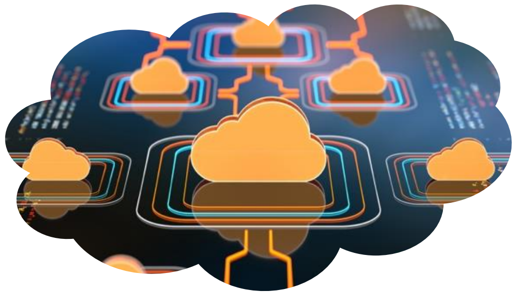
Assistant Professor

IMDEA Software Institute, Madrid, Spain

Sunday, March 30th 2025

GreenSys workshop at EuroSys/ASPLOS 2025
Rotterdam, Netherlands

Current Challenges in Cloud Computing



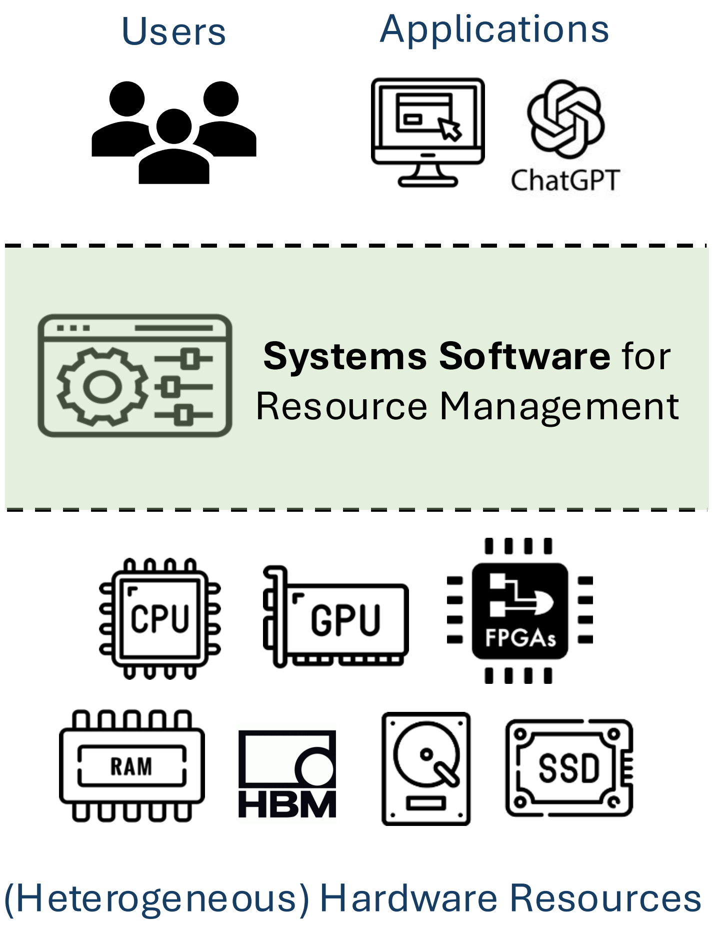
1. Cloud servers are severly under-utilized ~20%.

- **Cloud providers** over-provision resources to meet peak demand.
- **Users** over-estimate their resource needs.

2. Datacenters produce massive amounts of CO2.

- **Globally** datacenters emit millions of metric tons of CO2, which is equivalent to millions of long-haul flights.
- **Idle servers** still consume energy! Energy waste.
- **AI significantly contributes.** Training GPT-3 emits 284 tons of CO2.

System-level Resource Management



Resource Management Systems are responsible for:

Allocate resources to *any* users and applications.

Monitor resource usage, analyze data access patterns.

Dynamically and proactively adjust resource allocations,

dynamically move data across memory/storage,

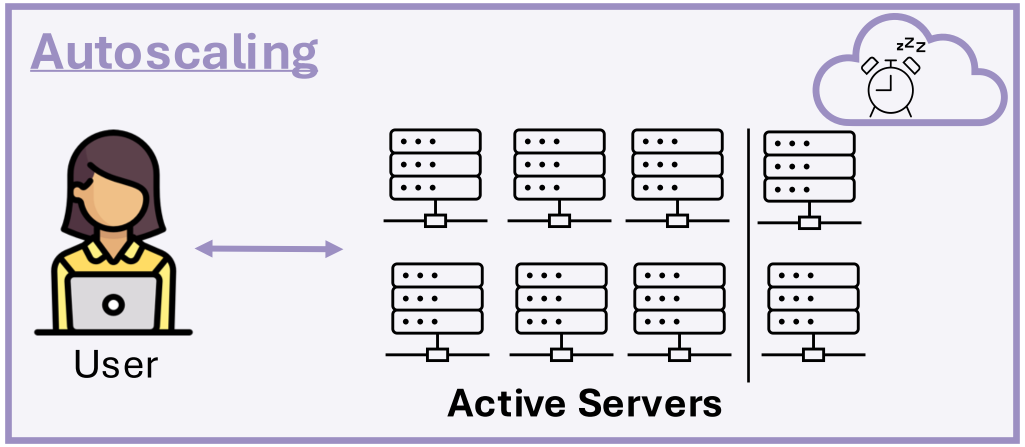
to improve **application performance** and **resource efficiency.**

Speed $ Cost Savings   Efficiency

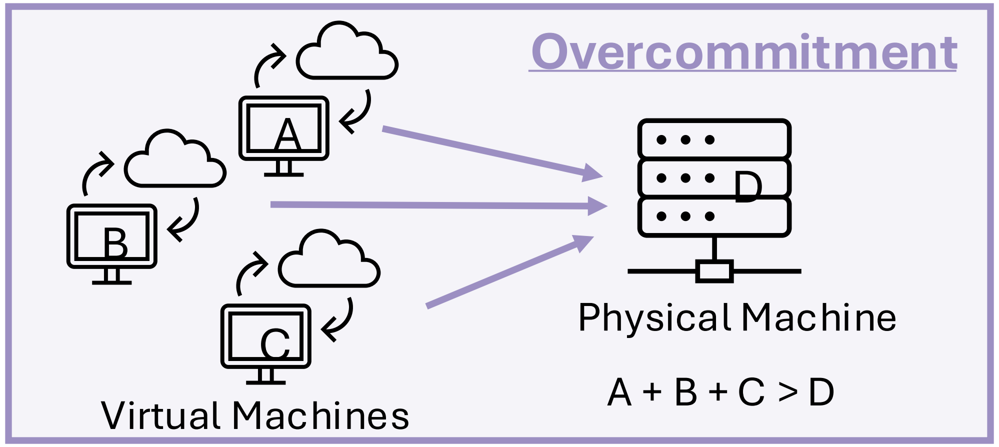
Cloud Resource Management Techniques

The following techniques help increase resource utilization and efficiency.

Basic idea: don't give to the user what they ask for, only what they actually use.

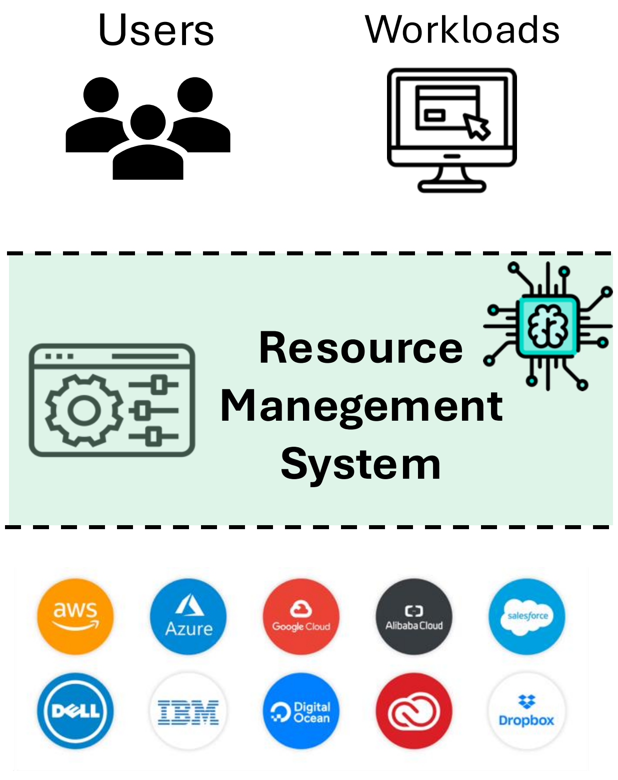


Dynamically **scale up or down** the number of computational resources e.g., active servers, number of CPUs.

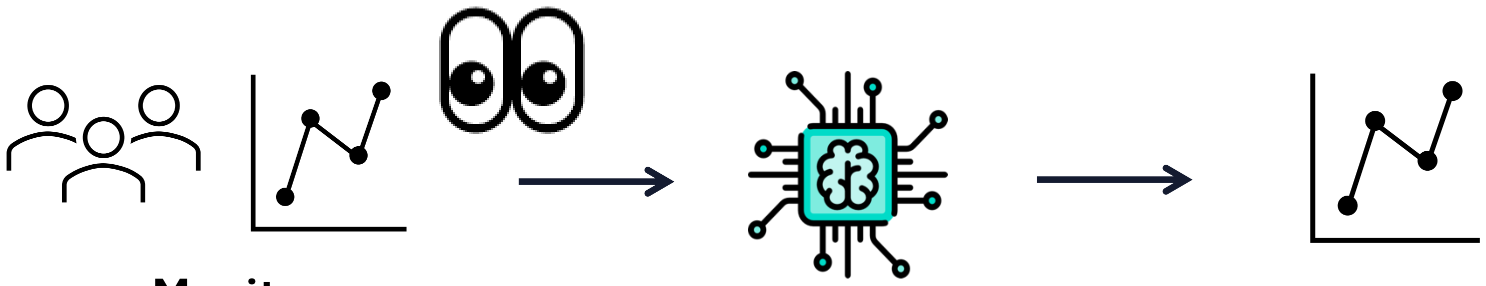


Allocate **more virtualized** resources than the ones physically available.

