AugPlug: An Automated Data Augmentation Model to Enhance Online Building Load Forecasting

Yang Deng, Rui Liang, Yaohui Liu, Jiaqi Fan, and Dan Wang

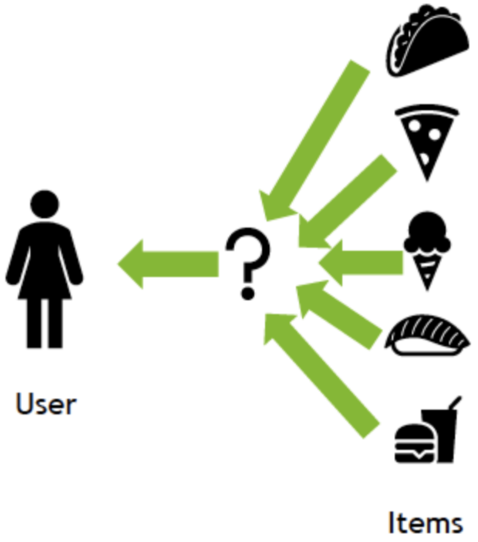
The Hong Kong Polytechnic University

THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

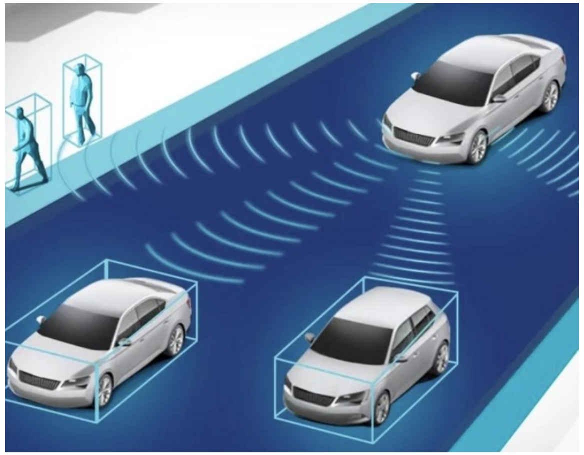
Department of Computing
電子計算學系

Background

- Online machine learning, i.e., the practitioner needs to continuously update the ML model during deployment phase
  - Scenarios: data is too large to be processed at once; or data distribution constantly changes.



Recommendation system



Automatic driving

Background

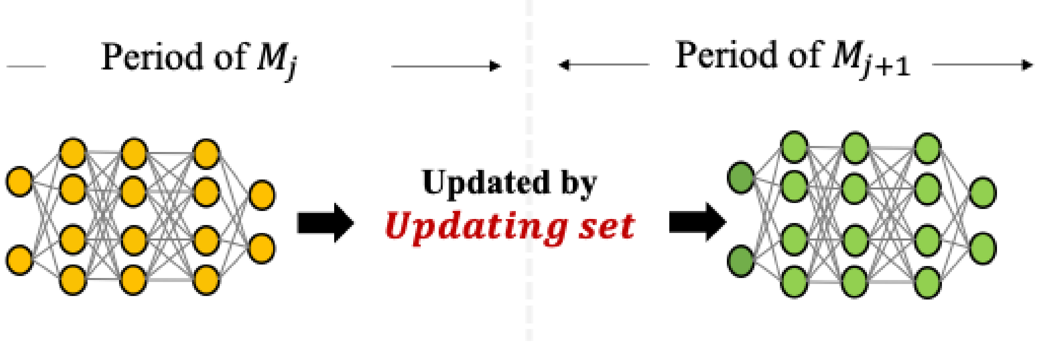
- Online machine learning, i.e., the practitioner needs to continuously update the ML model during deployment phase
  - Scenarios: data is too large to be processed at once; or data distribution constantly changes.

