

Is Machine Learning Necessary to Use in Cloud Resource Management?

Thaleia Dimitra Doudali

Assistant Professor

IMDEA Software Institute, Madrid, Spain

Monday, 22 April 2024

Talk at the ESwML workshop of EuroSys 2024

Cloud Resource Efficiency



Data center resource utilization ~20%.

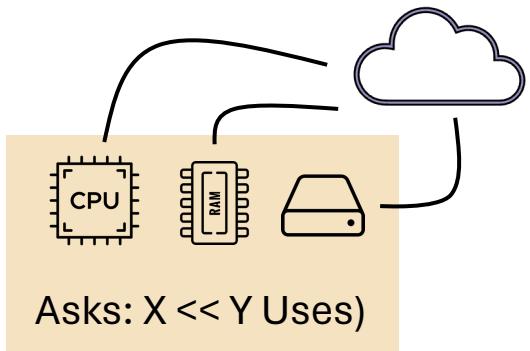


What causes low resource efficiency? 🤔

1



Users



Cloud Configuration

2



Cloud providers



SLAs

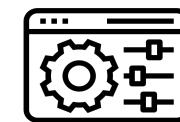


Provide Paid Resources

3



Management Systems



Suboptimal decisions

Cloud Resource Management Techniques

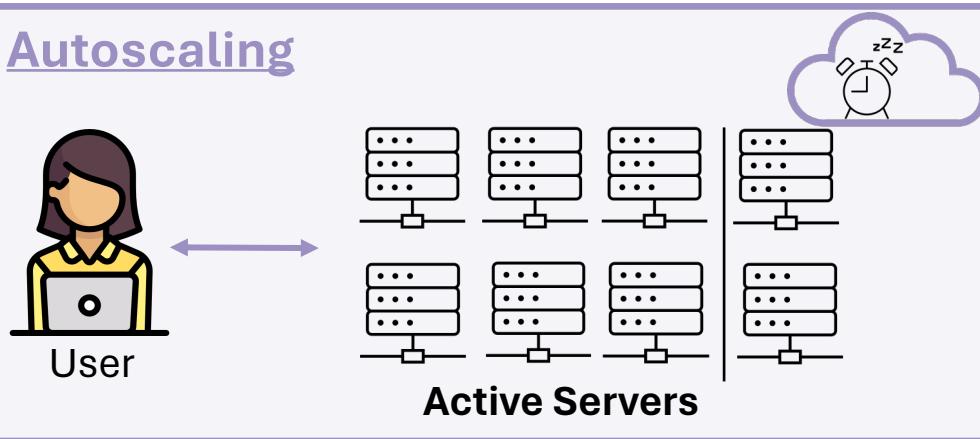


The following techniques help increase resource utilization and efficiency.



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

Autoscaling

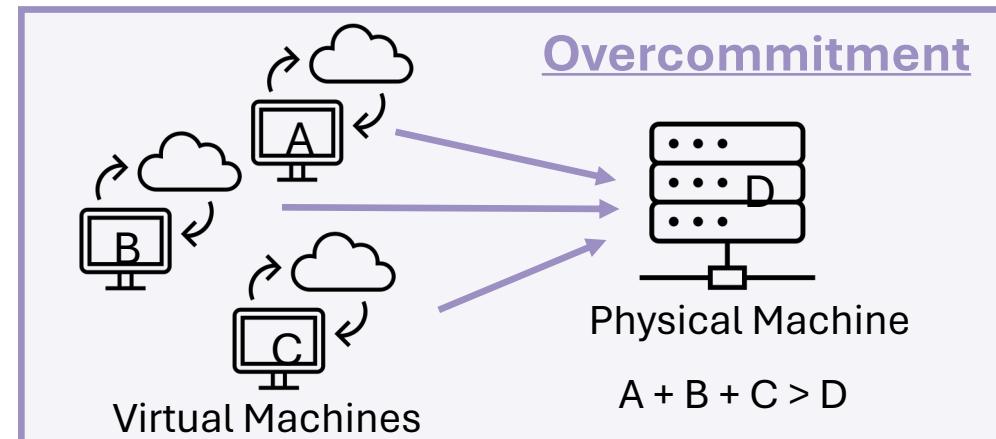


Dynamically **scale up or down** the number of computational resources
e.g., active servers, number of CPUs.



Delayed adjustment.
Performance reduction.