The Key: Resource Usage Forecasting



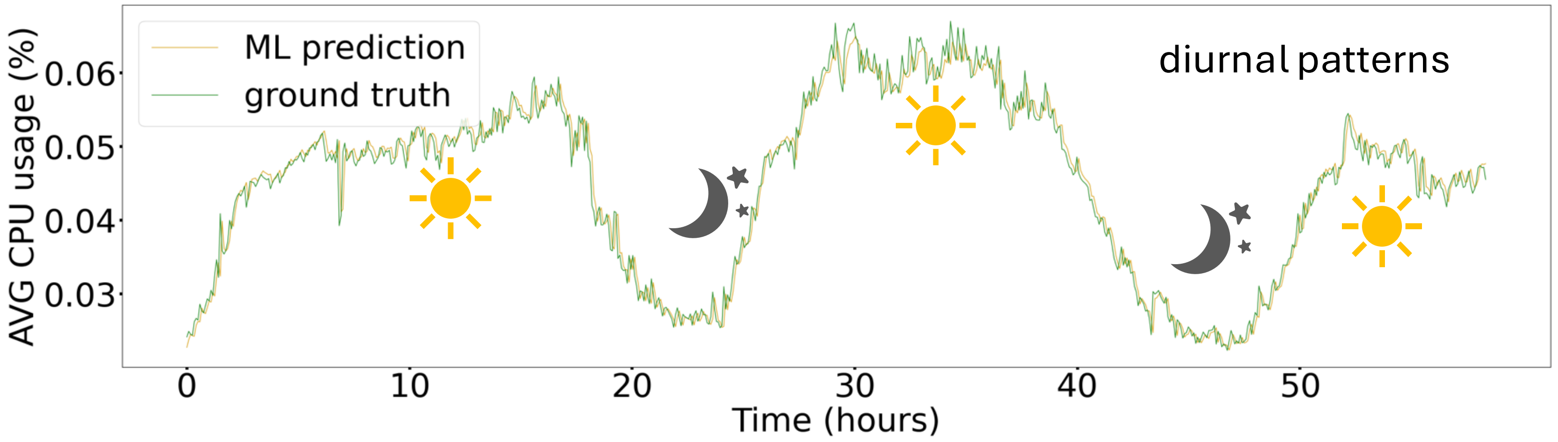
Hardware in the "Cloud"



Monitor
User Behavior
Workload Patterns

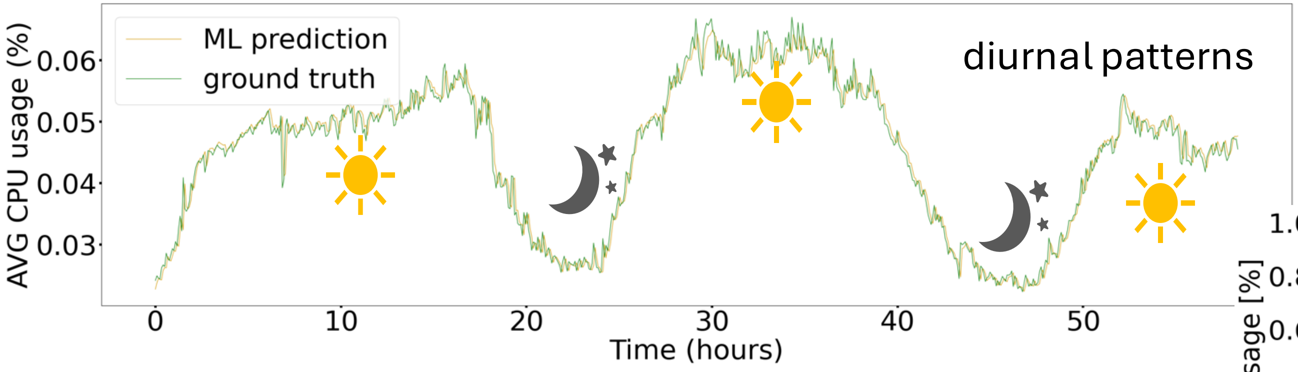
Predictor

Future
resource usage,
demand, load.

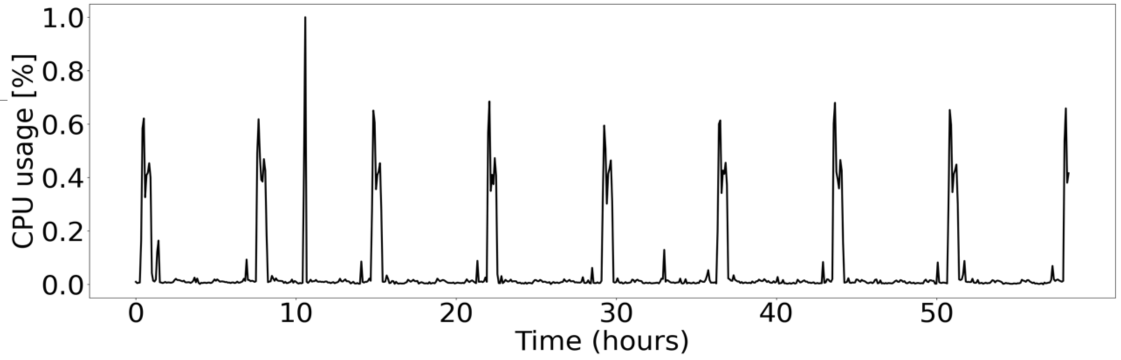


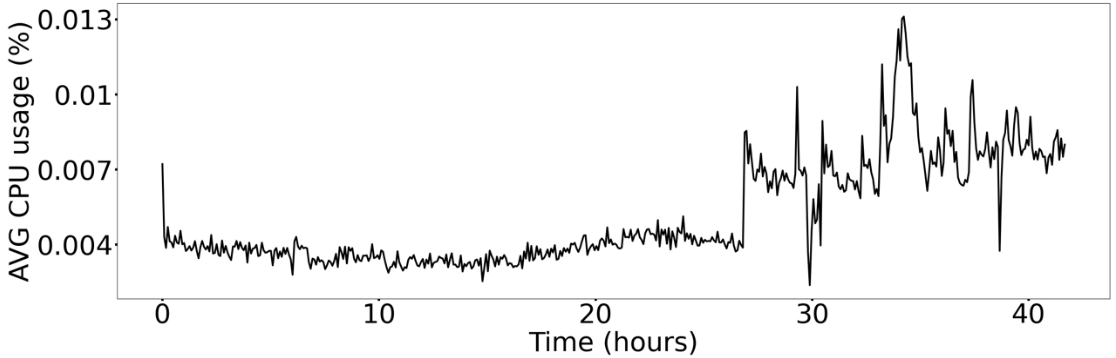
Accurate Predictiors → Timely and Effective Resource Management → Resource Efficiency.

Accurate Resource Usage Forecasting is Challenging

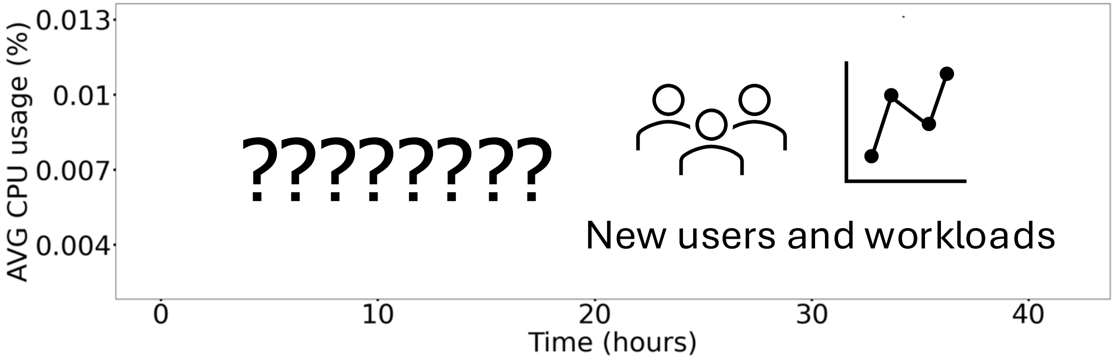


Stable, periodic, diurnal patterns are **predictable**.





Sudden changes, spikes, high dynamicity, are **hard to predict.**



Unseen patterns could be completely **unpredictable.**

Using Machine Learning in Resource Management

Advantages

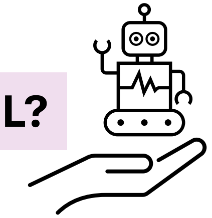
- Learn complex patterns.
- High accuracy.
- Use as a black-box.
- Transfer and continuous learning.