  - **Key components:**
    - 1) The model update strategy

Learning mode / Adaptation	Retrain	Fine-tune
Periodically		
Triggered		

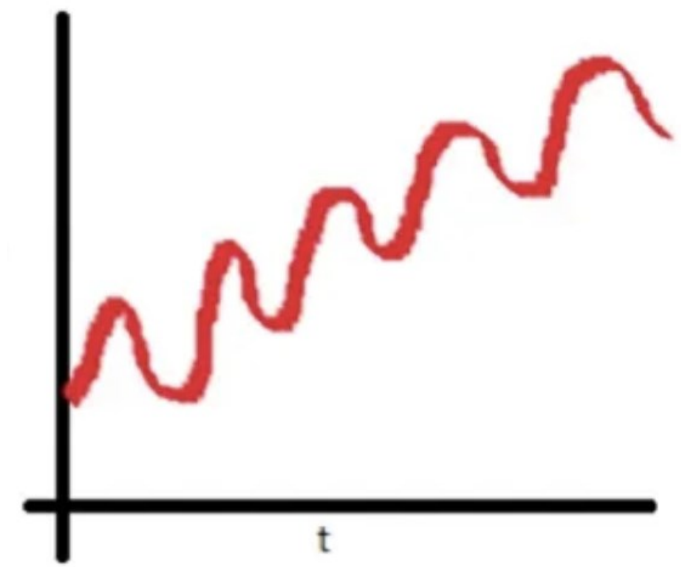
(The taxonomy of the update strategy)

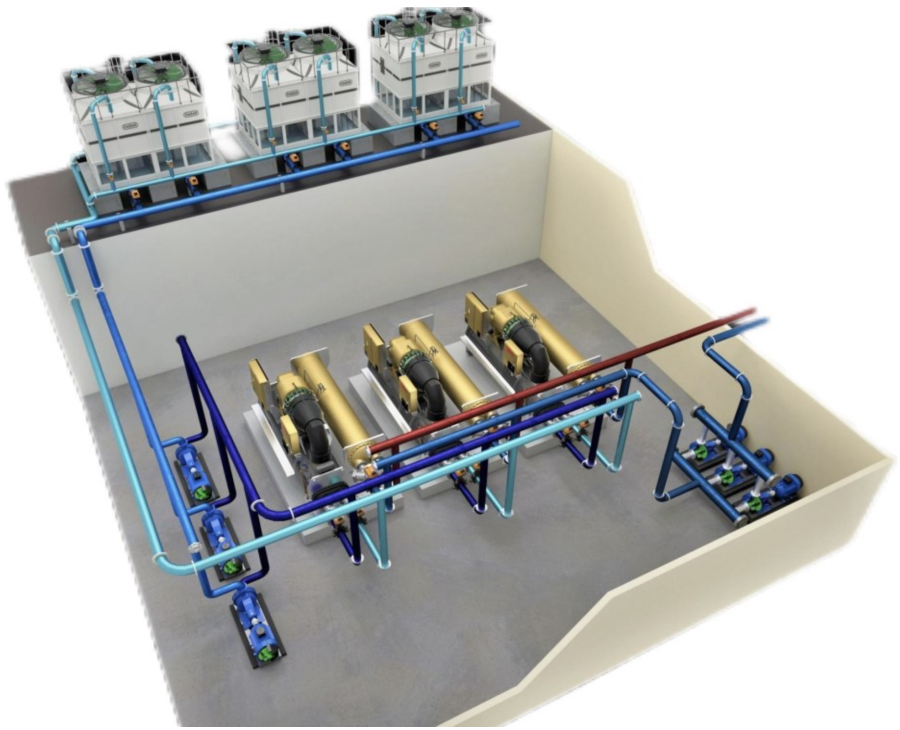
- 2) The updating set, i.e., the data used to perform model update.



Background

- Online building load forecasting (Online BLF)
  - Scenario: **data distribution is constantly changing**.





Background

- Online building load forecasting (Online BLF scheme)
  - Scenarios: data distribution is constantly changing.
  - The mentioned two key components:
    - 1) The model update strategy: the existing online BLF schemes falls into this taxonomy

Learning mode / Adaptation	Retrain	Fine-tune

Background

- Online building load forecasting (Online BLF scheme)
  - Scenarios: data distribution is constantly changing.
  - The mentioned two key components:
    - 1) The model update strategy: the existing online BLF schemes falls into this taxonomy

Update strategy:
weekly retrain is adopted to update the SVM-based BLF model in COVID-19