Overcommitment

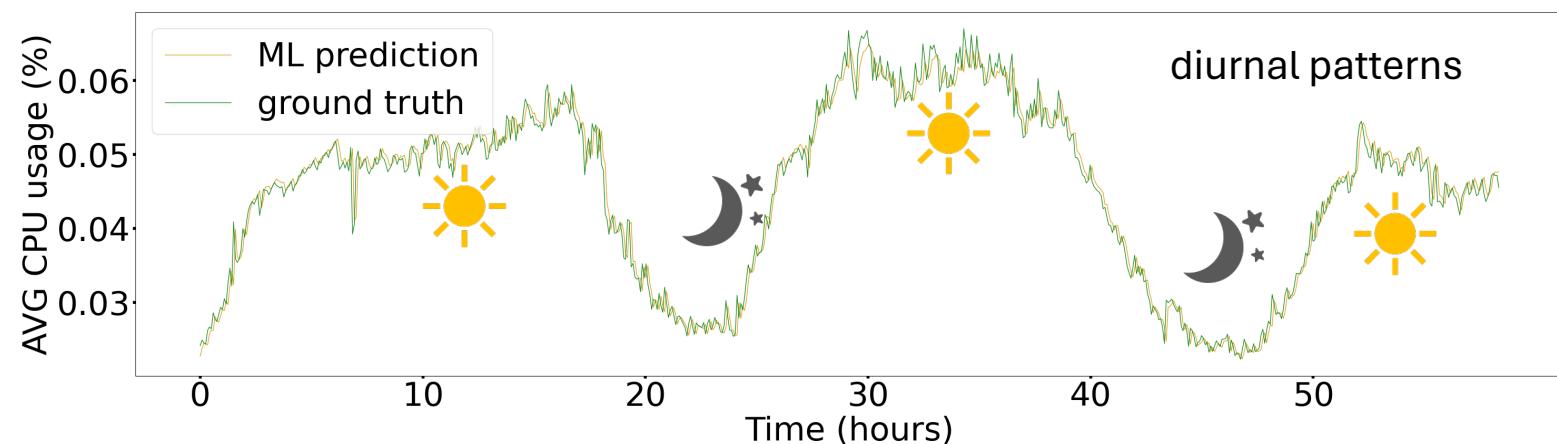
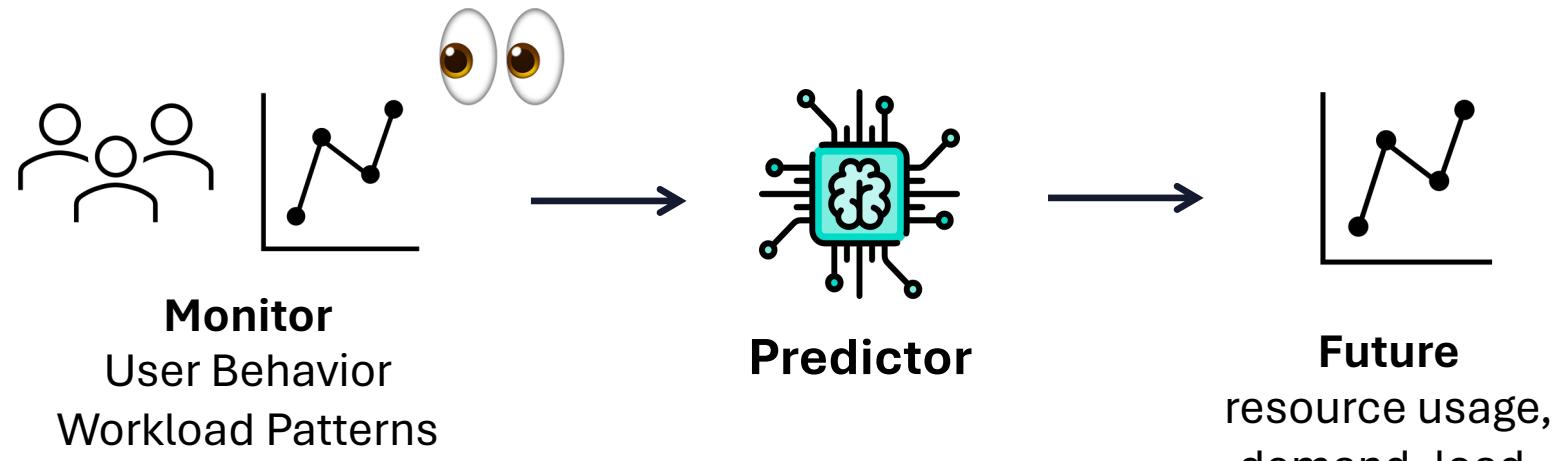


Allocate **more virtualized** resources than the ones physically available.
Assumes users *underutilize* resources.



Resource contention.
Performance reduction.

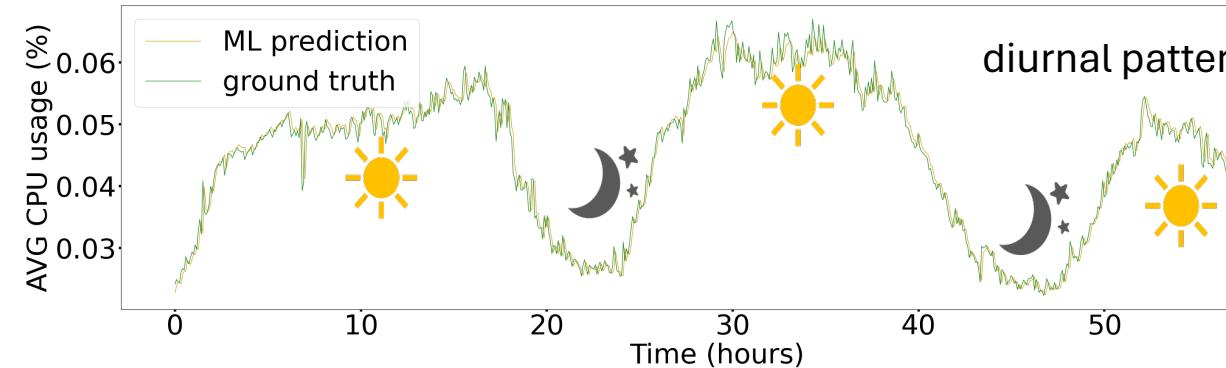
The Key: Resource Usage Forecasting



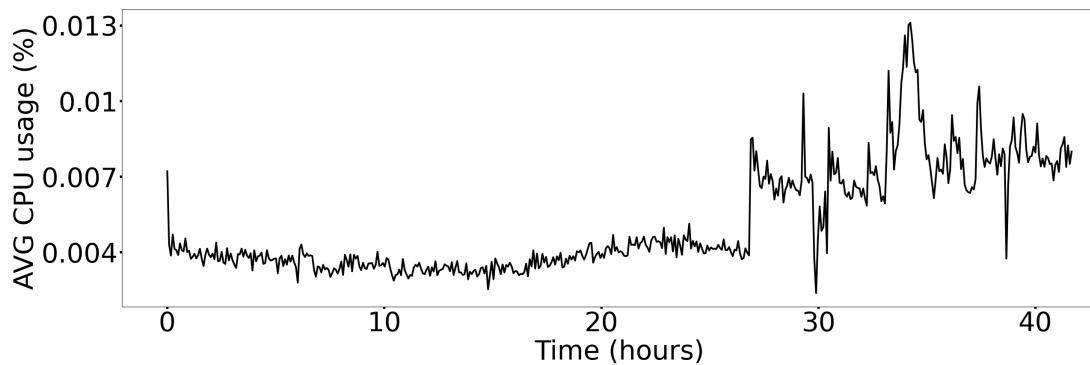
Accurate Predictions → Timely and Effective Resource Management → Resource Efficiency.



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**.

Predictors for Resource Overcommitment

Existing Predictors

Future Usage =

1. Borg
90% * Limit



Google Cloud

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



Microsoft Azure

3. N-Sigma
 $\mathbf{U} + N * std(\mathbf{U})$



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. 😔
- Unnecessary resource auto-scaling. 🌦️
- User SLA violations. 😱

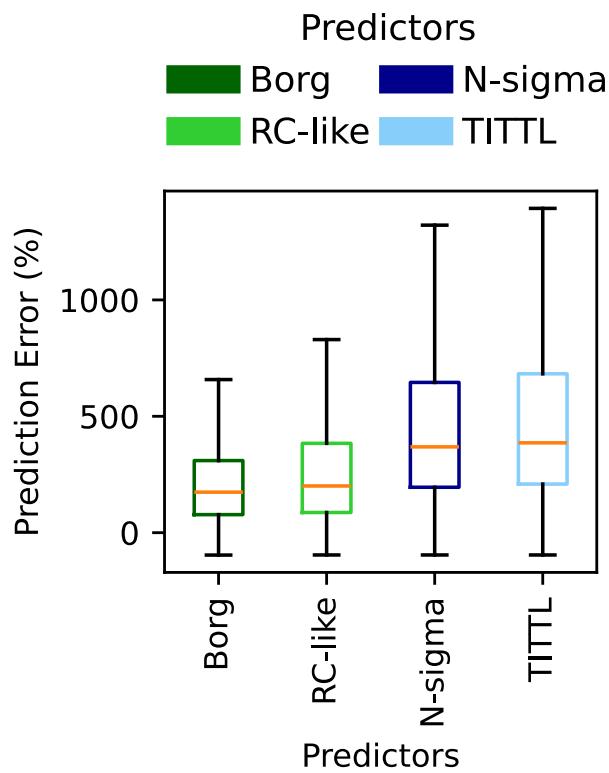


Simple, lightweight, explainable and easy to engineer in production-level.