Challenges

- High overheads (time, storage).
- Engineering effort for production-level use.
- Interpretability concerns.
- Sustainability concerns.

This talk: **(When)** Is Machine Learning **Necessary** to Use in System-level Cloud Resource Management?

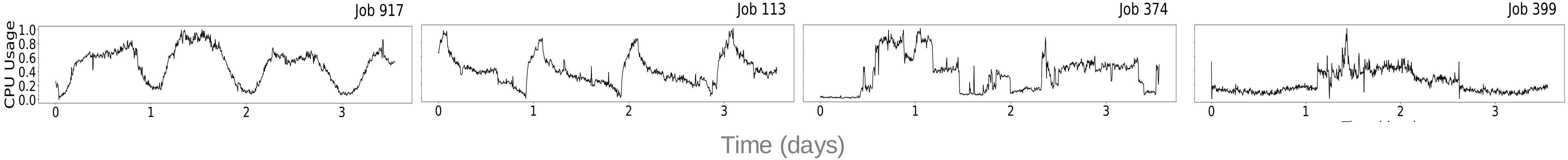
To ML or not to ML?



Effectiveness of *ML models* in Cloud Resource Usage *Prediction*

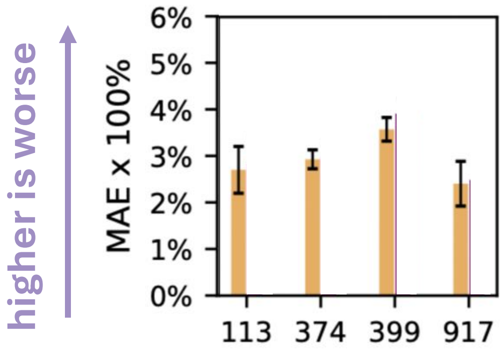
Systematic Experimentation with LSTMs

1 job → Many similar tasks



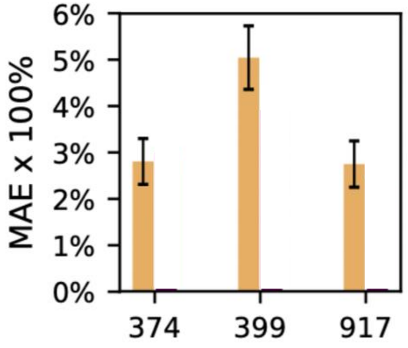
Trained 1 LSTM model per job.

Models tested across different tasks of the same job.



LSTMs generalize across **“similar”** patterns

Model 113 tested across jobs 374, 399, 917.

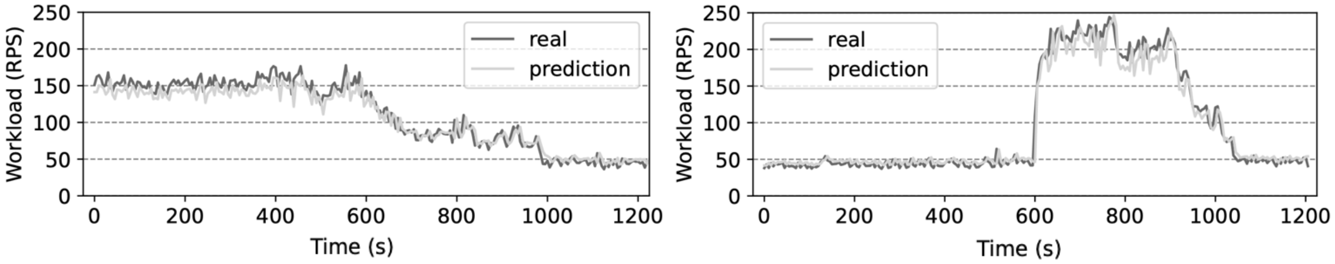


LSTMs generalize across **“unseen”** patterns

[SoCC ‘23] *Is Machine Learning Necessary for Cloud Resource Usage Forecasting?*
Georgia Christofidi , Konstantinos Papaioannou, Thaleia Dimitra Doudali.

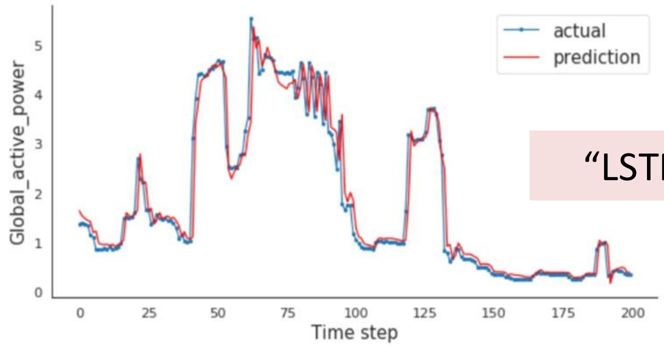
LSTMs Are Great! And Others Agree!

Use Case: ML inference Serving



[EuroMLSys '23] "Reconciling High Accuracy, Cost-Efficiency, and Low Latency of Inference Serving Systems" by Salmani et al.

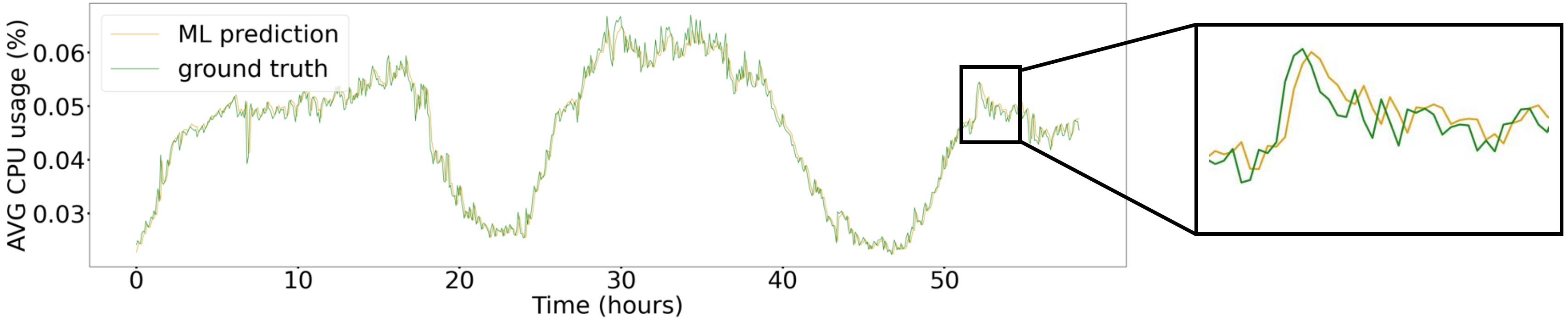
Use Case: Predict Power Consumption



"LSTMs are amazing!"

[medium.com]"Time Series Analysis, Visualization & Forecasting with LSTM" by Susan Li

Our Analysis:

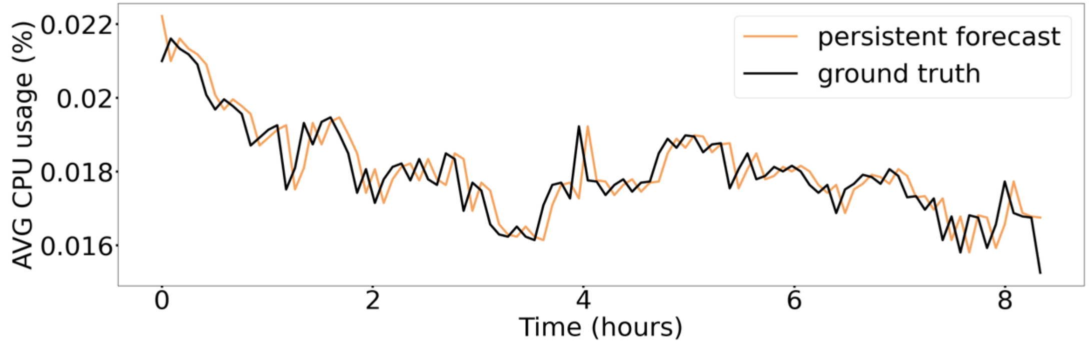


Insight: LSTM predictions look like **"shifted" versions** of the real (ground truth) data. They do not *truly* learn!
High accuracy was misleading!

[SoCC '23] *Is Machine Learning Necessary for Cloud Resource Usage Forecasting?* Georgia Christofidi , Konstantinos Papaioannou, Thaleia Dimitra Doudali.

A Simple and Practical non-ML Predictor

Idea: Predict a shifted version of the ground truth, similar to the LSTMs.



Persistent Forecast*

Predicted Value(t) =

Ground Truth(t – 1)

Can something so naïve actually work?

