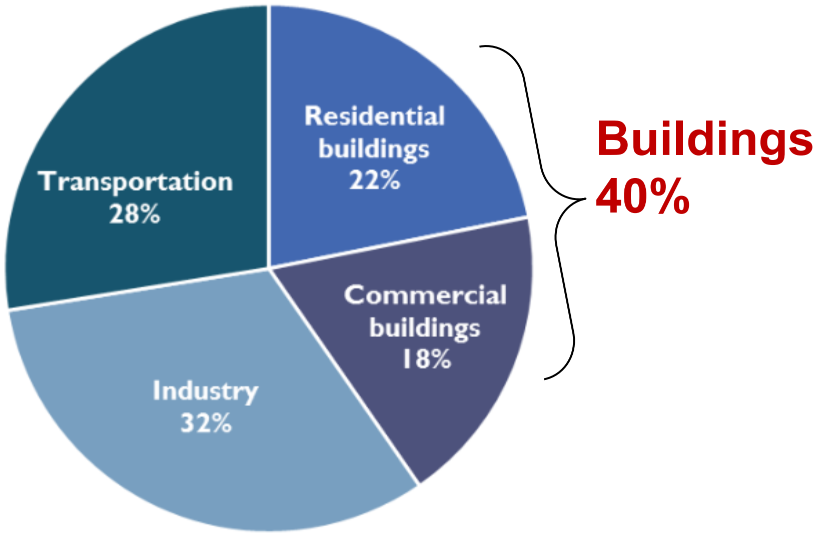
Behavior Testing of Load Forecasting Models using BuildChecks

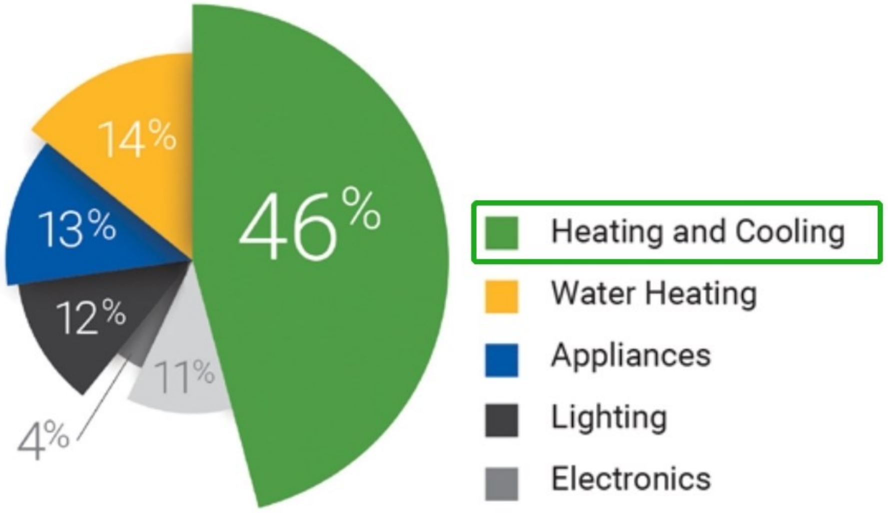
Yang Deng, Jiaqi Fan, Hao Jiang, Fang He,
Dan Wang, Ao Li and Fu Xiao

The Hong Kong Polytechnic University

Background

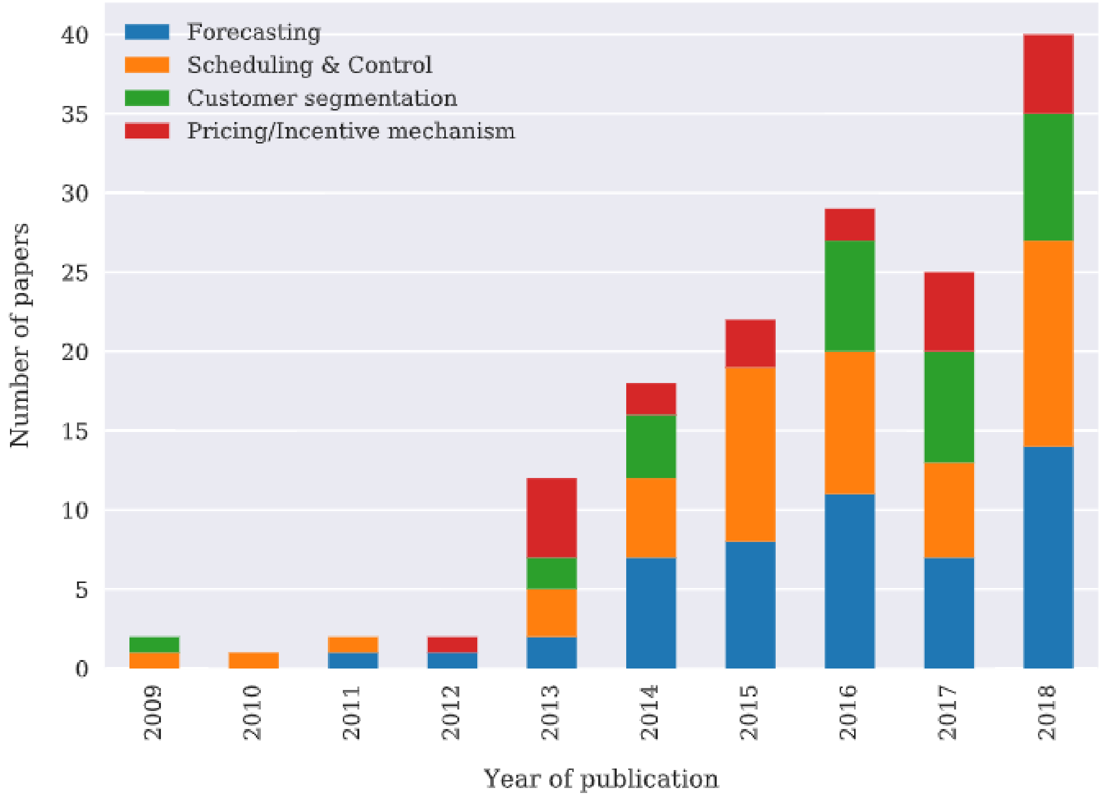
- Energy consumption in buildings
  - In the US, **buildings account for over 40%** of total energy usage. This percentage would be higher in other cities (e.g., 94% in Hong Kong).
  - HVAC system is the building's main energy consumer, **accounting for 46%** of all electricity used.