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

Do they Even Predict?



Prediction error is extremely high, especially for TITTL.



Predicted resource usage >> resource limit.



The cloud provider will NEVER allocate to a user more resources than requested and paid for!

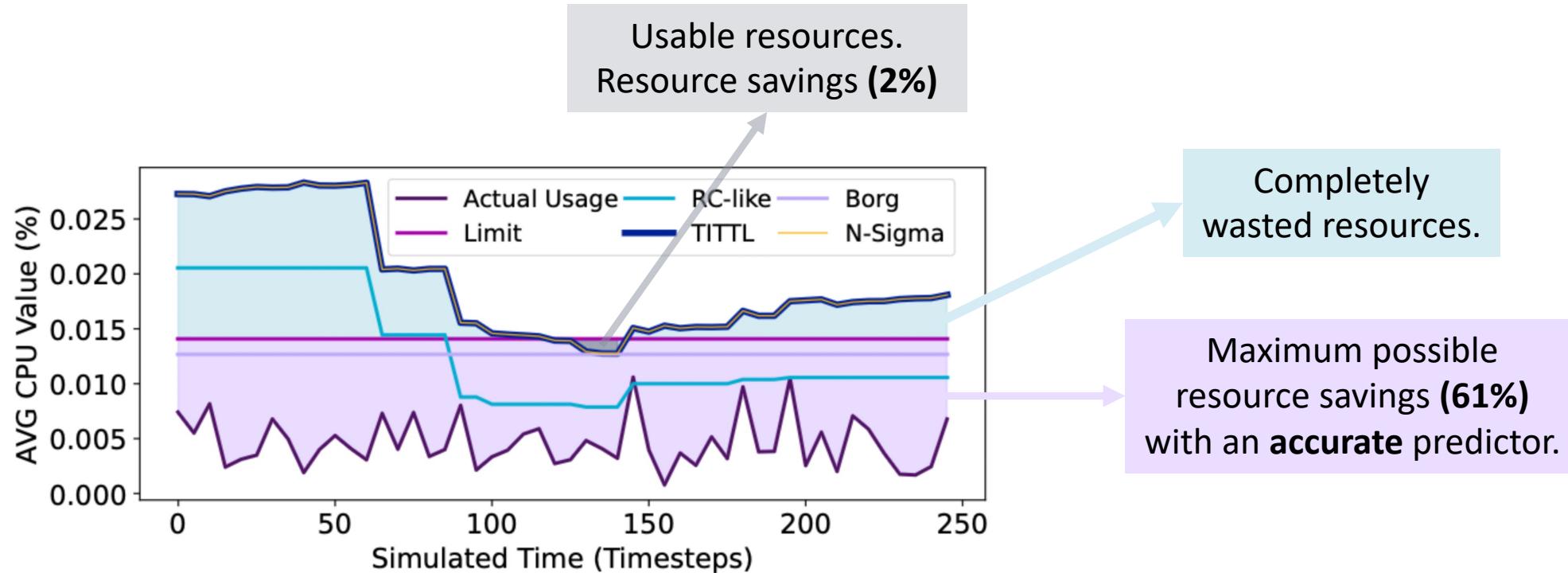


NO overcommitment is happening for **94%** of the cases we examined, due to the predictor's **OVER-estimations**.



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

Missed Opportunity for Resource Savings

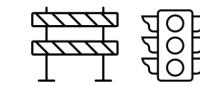


Here it seems to be **necessary** to have a more **accurate** and **intelligent** predictor!
At least a predictor that actually predicts!

Use Machine Learning? Let's use LSTMs!



Predict with high accuracy:



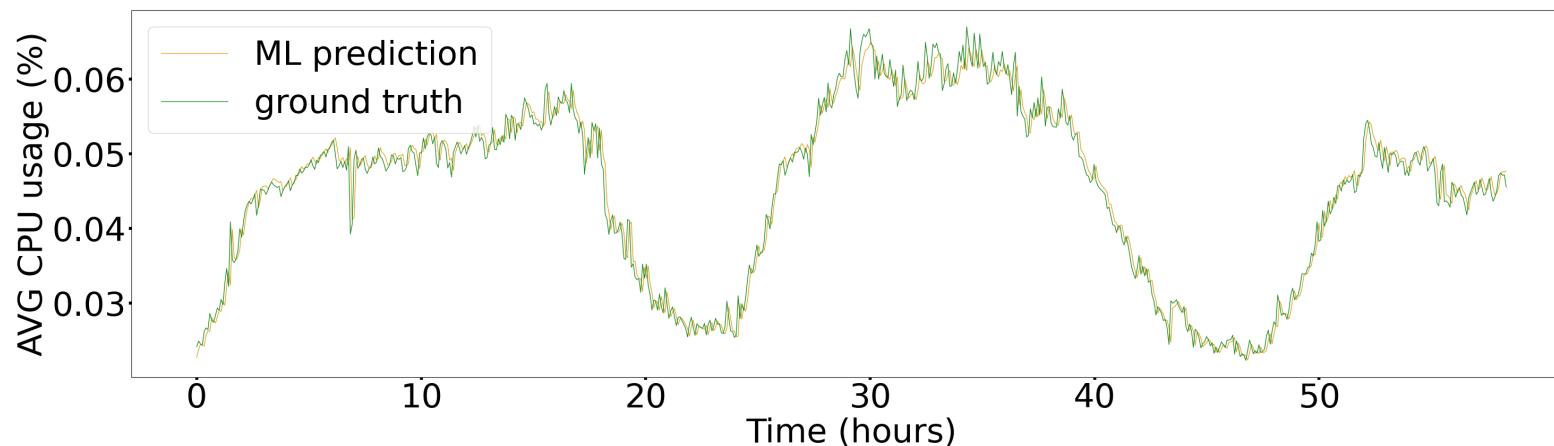
Traffic Conditions



Stock Market Prices

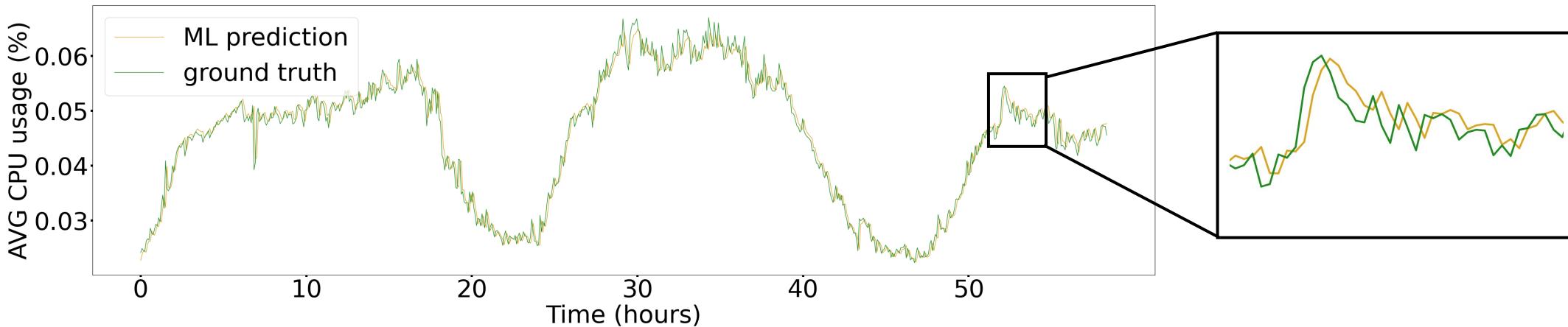


Weather

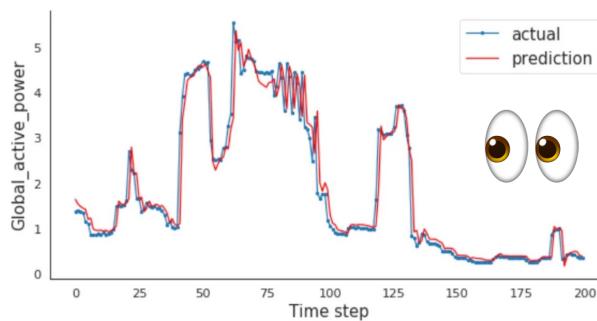


Looks quite accurate for cloud resource usage!!
And seems to generalize well across patterns!

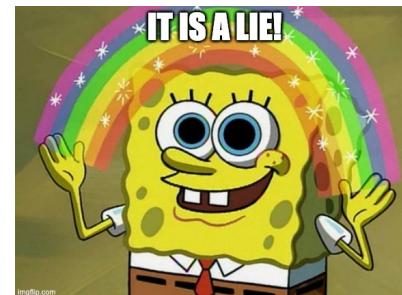
Let's Take a Closer Look!



Insight: LSTM predictions look like “shifted” versions of the real (ground truth) data.



Source “Time Series Analysis, Visualization & Forecasting with LSTM” on [towardsdatascience.com](https://towardsdatascience.com/time-series-analysis-visualization-forecasting-with-lstm-3a2f3a2e3a)

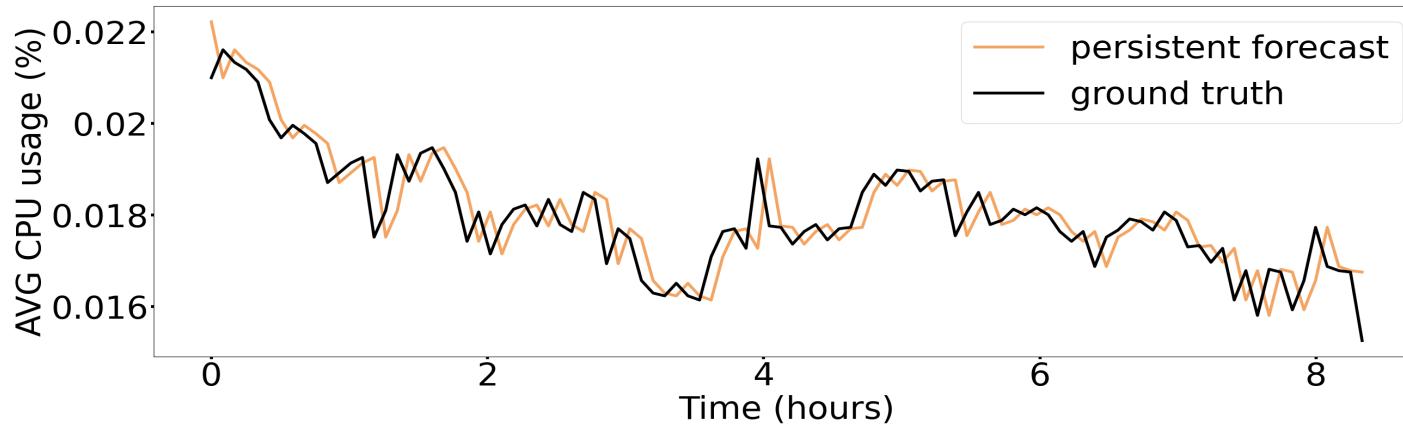


Lesson Learned
Validate that ML *actually* learns!