Extensive experimentation with public open-source datasets across different:

Cloud providers

Alibaba Cloud
Google Cloud
Microsoft Azure
bitbrains

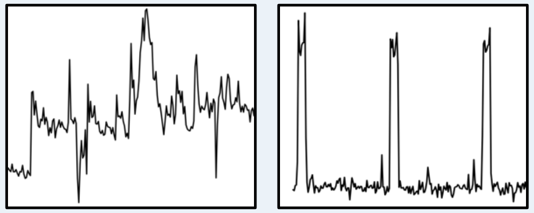
Resource Types

CPU
RAM

Resource Levels

Physical Machine
Virtual Machine
Workload

Usage Patterns

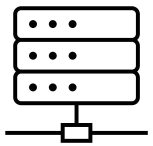


Frequency

Hourly/ Daily/ Weekly Windows

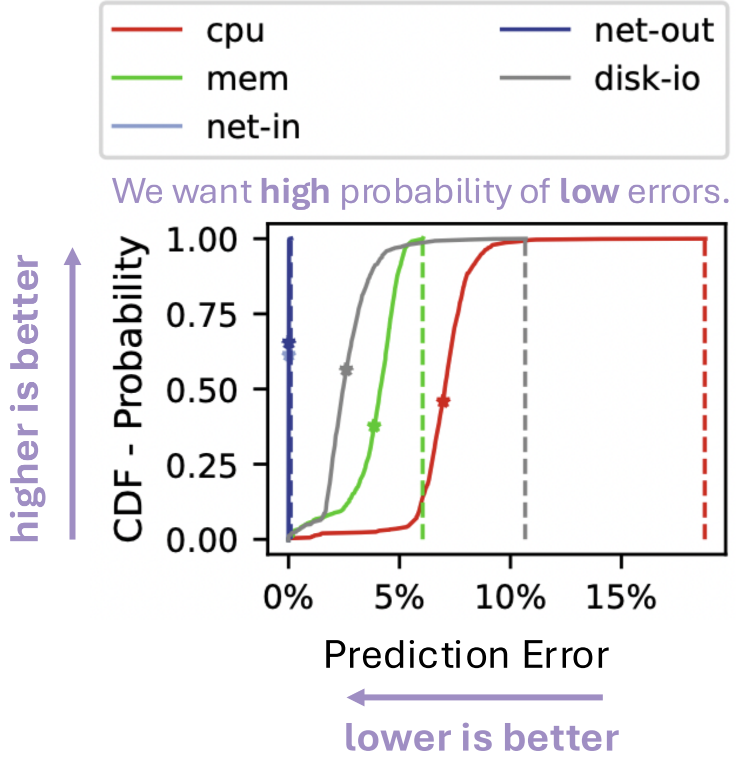
[SoCC ‘23] *Is Machine Learning Necessary for Cloud Resource Usage Forecasting?*
Georgia Christofidi , Konstantinos Papaioannou, Thaleia Dimitra Doudali.

Results – Physical Machines



Physical Machine

Alibaba Dataset
Physical Machine Level



Observations:

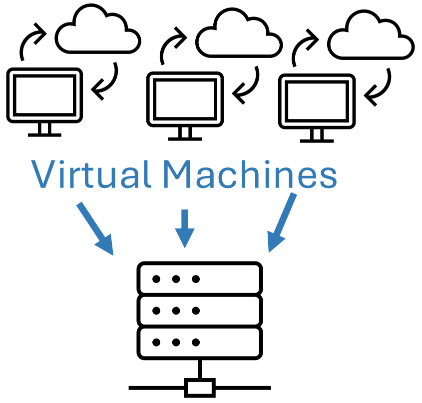
- net-in, net-out: Error ~ 0
- disk-io, mem: Avg. Error < 4 %. Short tail.
- cpu: Avg. Error ~ 7 %. Longer tail.

Physical Machines
Have **stable** load.

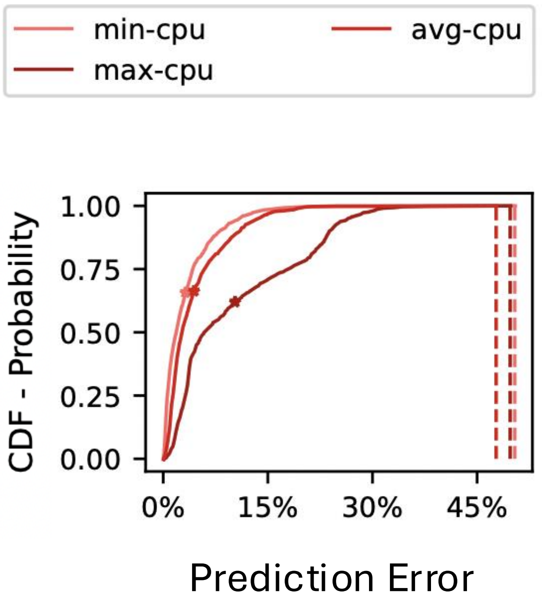
Persistent Forecast is **very** accurate!

[SoCC '23] *Is Machine Learning Necessary for Cloud Resource Usage Forecasting?*
Georgia Christofidi , Konstantinos Papaioannou, Thaleia Dimitra Doudali.

Results – Virtual Machine

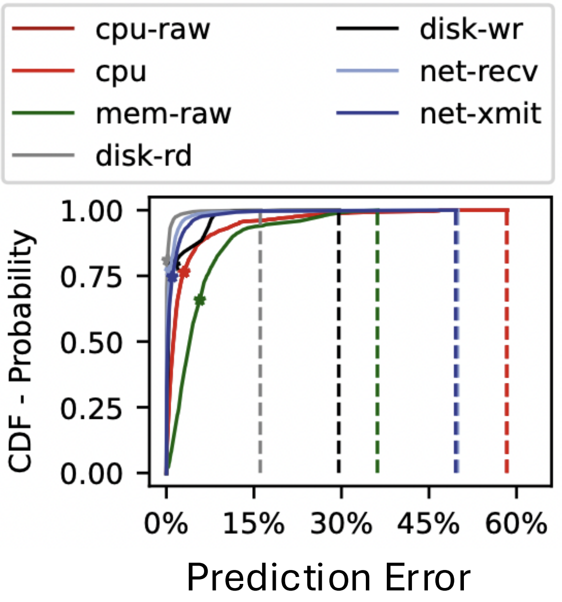


Azure Dataset
Virtual Machine Level

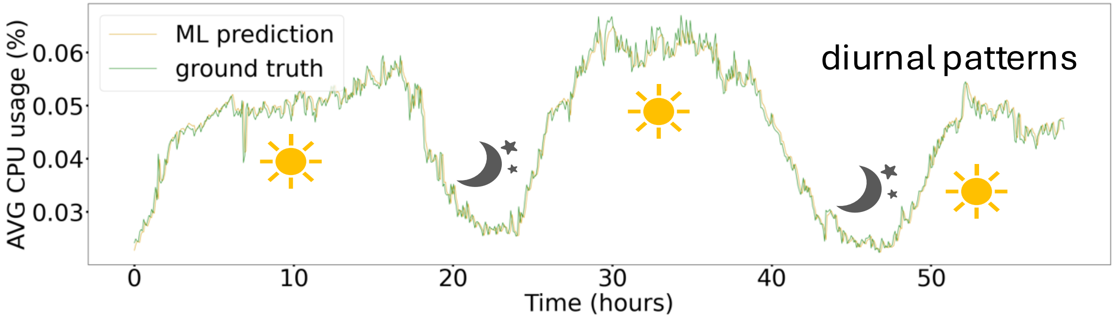


Average Prediction Error < 10%

Bitbrains Dataset
Virtual Machine Level



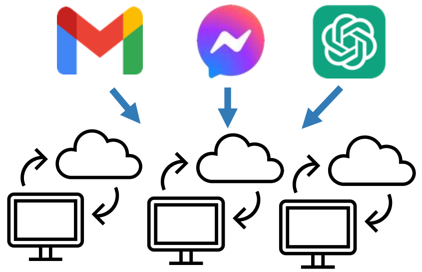
Average Prediction Error < 6 %



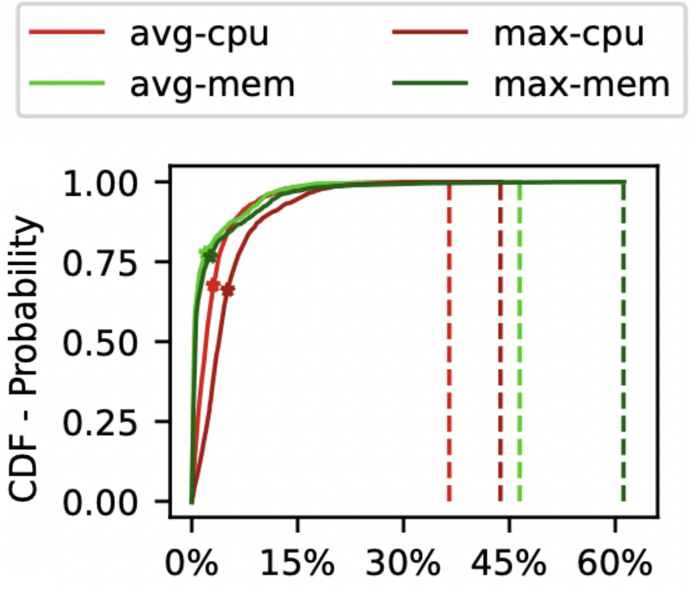
Virtual Machines
On average, **stable and periodic** load.
Patterns start becoming more dynamic.
(longer tails in the error)

[SoCC '23] *Is Machine Learning Necessary for Cloud Resource Usage Forecasting?*
Georgia Christofidi , Konstantinos Papaioannou, Thaleia Dimitra Doudali.