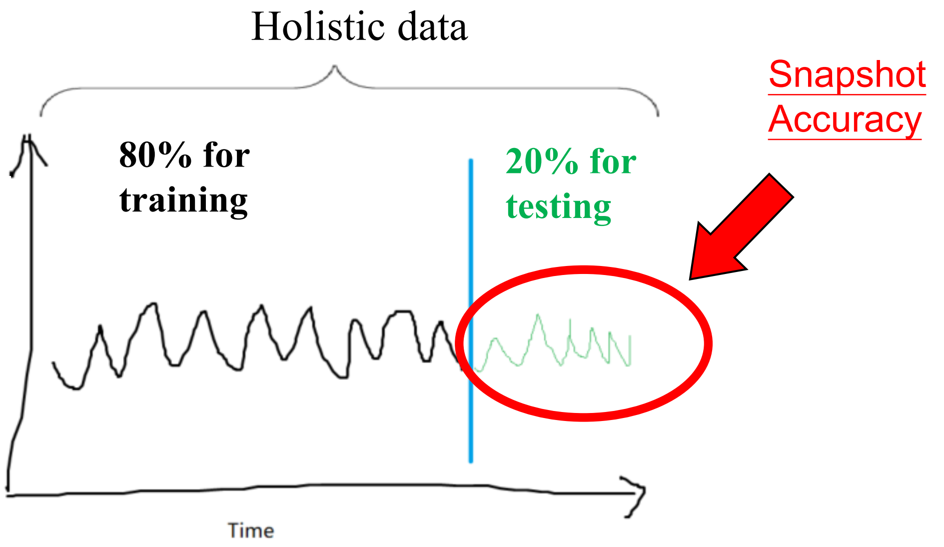
Background

- There is **an increasing number of ML models** target on building energy saving.
  - e.g., the forecasting models.



Background: How we evaluate ML model?

- In research, the conventional evaluation methodology for ML-based forecasting model is:
  - 1) Train the model in the train set, and
  - 2) **Statistic the accuracy in the test set**, e.g., mean absolute error (MAE).



Motivation

- How to check if the model suitable for me?

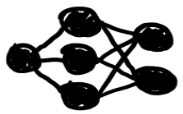
Model developer

Practitioner
(e.g., building operator)

Motivation

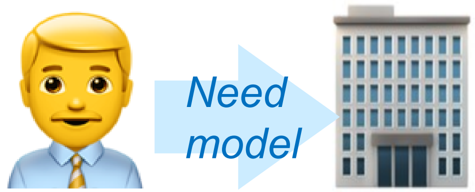
- How to check if the model suitable for me?

My new developed model has high accuracy in my dataset.



Model developer

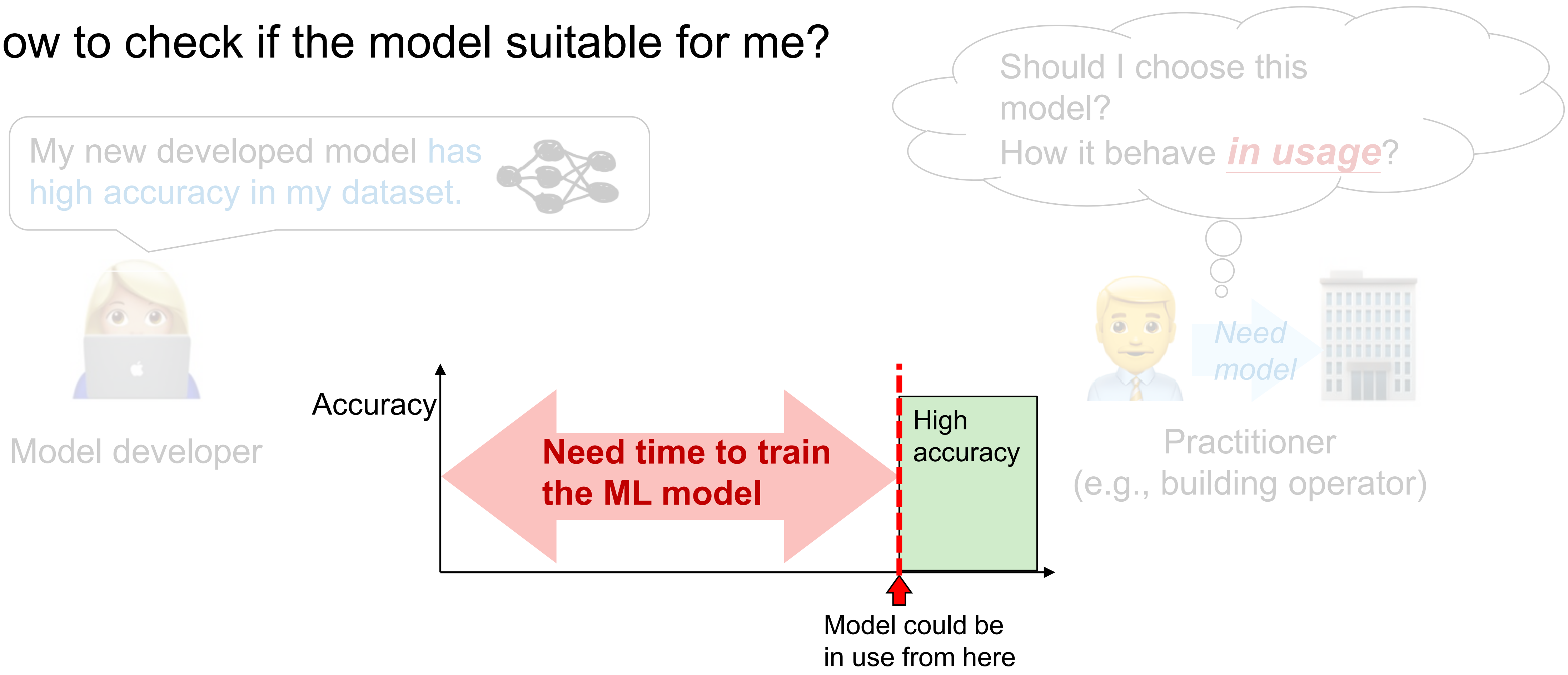
Should I choose this model?
How it behave ***in usage***?