A Simple and Practical Predictor



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



Persistent Forecast**

*Predicted Value(t) =
Ground Truth($t - 5 \text{ mins}$)*

**[VLDB '21] Seagull: An Infrastructure for Load Prediction and Optimized Resource Allocation. By Microsoft.



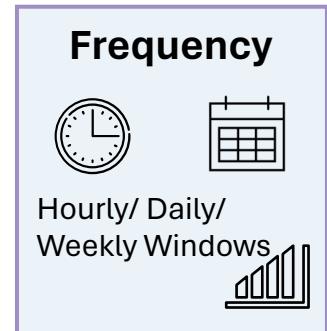
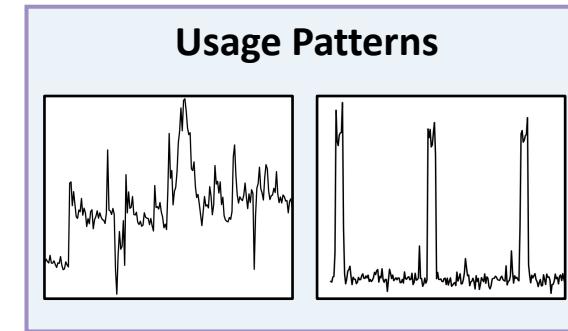
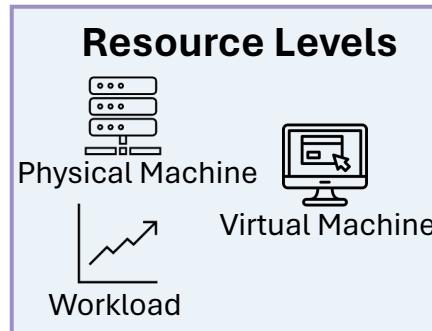
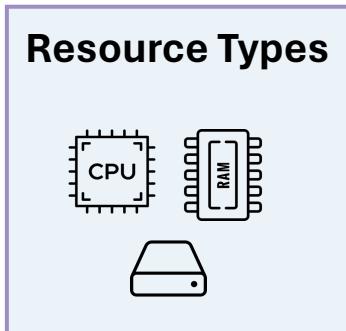
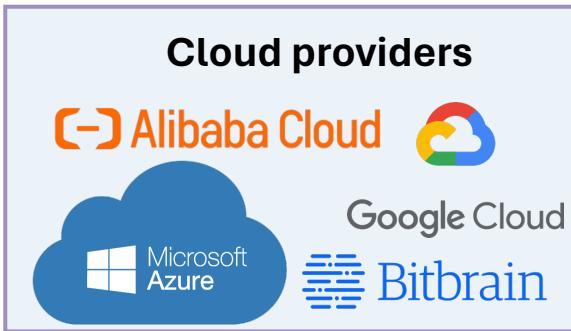
Simple, lightweight, explainable and easy to engineer in production-level.



Does it accurately predict resource usage?

Extensive Experimental Evaluation

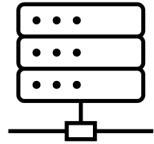
Public open-source datasets across different:



We calculate the **prediction error** of the persistent forecast.

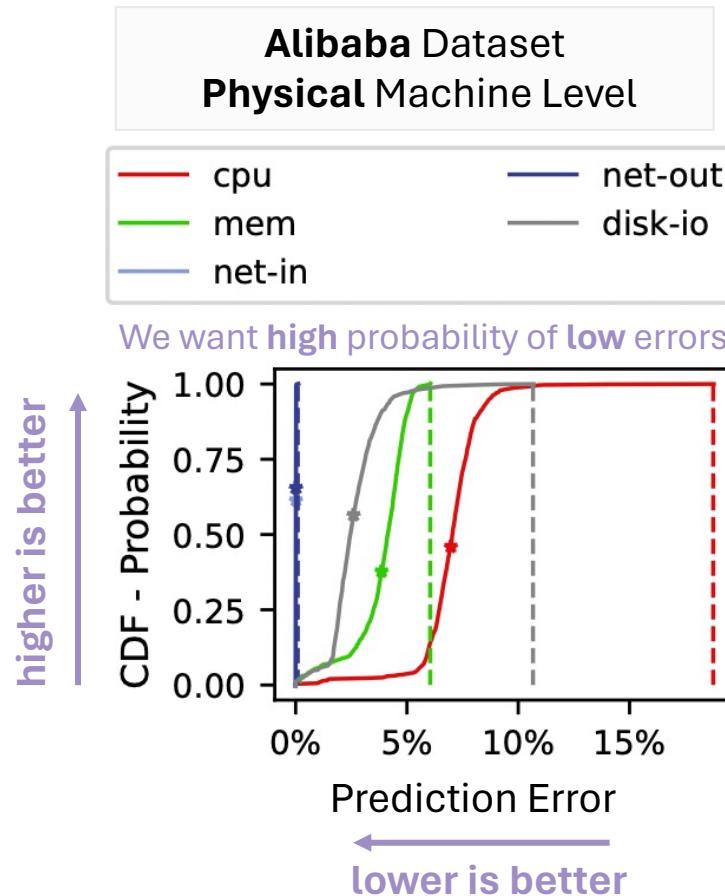


... will it work or do we *need* some other machine learning method?



Physical Machine

Results – Physical Machines



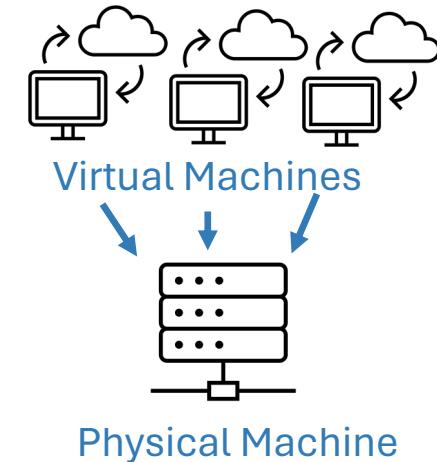
Observations:

- { net-in, net-out } Error ~ 0 😂
- { disk-io, mem } Avg. Error < 4 %
Short tail. 😃
- { cpu } Avg. Error ~ 7 %
Longer tail. 😊

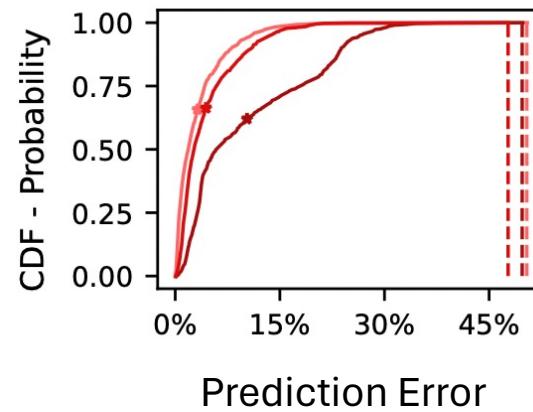
Physical Machines
Have **stable** load.

Persistent Forecast
is **very** accurate!