Learning mode / **Adaptation**	Retrain	Fine-tune
Periodically	• *SVM (Energy Reports. 2023)* • *Ensemble (SUSTAIN ENERGY GRIDS. 2021)*	• *HMM (Trans.Power Syst. 2021)* • *LSTM (Applied Energy. 2021)*
Triggered	• *Random Forest (Sustainability. 2022)* • *KNN (Sensors. 2020)*	• *Autoencoder (Applied Energy. 2022)* • *RNN (Applied Energy. 2020)*

(Some selected online BLF schemes from top journal)

Background

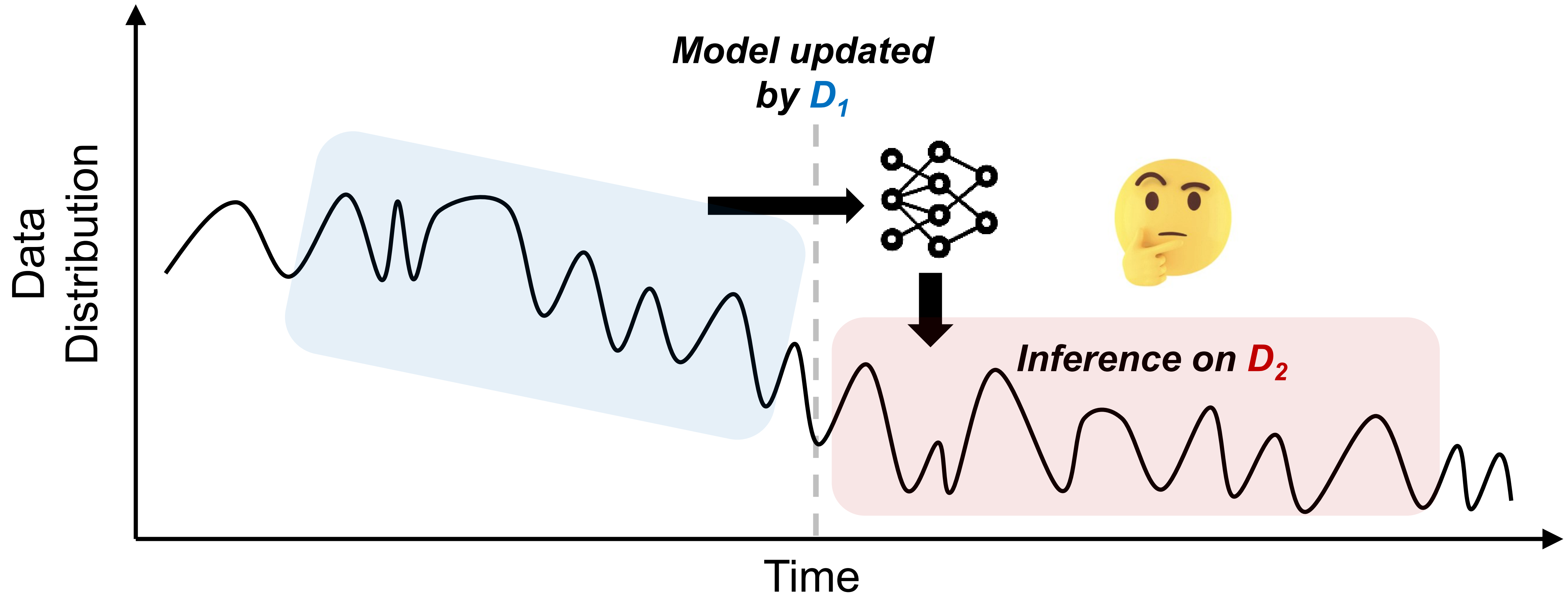
- Online building load forecasting (Online BLF scheme)
  - Scenarios: data distribution is constantly changing.
  - The mentioned two key components:
    - 1) The model update strategy: the existing online BLF schemes falls into this taxonomy

Learning mode / **Adaptation**	Retrain	Fine-tune
Periodically	• *SVM (Energy Reports. 2023)* • *Ensemble (SUSTAIN ENERGY GRIDS. 2021)*	• *HMM (Trans.Power Syst. 2021)* • *LSTM (Applied Energy. 2021)*
Triggered	• *Random Forest (Sustainability. 2022)* • *KNN (Sensors. 2020)*	• *Autoencoder (Applied Energy. 2022)* • *RNN (Applied Energy. 2020)*