Practitioner
(e.g., building operator)

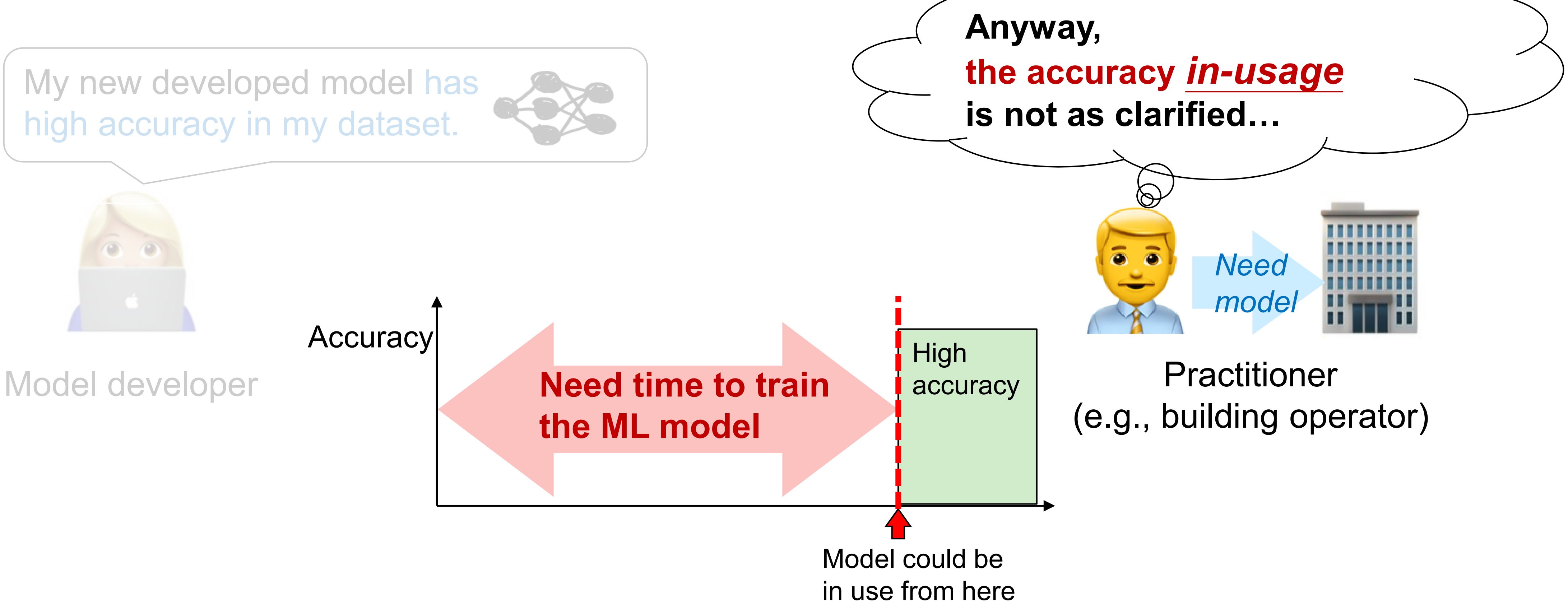
Motivation

- How to check if the model suitable for me?



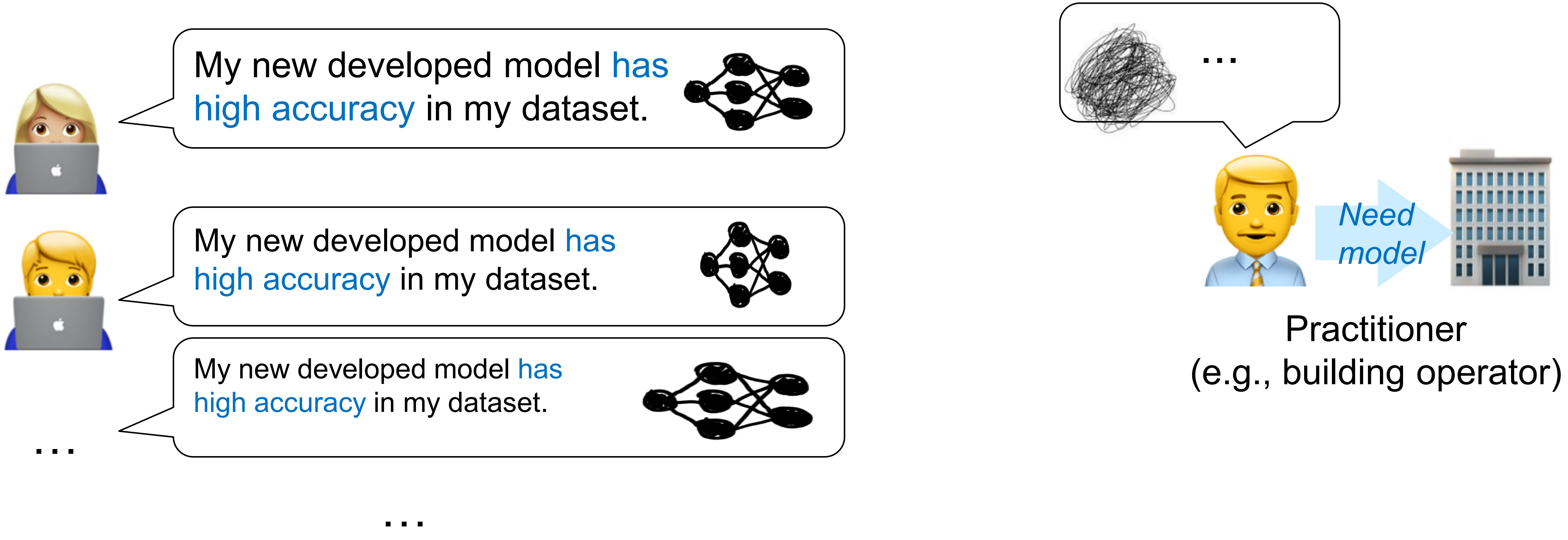
Motivation

- How to check if the model suitable for me?