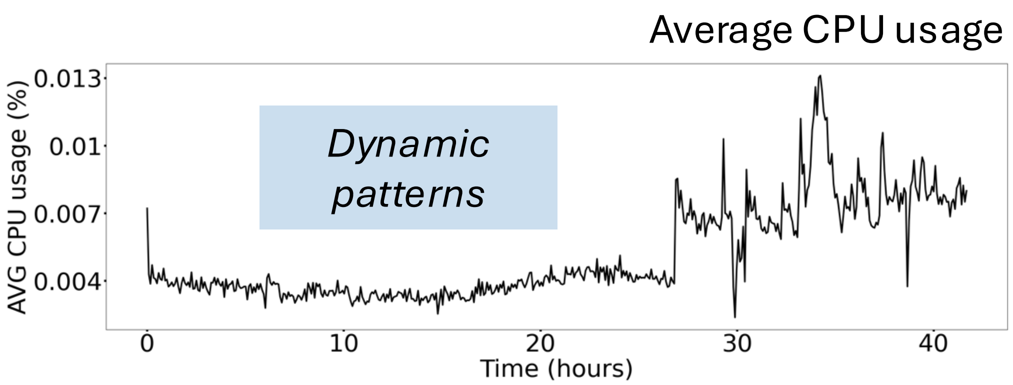
Results – Applications

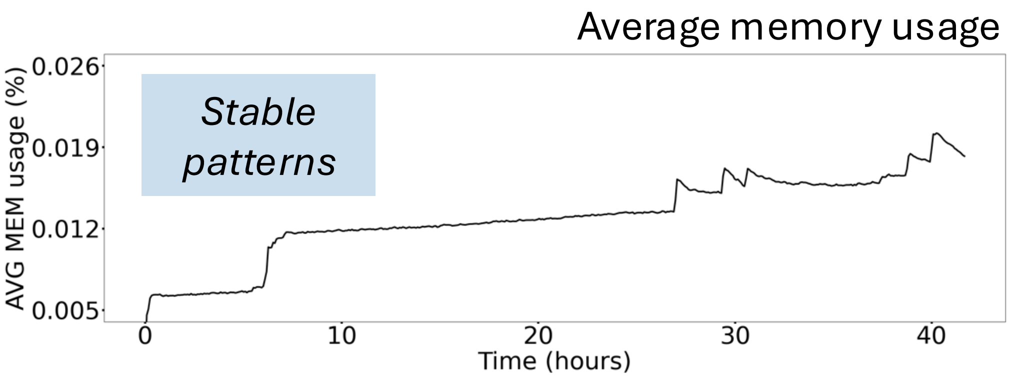


Google Dataset
Workload Level



Average Prediction Error < 6%



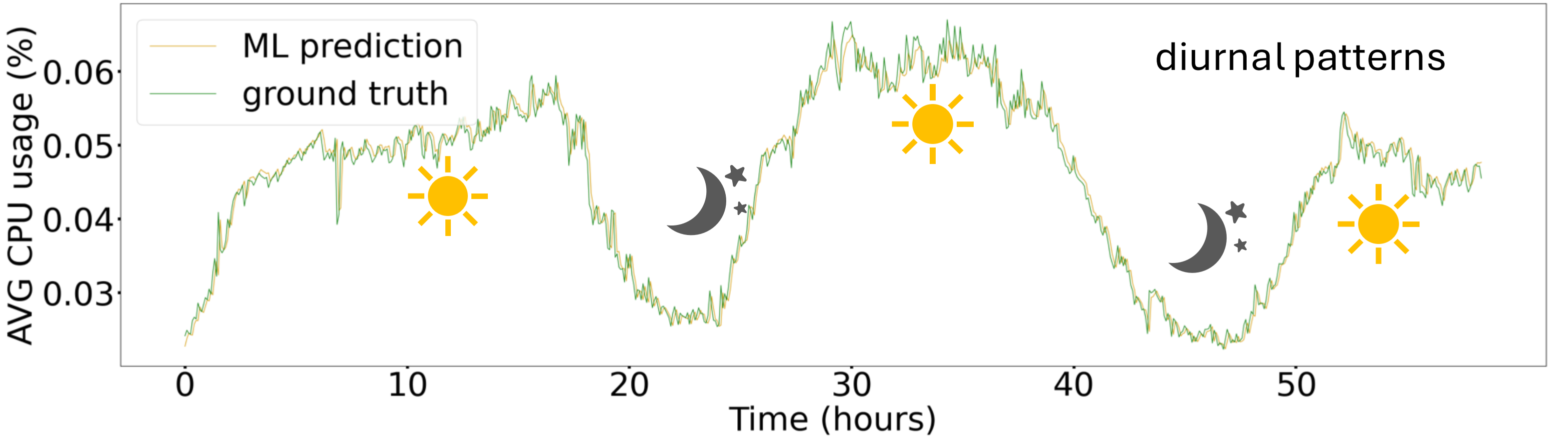


Applications
Most **dynamic** patterns.
(longest tails in the error)
Depends on the
type of resource!

[SoCC '23] *Is Machine Learning Necessary for Cloud Resource Usage Forecasting?*
Georgia Christofidi , Konstantinos Papaioannou, Thaleia Dimitra Doudali.

Why the Persistent Forecast Works?

Overall, on average, the persistent forecast is **very accurate**, prediction error < 6%. **Why?**

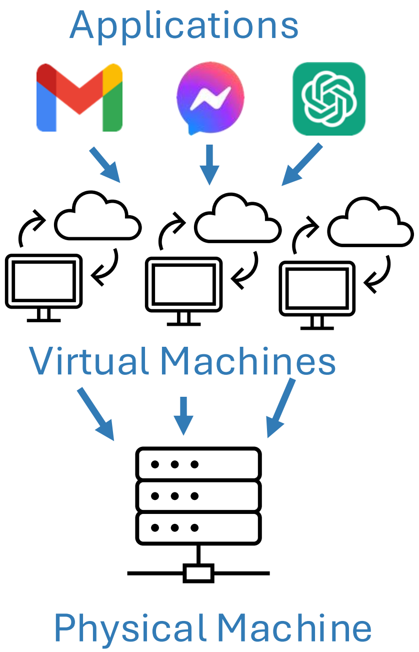


Because cloud resource usage is **highly persistent over time**, it changes very little every e.g. 5 minutes.

[SoCC '23] *Is Machine Learning Necessary for Cloud Resource Usage Forecasting?*
Georgia Christofidi , Konstantinos Papaioannou, Thaleia Dimitra Doudali.

Characterization Summary

To ML or not to ML?



Level	Pattern	Persistence
Application	Dynamic	**Low**
Virtual Machine	Periodic	Medium
Physical Machine	Stable	High

← Predict with **ML**

Predict with **non-ML**
it will be highly accurate!

Resource Type	Pattern	Persistence
CPU	Dynamic	**Low**
Memory	Stable	High
Disk	Stable	High
Network	Stable	High

← Predict with **ML**

Predict with **non-ML**
it will be highly accurate!

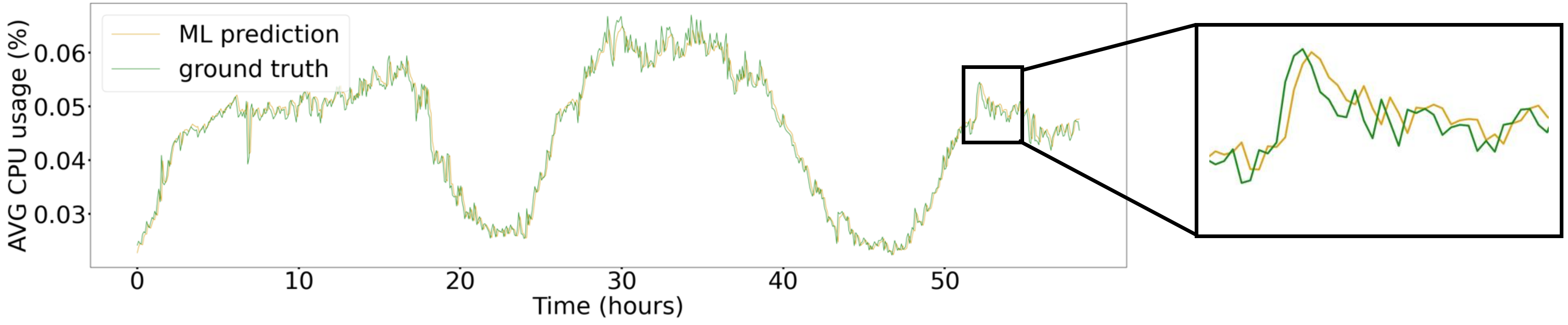
Data-driven choice!

[SoCC '23] *Is Machine Learning Necessary for Cloud Resource Usage Forecasting?*
Georgia Christofidi , Konstantinos Papaioannou, Thaleia Dimitra Doudali.