Results – Virtual Machine

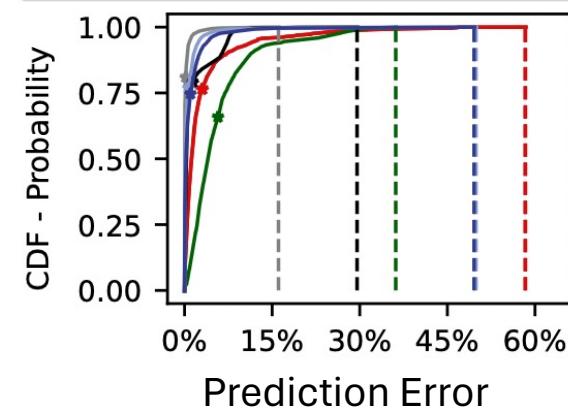


Azure Dataset
Virtual Machine Level

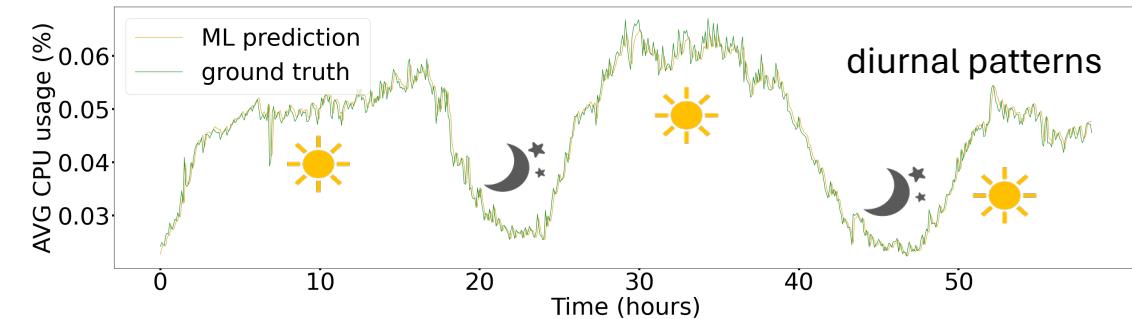


Average Prediction Error < 10%

Bitbrains Dataset
Virtual Machine Level

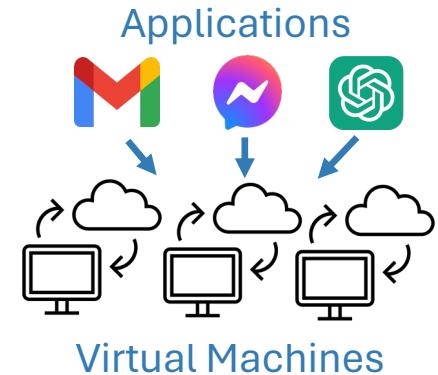
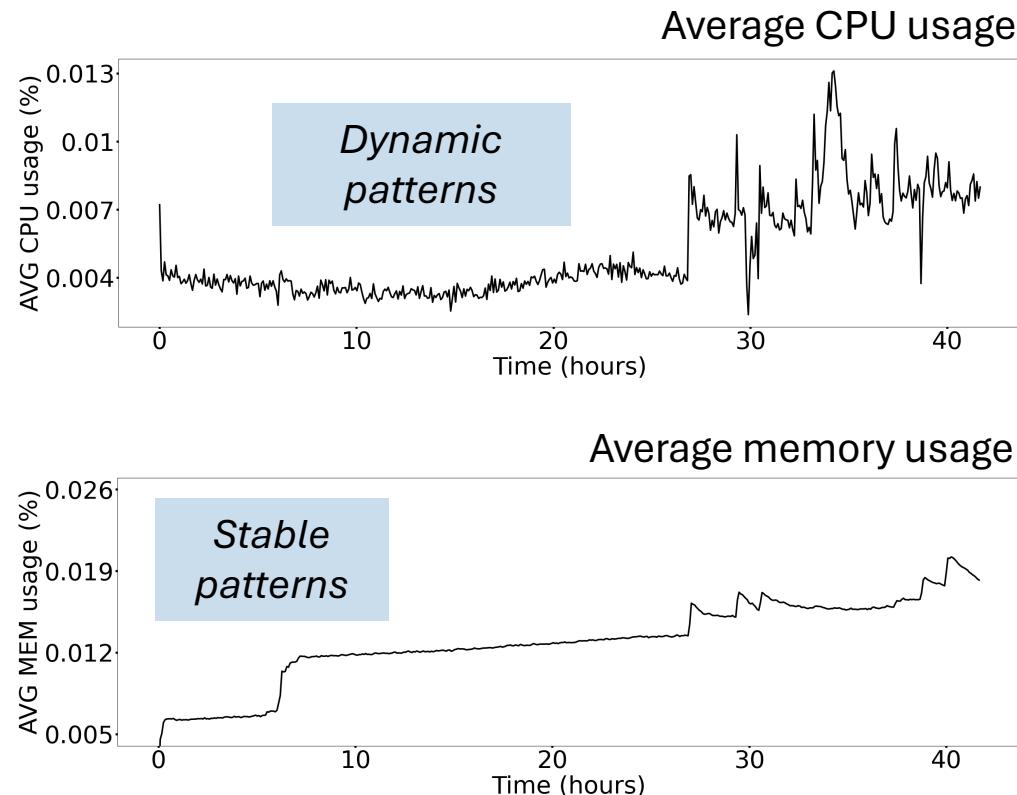
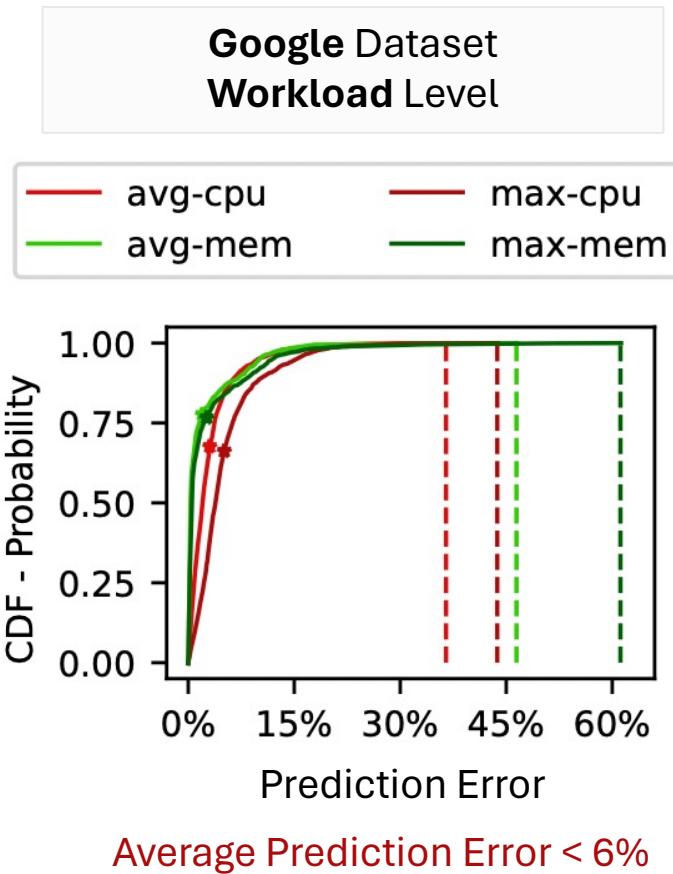