Motivation

- How do I check if the model suitable for me?



Motivation

- How do I check if the model suitable for me?

× **Snapshot accuracy ?= Accuracy in usage**

× **A lot of evaluation work on multi models**

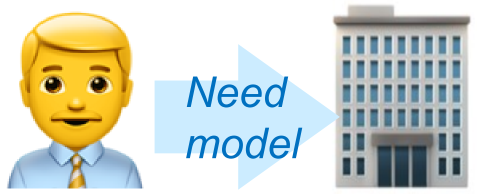
Practitioner
(e.g., building operator)

Motivation

- How do I check if the model suitable for me?

× **Snapshot accuracy ?= Accuracy in usage**

× **A lot of evaluation work on multi models**

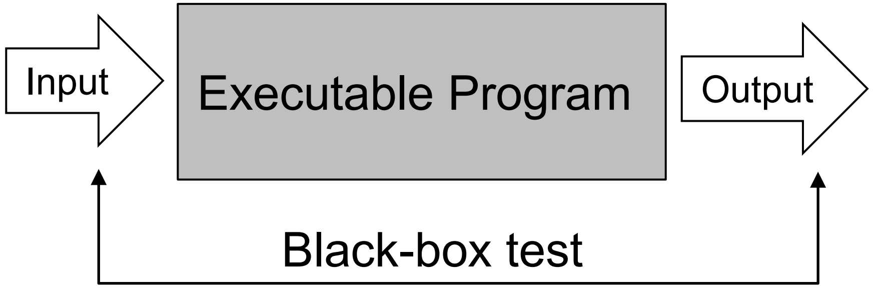


Practitioner
(e.g., building operator)

BuildChecks: <u>**behavior testing**</u> the ML model's in-usage behavior

Overview of BuildChecks

- Behavior testing [1]
  - Core: Behavior testing (also known as black-box testing), a software engineering concept that tests system capabilities by validating the input-output behavior.

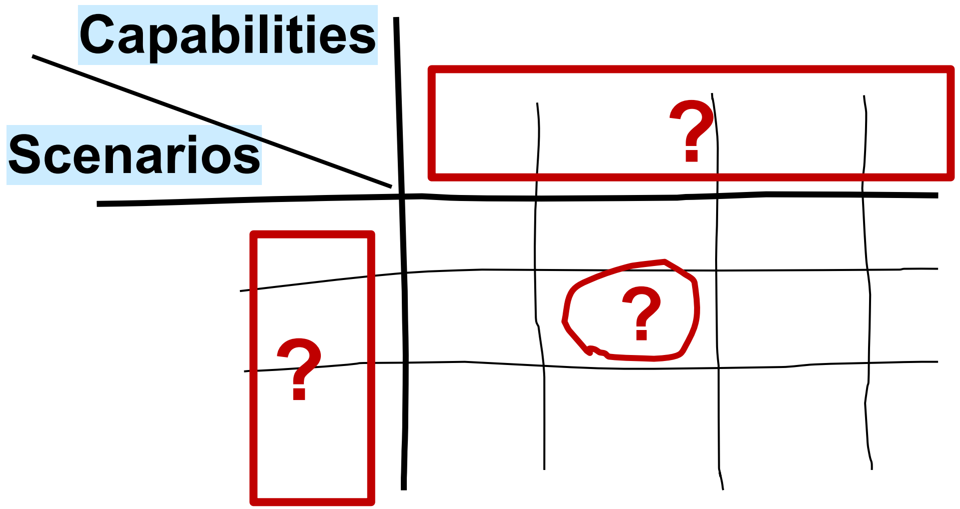


  - Why need it: Practitioners are more eager to understand and compare the in-usage behaviors of a wealth of models, without re-programming for model evaluation.

[1] Boris Beizer. 1995. *Black-box testing: techniques for functional testing of software and systems*. John Wiley & Sons, Inc.

Overview of BuildChecks

- Challenge of developing in-usage behavior testing in building
  - To define organized tests (i.e., what should (or should not) be tested) for diverse building scenarios. ---- ***What to evaluate?***



Overview of BuildChecks

- Behavior testing on building load forecasting model
  - What is building load forecasting?
    - Use monitored historical building loads, building information and ambient weather to forecast building load demand in the future.
  - Why we behavior testing load forecasting?
    - Widely applied in HVAC control optimization and building monitoring system, etc.

Behavior Testing Methodology

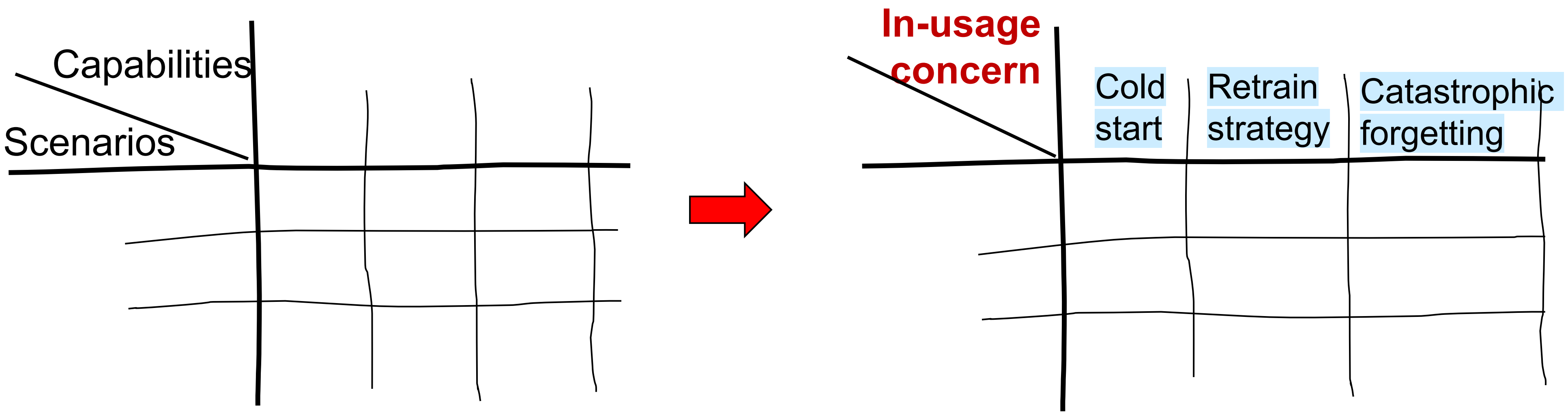
- We define behavior testing methodology from three organized aspects:

1. Selection of in-usage concerns. We bring three typical ones from AI community:
    - Cold start:
        - History data needs to be accumulated so that an ML model can be trained to an acceptable accuracy.
    - Retrain strategy:
        - Models face concept drifts (i.e., pattern change) and need to be retrained to maintain accuracy.
    - Catastrophic forgetting:
        - ML model may forget existing learned knowledge.

Behavior Testing Methodology

- We define behavior testing methodology from three organized aspects:

1. Selection of in-usage concerns. We bring three typical ones from AI community:



Behavior Testing Methodology

- We define behavior testing methodology from three organized aspects:

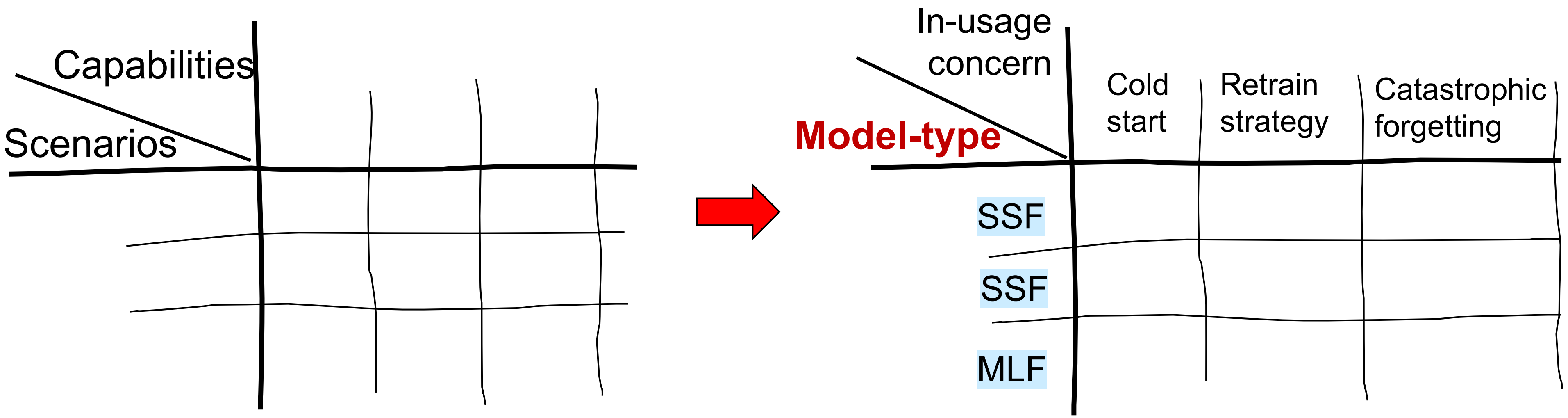
2. Load forecasting model categorization. we categorize ML models into three model-types, which are widely-accepted in building automation studies.

  - Short-term forecasting with snapshot designs (SSF):
    - i.e., The model focus on model structure design (e.g., how to design neural network layers).
  - Short-term forecasting with online learning (SOF):
    - i.e., The model with internal mechanisms to online update the model parameters.
  - Midterm/long-term forecasting (MLF)

Behavior Testing Methodology

- We define behavior testing methodology from three organized aspects:

2. Load forecasting model categorization. we categorize ML models into three model-types, which are widely-accepted in building automation studies.



Behavior Testing Methodology

- We define behavior testing methodology from three organized aspects:

3. Defining tests:

Concerns in Usage / Model Types	Cold Start	Retrain Strategy	Catastrophic Forgetting
Short-term forecasting with snapshot design (SSF)			
Short-term forecasting with online learning (SOF)			
Middle\long-term forecasting (MLF)			

√: It should be evaluated.　×: It is not meaningful to evaluated.　N/A: Cannot evaluate.

Behavior Testing Methodology

- We define behavior testing methodology from three organized aspects:

3. Defining tests:

Concerns in Usage / Model Types	Cold Start	Retrain Strategy	Catastrophic Forgetting
Short-term forecasting with snapshot design (SSF)	√	√	N/A
Short-term forecasting with online learning (SOF)			
Middle\long-term forecasting (MLF)			

√: It should be evaluated. ×: It is not meaningful to evaluated. N/A: Cannot evaluate.

Behavior Testing Methodology

- We define behavior testing methodology from three organized aspects:

3. Defining tests:

Concerns in Usage / Model Types	Cold Start	Retrain Strategy	Catastrophic Forgetting
Short-term forecasting with snapshot design (SSF)	√	√	N/A
Short-term forecasting with online learning (SOF)	√	√	√
Middle\long-term forecasting (MLF)			

√: It should be evaluated.   ×: It is not meaningful to evaluated.   N/A: Cannot evaluate.

Behavior Testing Methodology

- We define behavior testing methodology from three organized aspects:

3. Defining tests:

Concerns in Usage / Model Types	Cold Start	Retrain Strategy	Catastrophic Forgetting
Short-term forecasting with snapshot design (SSF)	√	√	N/A
Short-term forecasting with online learning (SOF)	√	√	√
Middle\long-term forecasting (MLF)	×	√	√

√: It should be evaluated. ×: It is not meaningful to evaluated. N/A: Cannot evaluate.