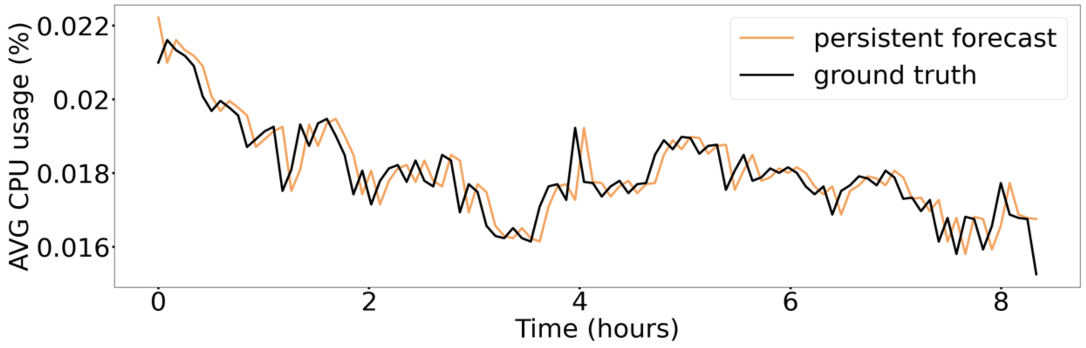
Lessons Learned

When using ML models to learn data across time

Lesson 1: Sometimes ML doesn't truly learn. Need for fine-tuning. Visualize to validate achieved accuracy.



Lesson 2: Not everything needs ML. Naïve, simple forecasts can accurately predict highly persistent data.



Effectiveness of *Predictor Models* in Cloud Resource *Management*

Non-ML Predictors for Resource Overcommitment

Existing Predictors
Future **U**sage =

1. Borg
90% * **Limit**

Google Cloud

2. Resource Central
sum of the 99-th%-ile

Microsoft Azure

3. N-Sigma
U+$N * std$(**U**)

Google Cloud

4. Take-it-to-the-limit (TITTL) = Max (1, 2, 3)

Why Max?

To eliminate potential *under*-estimations, which may cause:

- Degraded workload performance. 😔
- Unecessary resource auto-scaling. 😓
- User SLA violations. 😱

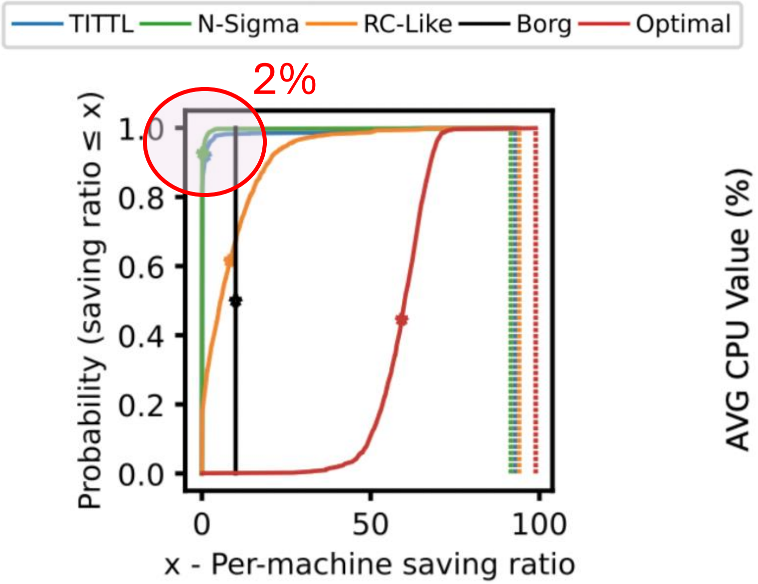
😃 **Simple, lightweight, explainable** and easy to engineer in production-level.

🤔 Do they **accurately predict** resource usage or just protect from under-estimations??

[EuroMLSys '24] *Do Predictors for Resource Overcommitment Even Predict?* Georgia Christofidi and Thaleia Dimitra Doudali.

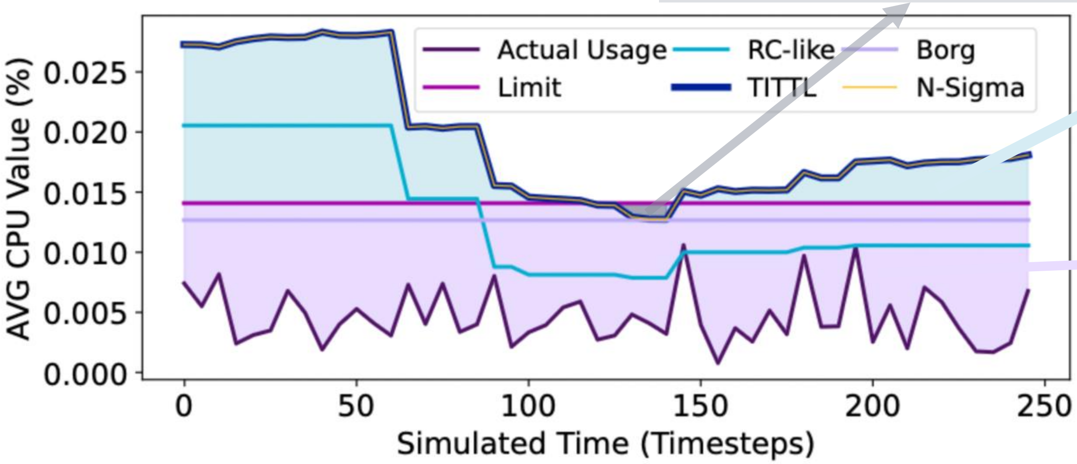
Current Predictors Allow for Low Resource Savings

Resource savings: Excess resources that can be reused/reallocated to other users / workloads.



lower is worse

Usable resources.
Resource savings **(2%)**

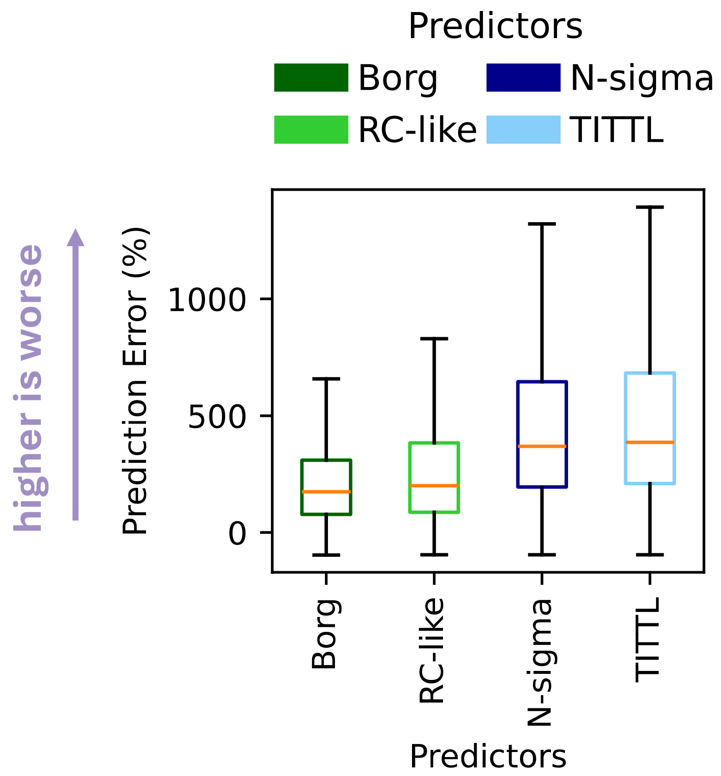


Completely wasted resources.

Maximum possible resource savings **(61%)** with an **optimal** predictor.

Savings and overcommitment are **possible** only when predictions are **lower** than the resource **limit.**

Do they Even Predict?



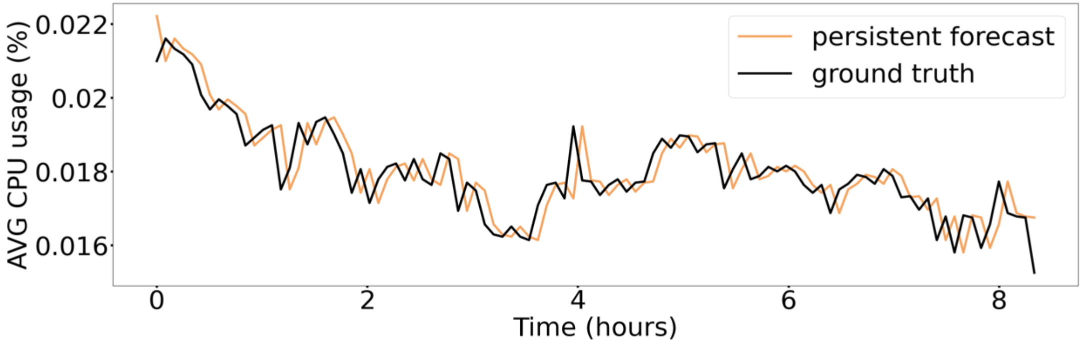
Prediction **error** is extremely **high**, especially for TITTL.

Predicted resource usage >> resource limit.
The system caps predictions to the value of the limit.

NO overcommitment is happening for **94%** of the cases we examined, due to the predictor's **OVER-estimations.**

Predictors just protect from under-estimations, ***allowing*** overcommitment **only 6%** of the times.