    - **2) The updating set: All these schemes using the *latest arrived data* to update the ML model**

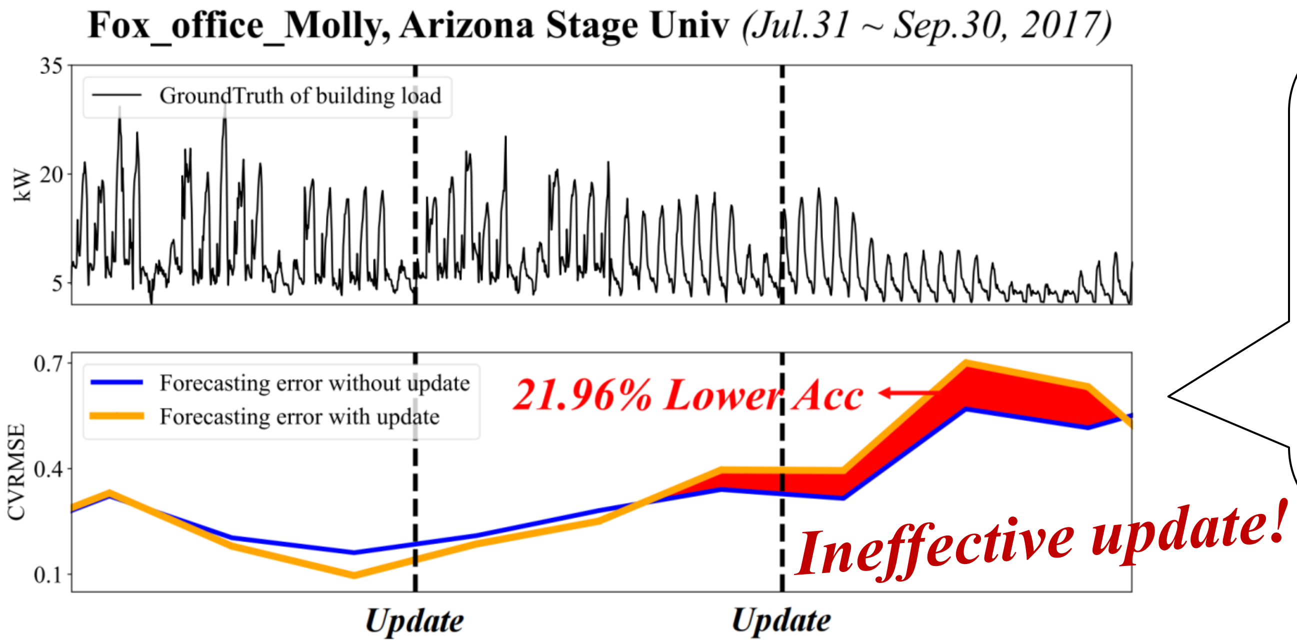
What's the problem?

- *Collected historical data* may not reflect the characteristics of the *future data distribution*.



Motivation

- The tested online BLF scheme: *LSTM (Applied Energy. 2021)*
- The deployed Building: *a university office building with a two-year length*
- Experiment settings: 1) 24h ahead forecasting: first year for training + second year for deployment; 2) online updating *vs* frozen model.



1. Updates can not always bring accuracy improvement
2. The model with updates even worse than without update

Motivation

- Settings:
  - The online BLF schemes: from the four groups (different update strategy)
  - Buildings: using a public building dataset covers 500+ buildings with different building types, e.g., education, residential.
  - Metrics: A/B testing → "*Ineffective* update"
- Key observation:
  - 30.6% updates are ineffective
  - About 12% updates result in "> 10% acc decay"

Adaptation \ Learning mode	Retrain	Fine-tune
Periodically	SVM (Energy Reports. 2023) Ensemble (SUSTAIN ENERGY GRIDS. 2021)	HMM (Trans.Power Syst. 2021) LSTM (Applied Energy. 2021)
Triggered	Random Forest (Sustainability. 2022) KNN (Sensors. 2020)	Autoencoder (Applied Energy. 2022) RNN (Applied Energy. 2020)