Metrics

- Cold start
  - Holdout and prequential method [1]
- Retrain strategy [2]
  - Periodically retrain
  - Informed retrain (involves two classical concept drift detection algorithms: DDM and ADWIN)
- Catastrophic forgetting [3]
  - Average forgetting

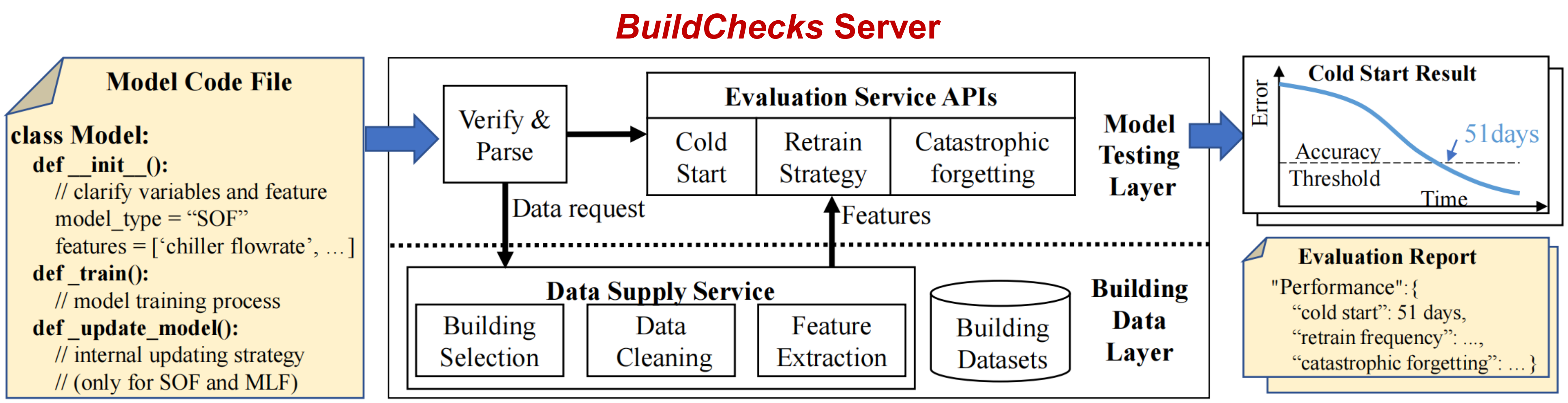
[1] On evaluating stream learning algorithms. Machine learning 90, 3 (2013),317-346
[2] A survey on concept drift adaptation. ACM computing surveys (CSUR) 46, 4 (2014),1-37
[3] A continual learning survey: Defying forgetting in classification tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)

BuildChecks Platform Design

- Goals
  - Goal 1: Provide automatic evaluation services.
  - Goal 2: BuildChecks can also serve the users, if they do not have data.

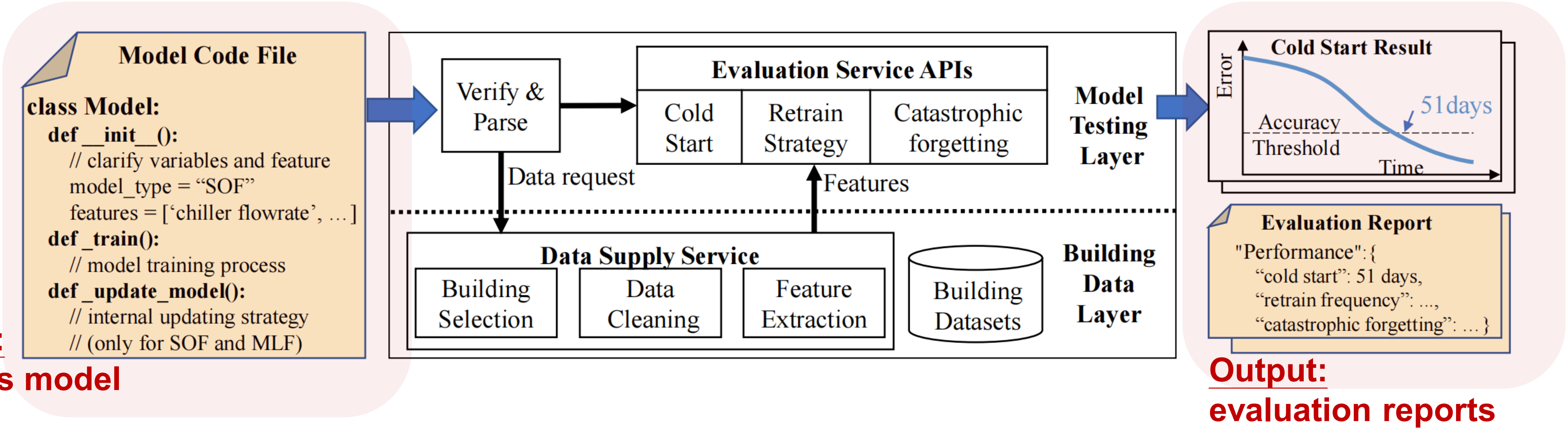
BuildChecks Platform Design

- Goals
  - Goal 1: Provide automatic evaluation services.
  - Goal 2: BuildChecks can also serve the users, if they do not have data.



BuildChecks Platform Design

- Goals
  - Goal 1: Provide automatic evaluation services.
  - Goal 2: BuildChecks can also serve the users, if they do not have data.