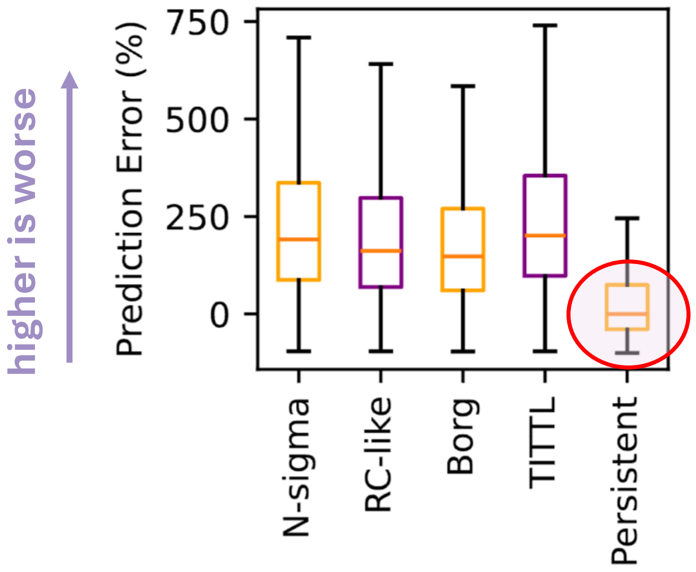
A Simple and Practical Predictor

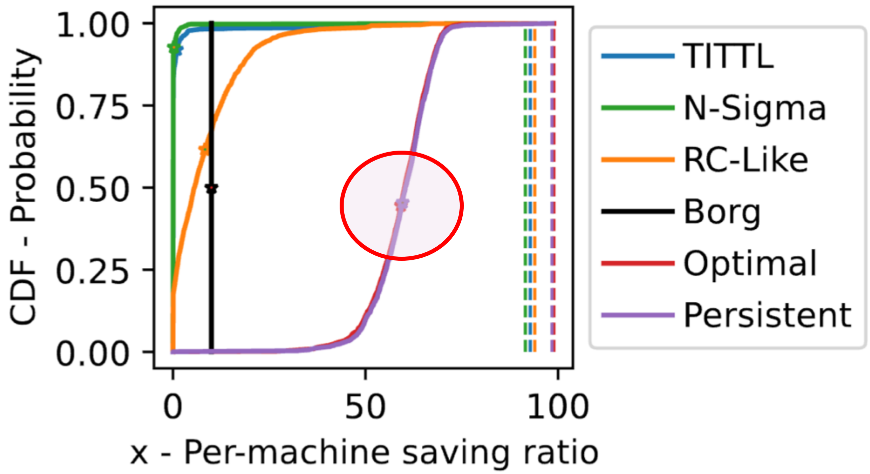


Persistent Forecast*

Predicted Value(t) =

Ground Truth(t – 1)



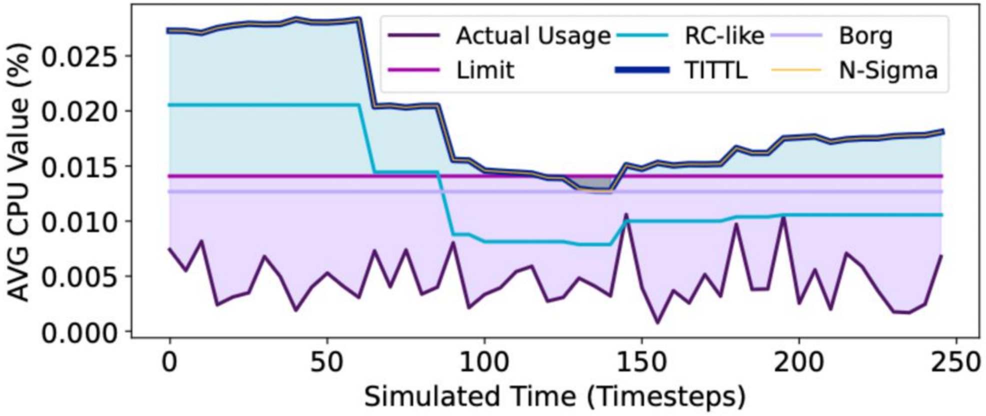


Takeaway:
Lower error enables high resource savings

Lessons Learned

When Integrating Prediction Models in Systems

Lesson 1: High Prediction error leads to predictions that **don’t make sense** > resource limit.

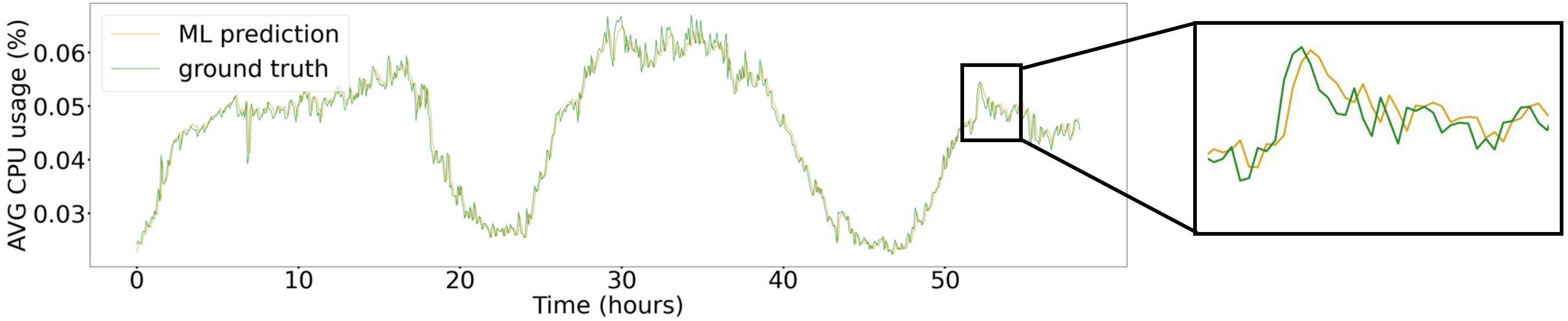


Lesson 2: High Prediction error allowed for very low resource savings and minimal overcommittment, **defeating the purpose for which it was used for.**

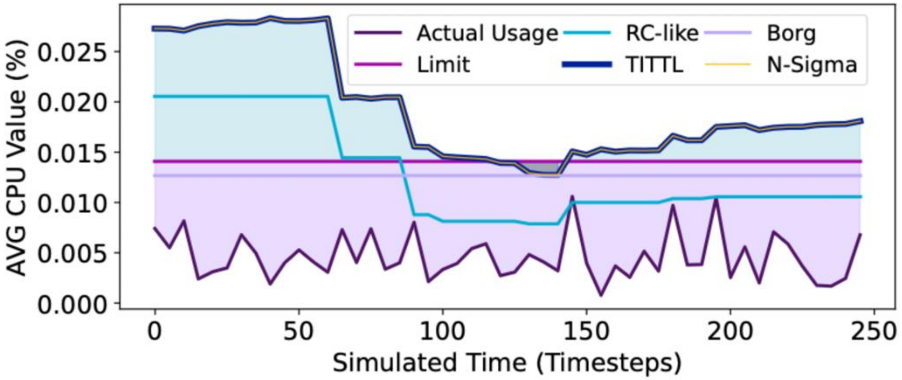
Recommendations on the Integration of ML in Systems

Recommendation 1- Visualize!

“1 image is worth 1000 words”.. Always visualize predictions!



Insight 1:
LSTMs just shift data, do not truly learn.

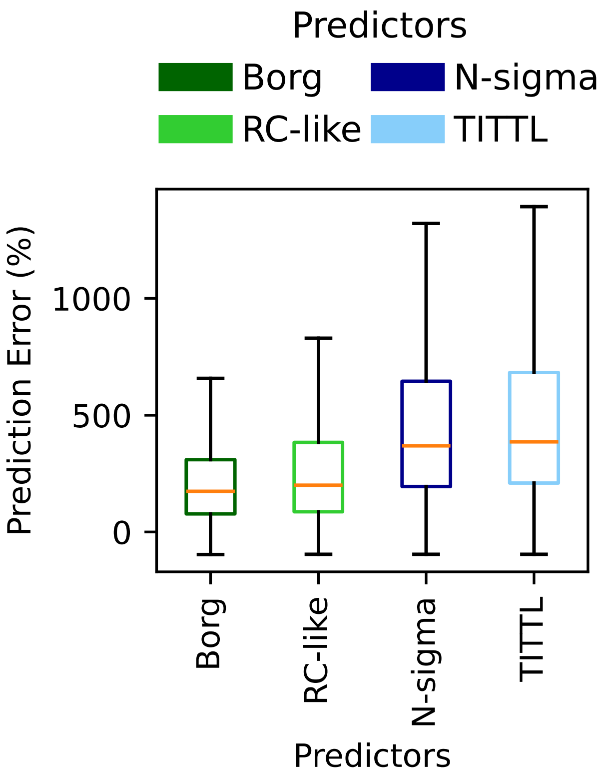


Insight 2:
Predictions were capped to resource limit.
Extreme over-predictions.