Average Prediction Error < 6 %



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

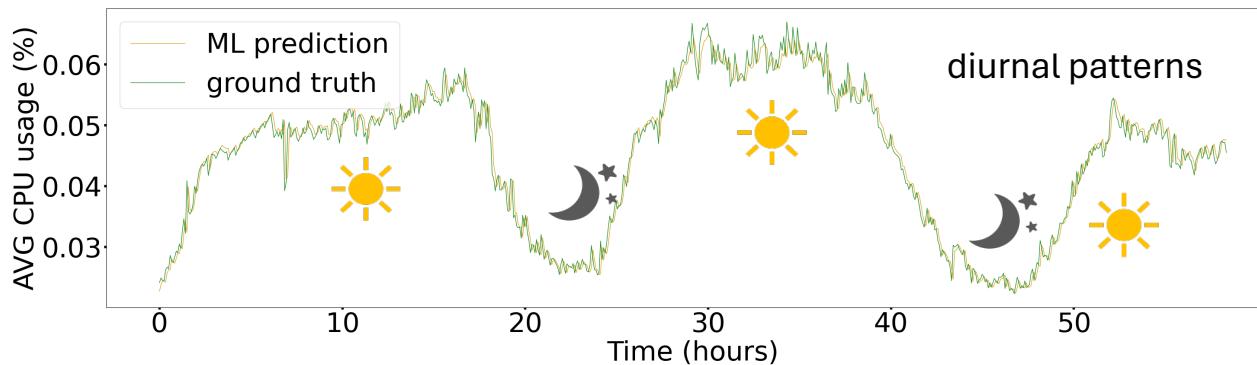
Results – Applications



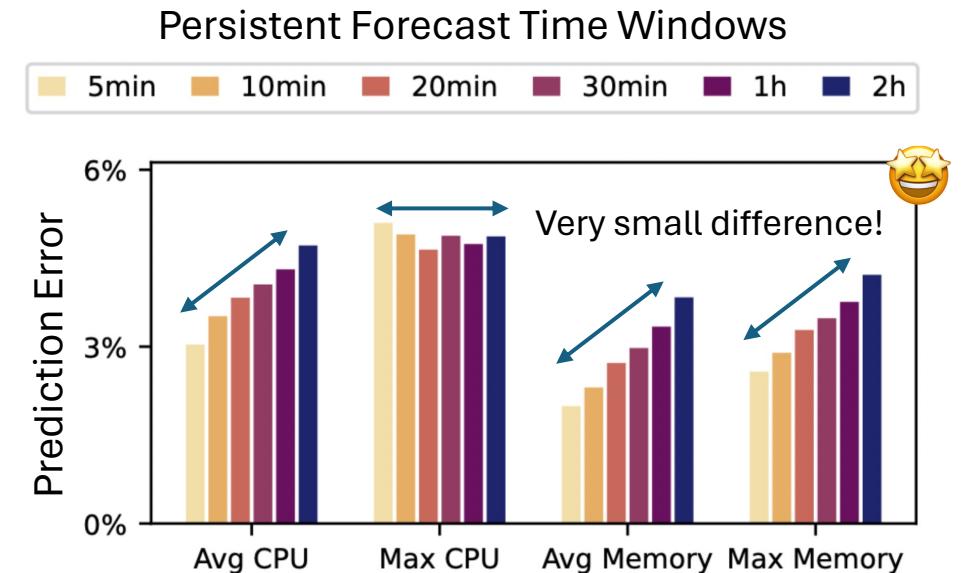
Applications
Most **dynamic** patterns.
(longest tails in the error)
Depends on the
type of resource!

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.



Low sensitivity to the time window length, reveals potential **repeating patterns** in the data. This unlocks **opportunity** for even lower errors, if the time window matches the data periodicity.

Is Machine Learning Necessary?

No!!



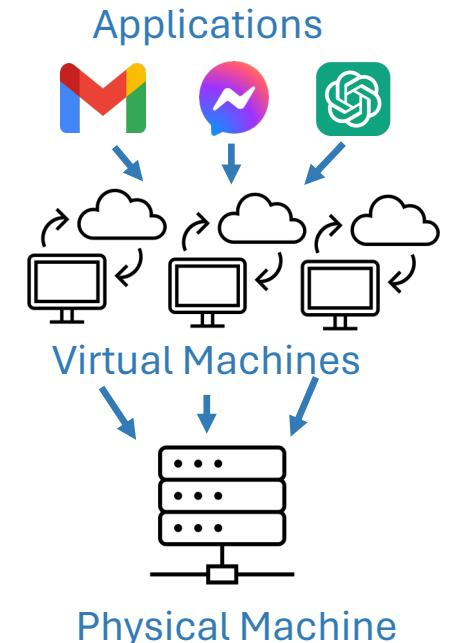
At least not always.. 😬



When is ML necessary?

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

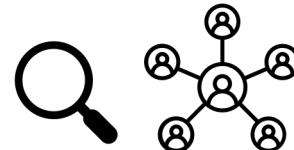
Type	Pattern	ML?
CPU	Dynamic	Yes (?)
Memory	Stable	No
Disk	Stable	No
Network	Stable	No



ML can be useful for reacting to unseen patterns: (1) if similar-seen: predict (2) if unseen: learn.

New Questions and Challenges to Adress

1. When to Use ML?



Build data-driven methods to identify whether to use ML or not, e.g., Pattern detection and classification.

2. Which ML to use?

Probably not LSTMs.. 😭

One of the current SOTA.. transformers? ChatGPT?

[AI4Sys @ HPDC 2024] *Toward Using Representation Learning for Cloud Resource Usage Forecasting*. Razine Ghorab, Thaleia Dimitra Doudali.

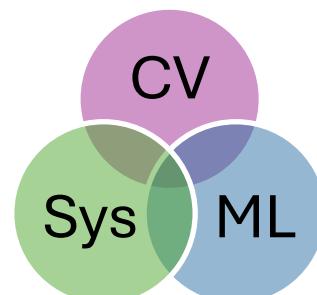
Production likes **simple and explainable**, e.g. decision trees?

3. Validate ML actually learns!

Definitely use visual validation!

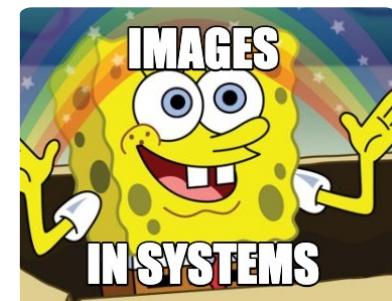


Maybe even some
Computer Vision (CV)
methods?