Online BLF scheme	Num of Update (10^3)	Ratio of Ineffec-tiveness
SVM [35]	10.7	29.9%
LSTM [15]	6.5	27.3%
RF[29]	15.3	32.3%
Autoencoder [14]	11.4	30.7%

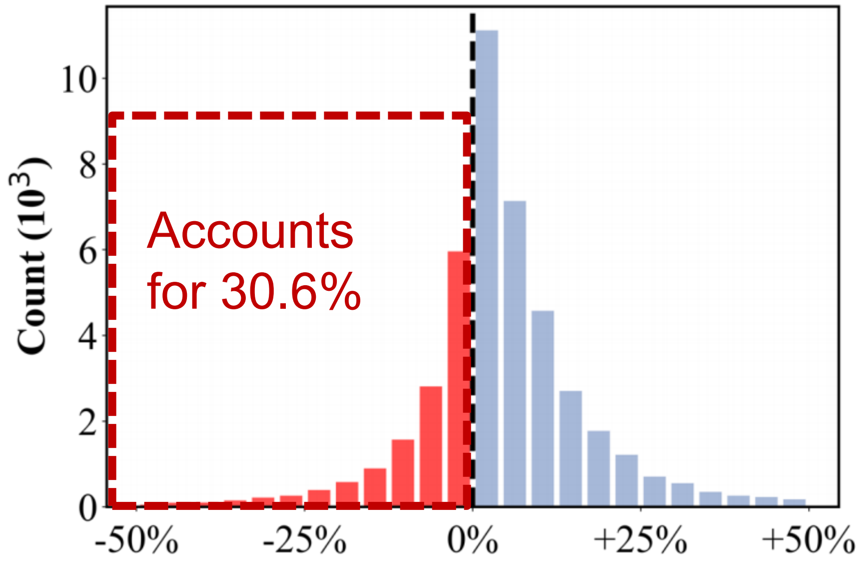


Figure 2: Accuracy improved through updates.

Potential approach ?

- Two directions for enhancing the performance of online BLF
  - 1) Dynamically modify the ML model’s structure in real time
  - 2) To generate synthetic data to serve as the updating set, which involves the characteristics of the upcoming data stream

Potential approach ?

- Two directions for enhancing the performance of online BLF
  - 1) Dynamically modify the ML model's structure in real time
  - 2) To generate synthetic data to serve as the updating set, which involves the characteristics of the upcoming data stream

1) The existing solutions in building scenario are not suitable for data distribution changes.

2) Directly forecast the data stream is impractical, especially the long period

3) Preparing an appropriate update set requires manual effort and expertise.

Potential approach: AutoDA

- Two directions for enhancing the performance of online BLF
  - ~~1) Dynamically modify the ML model's structure in real time~~
  - 2) To generate synthetic data to serve as the updating set, which involves the characteristics of the upcoming data stream

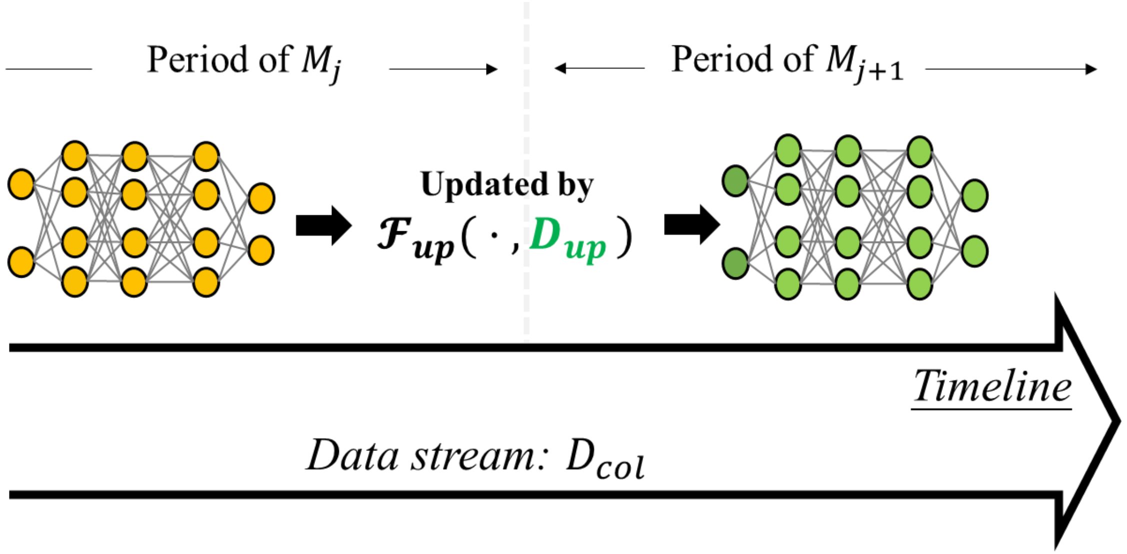
Automated data augmentation [1] (AutoDA):