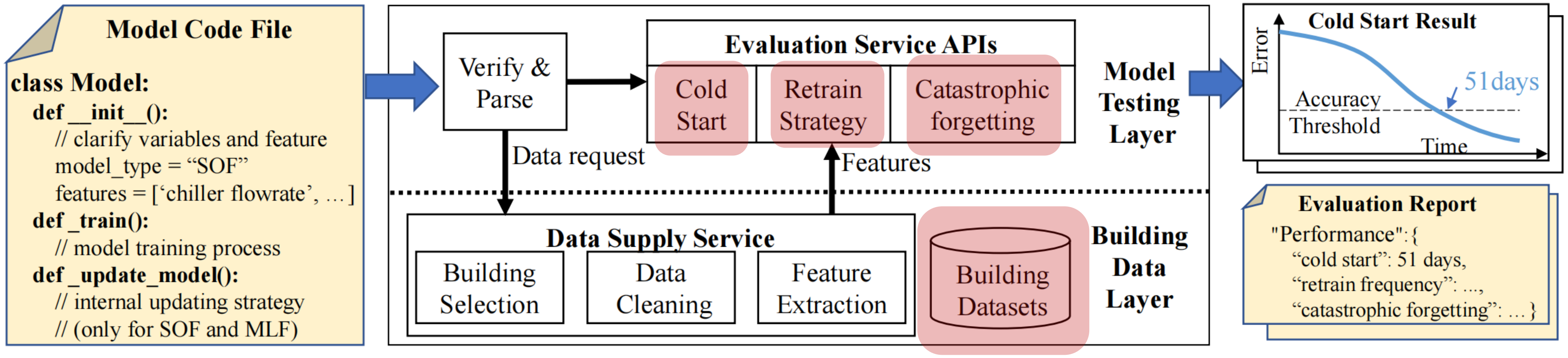


Input: User's model

Output: evaluation reports

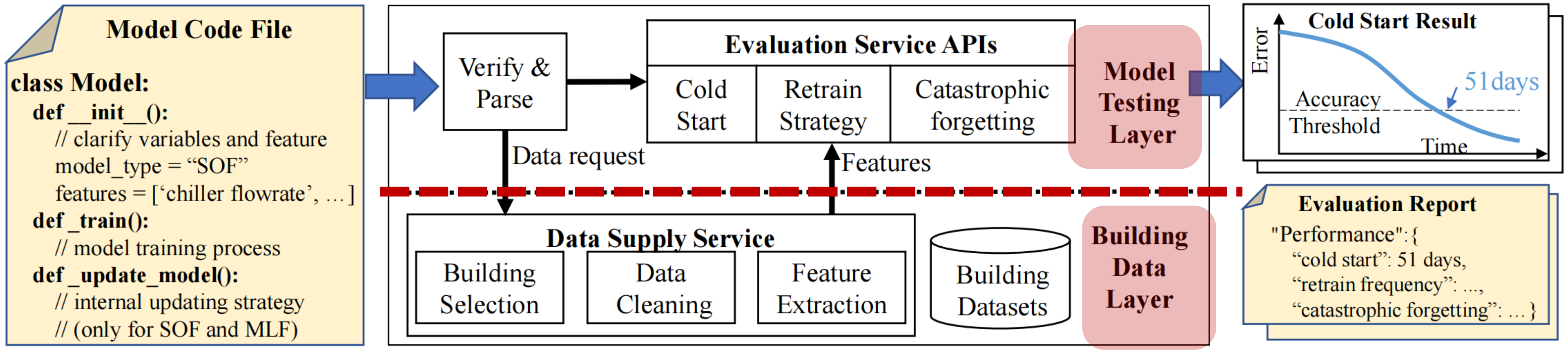
BuildChecks Platform Design

- Goals
  - Goal 1: Provide automatic evaluation services.
  - Solution: To integrate default data and test algorithms (Binary-search for cold start; three retrain strategies (periodical, DDM and ADWIN based); statistic built-in function for catastrophic forgetting).



BuildChecks Platform Design

- Goals
  - Goal 2: BuildChecks can also serve the users, if they do not have data.
  - Solution: Separate model testing layer and building data layer.



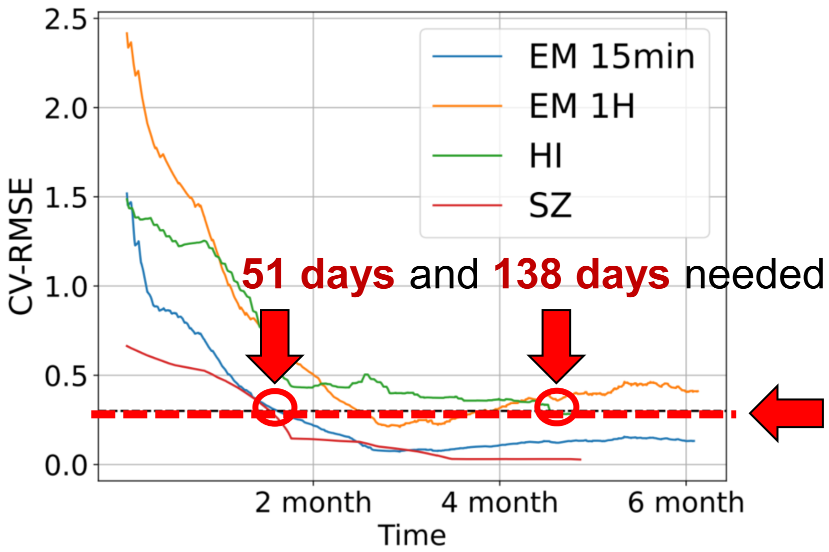
Case studies

- We use BuildChecks to evaluate the behaviors of two representative models of SOF and SSF type:
  - The London-Residential model [1]:
    - SOF model. The model is based on LSTM, with adaptive buffering to handle contextual adaptation.
  - The HK-ICC model [2]:
    - SSF model. It has an attention based NN to extract the temporal load pattern.

[1] Mohammad Navid Fekri, et al. Deep learning for load forecasting with smart meter data: Online adaptive recurrent neural network. *Applied Energy* 282 (2021), 116177.
[2] Ao Li , et al. Attention-based interpretable neural network for building cooling load prediction. Applied Energy 299 (2021),117238.

Evaluation results of *London-Residential*

- Evaluation result on Cold Start



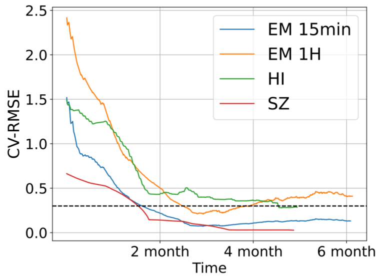
51 days and **138 days** needed in dataset SZ and HI respectively, about **2.7 times** greater.