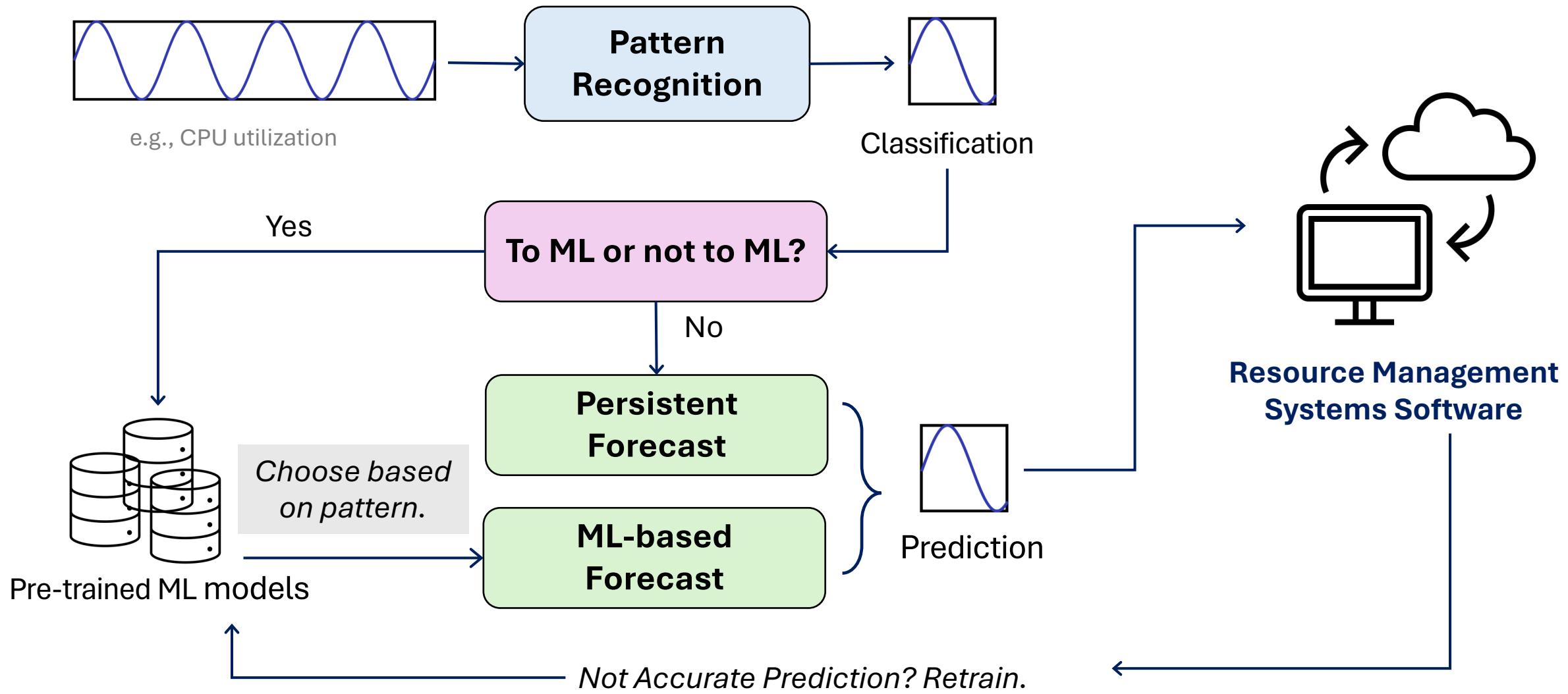


[Wild and Crazy Ideas @ ASPLOS 2022] A picture is worth a 1000... features!

Using Computer Vision alongside Machine Learning in Computer Systems Thaleia Dimitra Doudali.



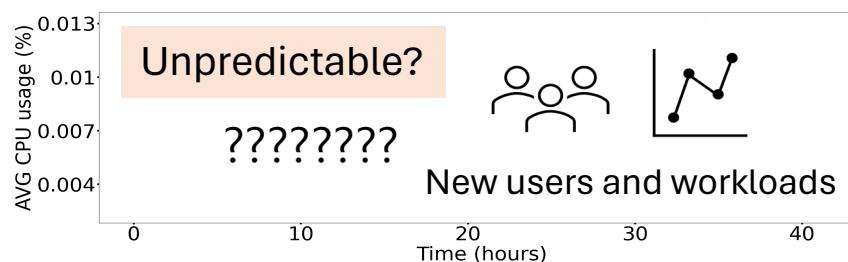
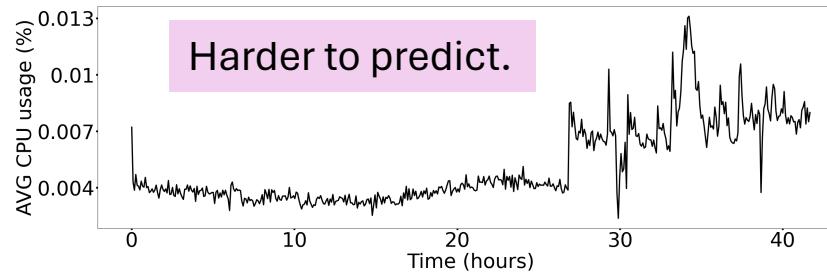
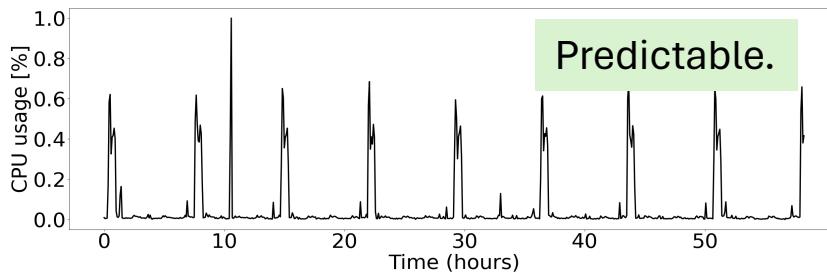
Use ML to Augment, not Replace** - Our Vision



Summary

Machine Learning is ***not always*** necessary in cloud resource management.

It all depends on the patterns.



Use simple, explainable, lightweight ML methods to augment, not replace robust analytical models, like the **persistent forecast**.



[SIGOPS Blog] KISS: Keep it Simple, Smart.

[SIGARCH Blog] Think Twice Before Using Machine Learning to Manage Cloud Resources.

Both written by Thaleia Dimitra Doudali

Thanks to my brilliant PhD students! 🤝



Georgia
Christofidi



Konstantinos
Papaioannou



Website



Github