- *Definition:* the task of searching for suitable data augmentation ***policy***
  - *Policy* → the choices and orders of the data transformation operations
- *Real world example*: Google self-driving product (*Waymo*).
- *The core:* the search algorithm as well as the search space
  - *Reinforcement learning (RL)* is commonly used where an *RNN-based RL agent* is applied to search the policy.

[1] Tsz-Him Cheung and Dit-Yan Yeung. A Survey of Automated Data Augmentation for Image Classification: Learning to Compose, Mix, and Generate. *IEEE Transactions on Neural Networks and Learning Systems (2023)*.

Problem Statement

- Online BLF scheme
  - BLF model sequence of $\{M_1, M_2, \dots\}$ in the deployment phase, and the accuracy of M_j in its time slot is $ACC_{val}(M_j)$
  - The original updating set $D_{up} = \{(x_i, y_i)\}_{i=1}^{m}$ is extracted from the observed data stream D_{col}
  - The update strategy function (designed in the scheme): $F_{up}: D_{up}^{j}, M_j \to M_{j+1}$



Problem Statement

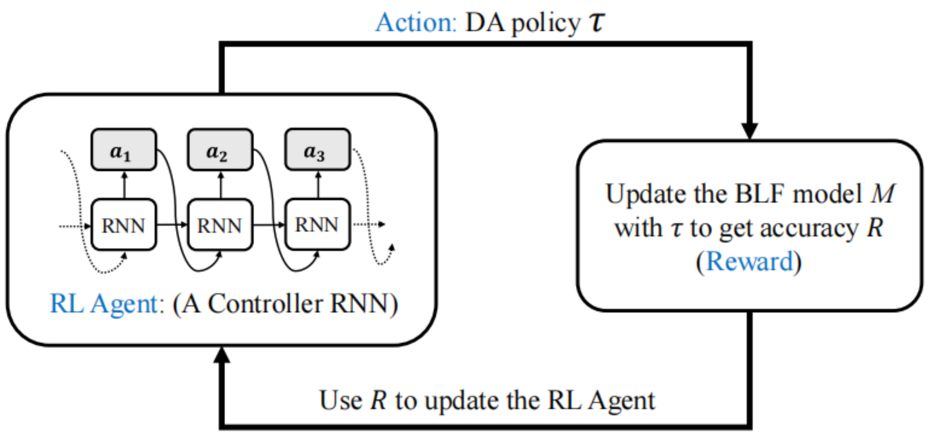
- Online BLF scheme
  - BLF model sequence of $\{M_1, M_2, \dots\}$
  - The original updating set $D_{up} = \{(x_i, y_i)\}_{i=1}^{m}$ is extracted from the observed data stream D_{col}
  - The pre-designed update strategy function in the scheme: $F_{up}: D_{up}^{j}, M_j \to M_{j+1}$
  - The accuracy of a M_j in its time slot is $ACC_{val}(M_j)$
- **For each update operation $M_j \to M_{j+1}$:**
  - Given the operational BLF model M_j, the update function F_{up}, and the observation data D_{col}^{j},
  - Then a data augmentation policy τ transforms the original updating set D_{up}^{j} to a synthetic updating set $D_{\tau}^{j} = \tau(D_{up}^{j})$, and leads to a high accuracy of the updated model M_{j+1}
  - The goal of the AutoDA model is to find the optimal τ^*:

$$\tau^* = \arg\max_{\tau} \mathcal{ACC}_{val}(M_{j+1}^{\tau}), \quad j = 1, 2, \dots$$

$$\text{s.t. } M_{j+1}^{\tau} = \mathcal{F}_{up}(M_j, D_{\tau}^{j}),$$

Solution design

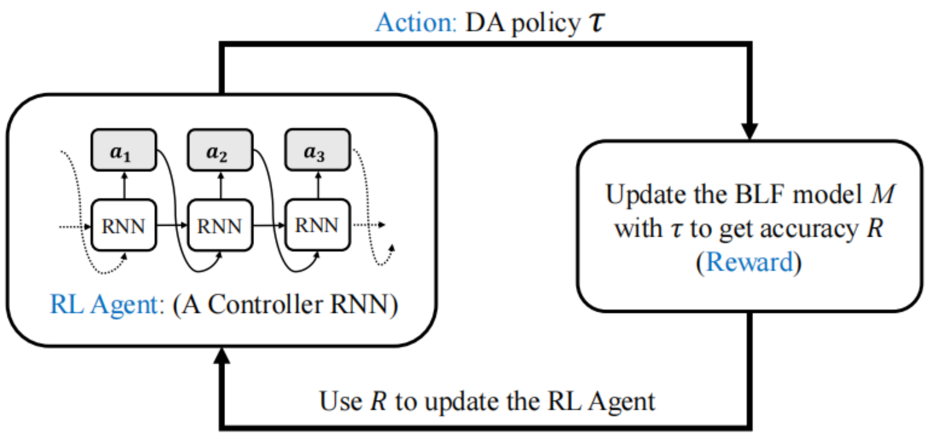
- Reinforcement learning (RL) formulation
  - State
  - Action & search space
  - Reward



Overview: A controller RNN (RL agent) predicts a DA policy τ from the search space. The forecasting model is updated to achieve an accuracy R. The reward R will be used with the policy gradient method to update the controller so that it can generate better policies over time.

Solution design

- Reinforcement learning (RL) formulation
  - State: the collected observed data D_{col}^{j}
    - the historical data stream (energy consumption trace, outdoor temperature, etc.)
    - the sequential records of the accuracy of BLF



Solution design

- Reinforcement learning formulation
  - Action: data transformation operators, defined as:

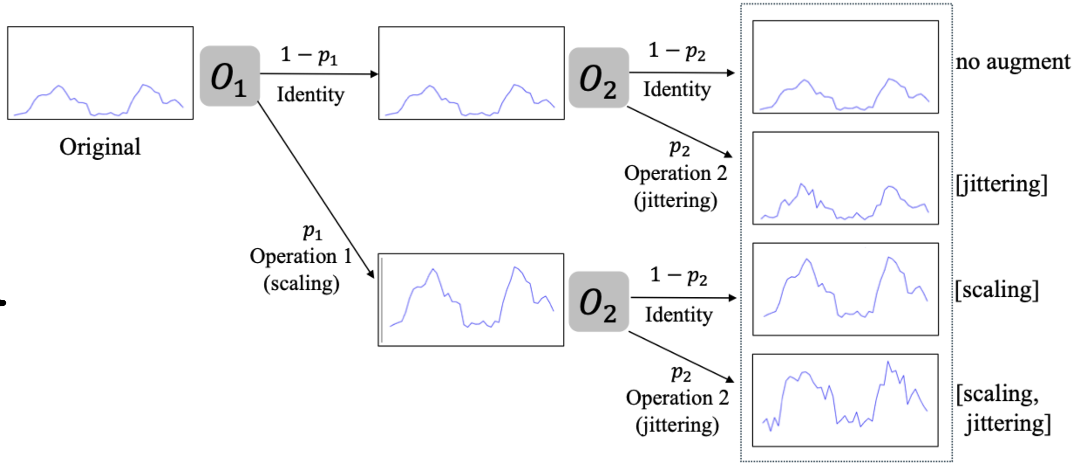
    (1) the type of transformation t ; (2) the magnitude with which the operation is applied λ; and (3) the probability of applying this operation p.