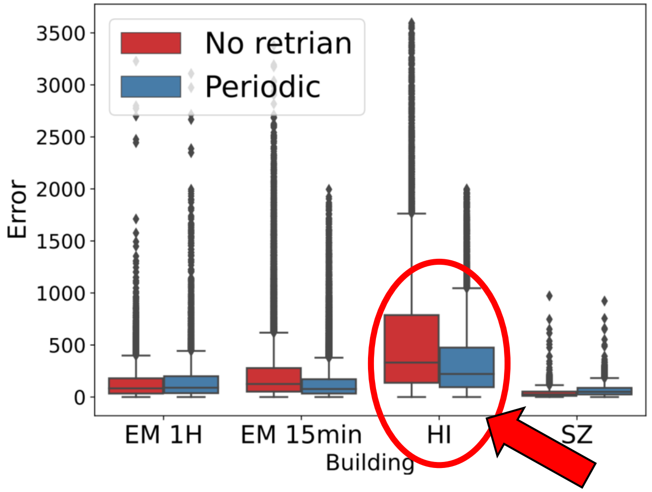
Accuracy threshold:
CV-RMSE < 30% defined by ASHRAE

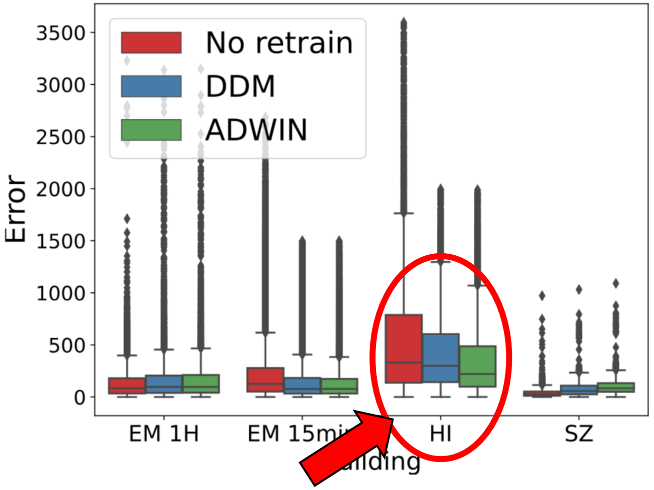
(a) The cold start is different in buildings: 84 days, 58 days, 138 days and 51 days in the four datasets.
(b) A shorter cold start does not mean greater accuracy in usage. The accuracy in usage is 27% lower in EM_1H while compared to in HI, although the cold start is shorter in EM_1H.

Evaluation results of *London-Residential*

- Evaluation result on Retrain strategy







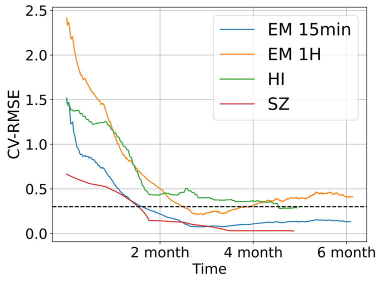
High benefit by the retrain strategies, especially periodic method, achieve **36%** improvement.

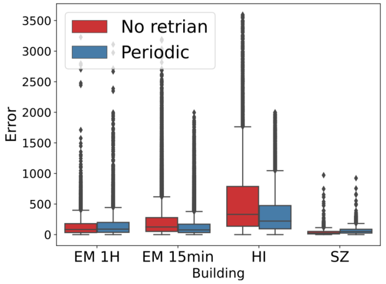
(a) In HI building, the accuracy in use can be improved by 36%, 29% and 14% for the three retrain strategies (periodic, ADWIN and DDM).

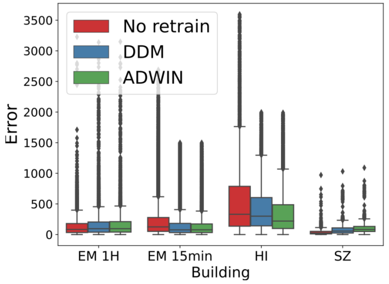
(b) The improvement in other datasets is within 6%. there is a very slight accuracy decay in SZ.

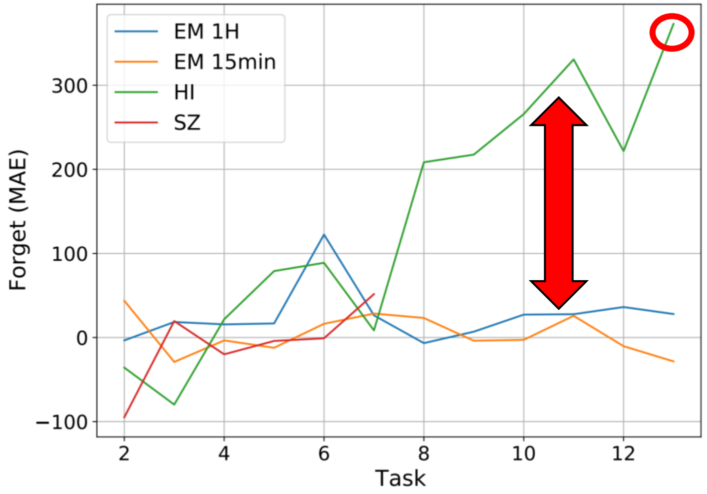
Evaluation results of *London-Residential*

- Evaluation result on Catastrophic forgetting









Progressively worse over time. (mean error is **over 378**)

No significant forgetting.

(a) During the first five updates, not only the model does not forget learned knowledge, but also enhances its prediction capacity.
(b) The forget value is gradually increasing to over 378 in HI after model update 12 times, which illustrates the model is getting worse on the previous load context.

Conclusion

- BuildChecks, a behavior testing methodology and an associated platform to evaluate the behaviors of ML-based building load forecasting models.
- BuildChecks can output reports on the three in-usage concerns. BuildChecks complements the understanding of the ML models from the perspective of model accuracy to a richer context.
- We comment that BuildChecks is designed to provide guidelines on the behavior testing process in smart building, so that models can be comprehensively compared.

Thank you!

Q&A