Recommendation 2 – Accuracy is not just a number

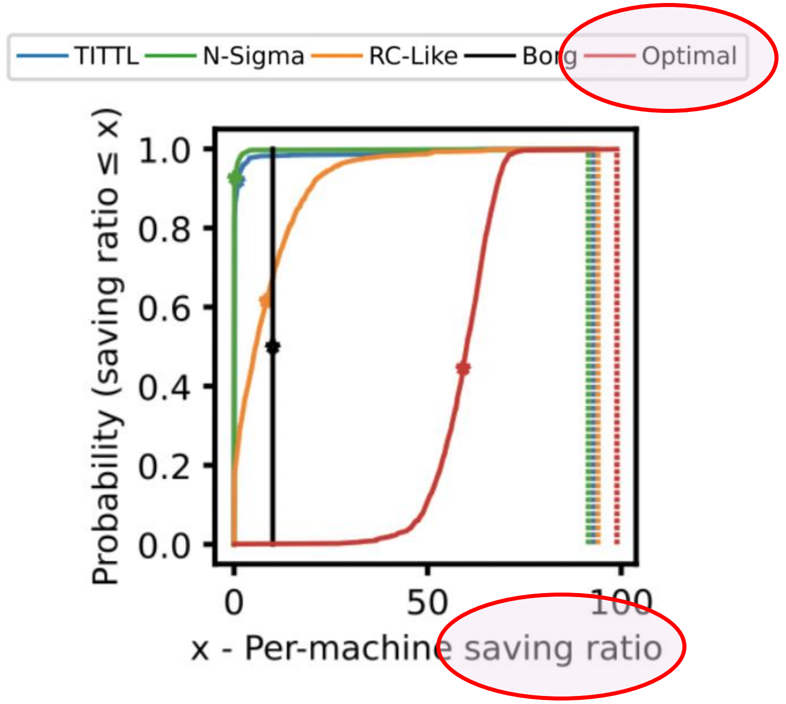
Check impact on **system-level metrics**, such as resource efficiency, application performance etc.

Compare with **optimal** system baseline, if 100% accurate predictions were possible



accuracy

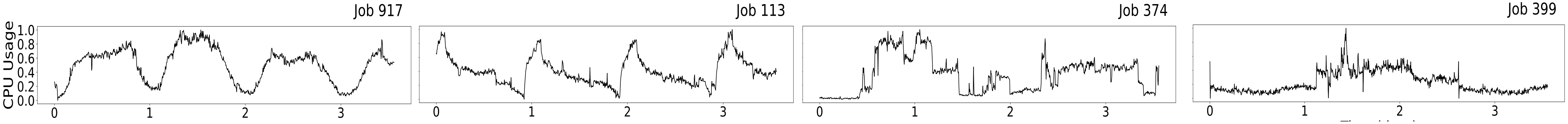
resource savings



Recommendation 3 – Choose Simplicity!

Can you get similar accuracy with non-ML methods, to allow for simplicity and no overheads?

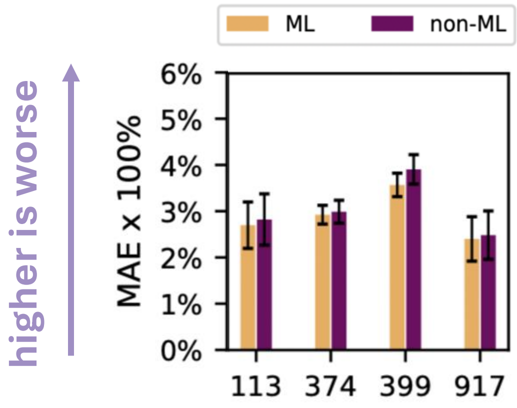
1 job → Many similar tasks



Time (days)

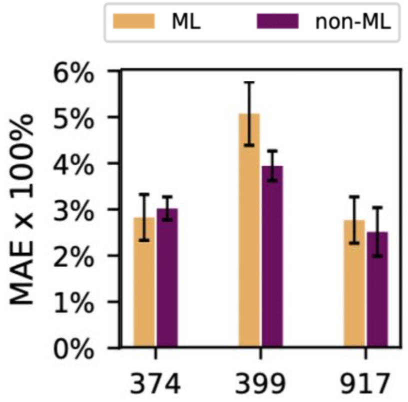
Trained 1 LSTM model per job.

Models tested across different tasks of the same job.



Similar accuracy for ML vs non-ML

Model 113 tested across jobs 374, 399, 917.



How to Integrate ML in System-level Resource Management?

Proposed Approach - K.I.S.S.

We build upon the KISS system design principle [US Navy 1960].

-- Simplicity should be a design goal!

Use ML only when and where necessary.

We propose **K.I.S.S = Keep it Simple, Smart / Sustainable!**

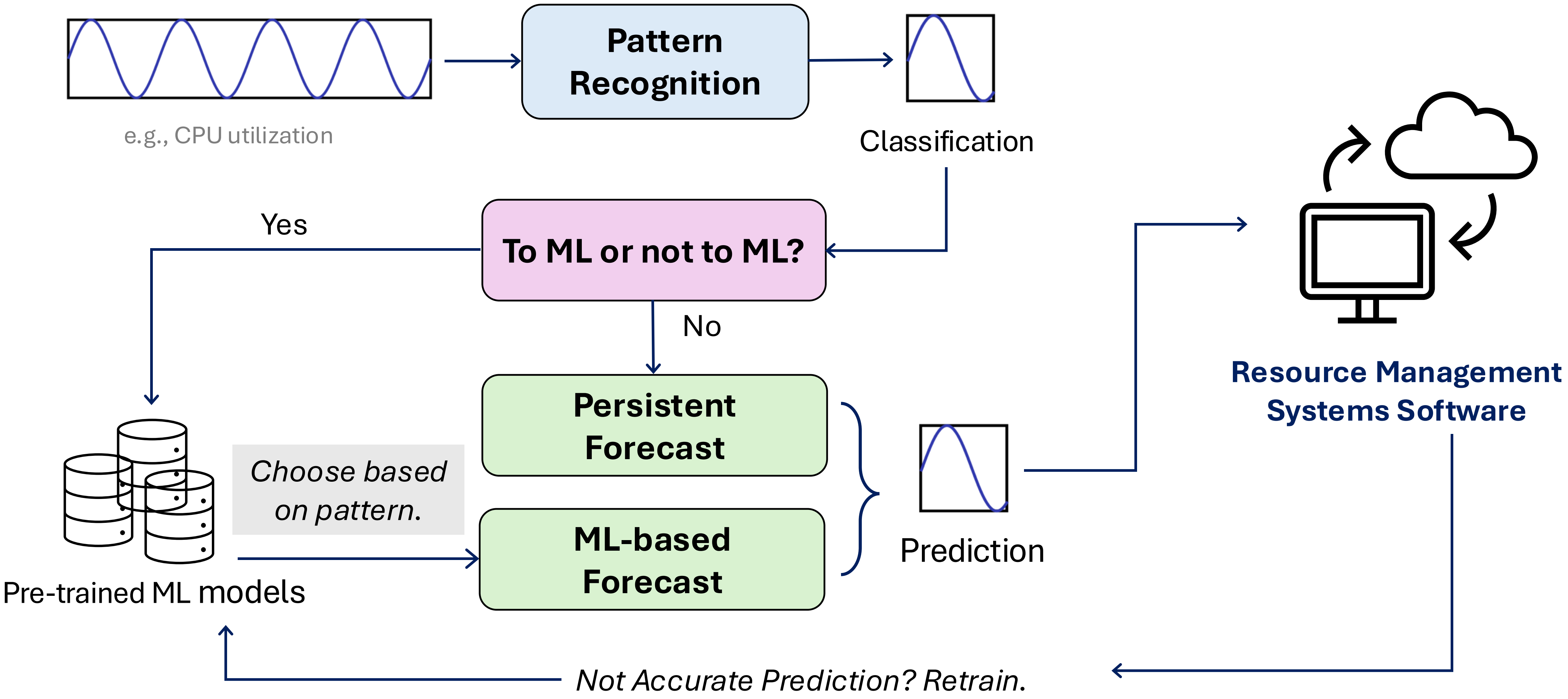
- **Smart:** (Clever) Use of ML
- **Sustainable:** Minimal use of ML

Goal: Maximize prediction accuracy and resource efficiency,

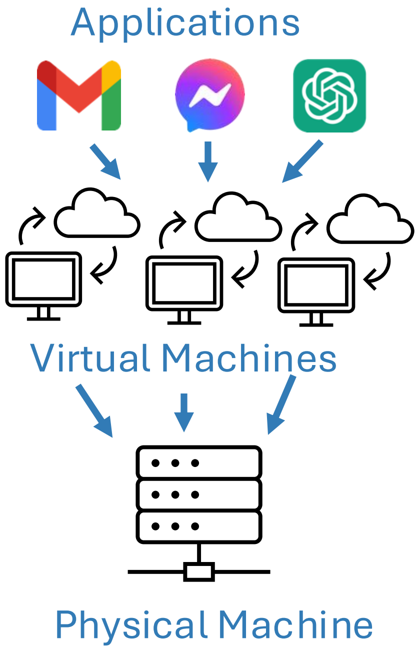
in return for minimal ML overheads and impact.

[Thaleia Dimitra Doudali] https://www.sigops.org/2023/k-i-s-s-keep-it-simple-smart/
[Thaleia Dimitra Doudali] https://www.sigarch.org/think-twice-before-using-machine-learning-to-manage-cloud-resources/

Proposed System Design for ML Integration



Is Machine Learning Necessary?



No!!

At least not always..

When is ML necessary?

Level	Pattern	ML?
Application	Dynamic	**Yes**
Virtual Machine	Periodic	No
Physical Machine	Stable	No

Type	Pattern	ML?
CPU	Dynamic	**Yes**
Memory	Stable	No
Disk	Stable	No
Network	Stable	No

Website

We propose **K.I.S.S = Keep it Simple, Smart / Sustainable!**

Thank you!