    $$\tau = \{O_n(t_n, \lambda_n, p_n) : n = 1, 2, ..., N\} \quad (3)$$

  - Search space: Time-series transformations and the associated magnitude range

Type	Description	Magnitudes
Scaling	Multiplies the entire series controlled by λ.	[1,3], [0.3,1]
Jittering	Adds white noise with σ controlled by λ .	[0, 0.1]
Smoothing	Performs low-pass filtering using a average window (with size λ).	(0, 11]
Shifting	Adding λ on the entire series.	[-0.5, 0.5]

$$\tau = \{O_1\ (\ t{=}scaling,\quad \lambda{=}2.0,\quad p{=}p_1\ ),\ O_2\ (\ t{=}jittering,\quad \lambda{=}0.05,\quad p{=}p_2\ )\}$$

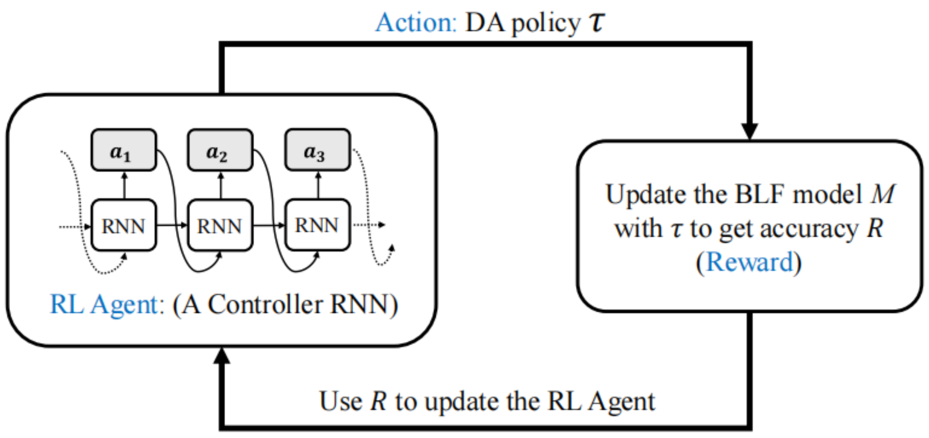


An Example of DA policy with two operators

Solution design

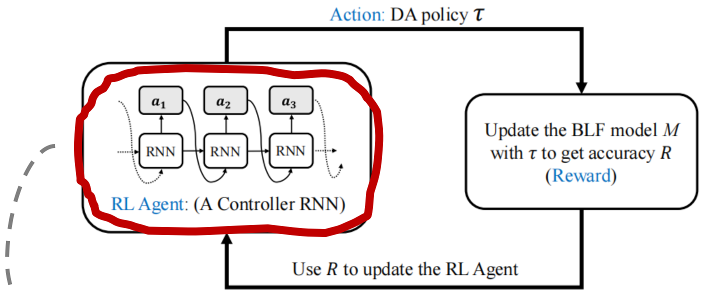
- Reinforcement learning formulation
  - Reward: Our objective is to enhance the accuracy of the BLF model M, thus the reward is an improvement in the accuracy in the time slot of M_{j+1} (i.e., from conducting an update on M_j to the next update).

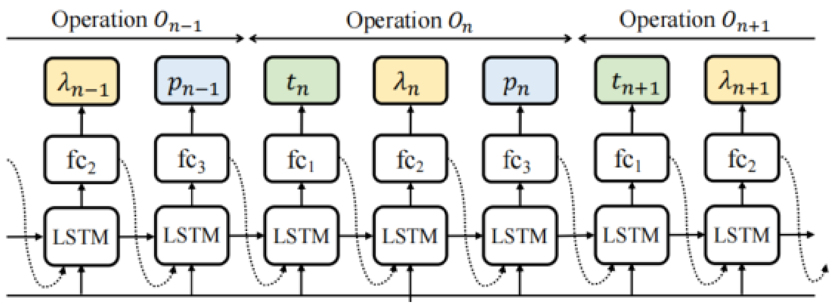
$$R = \mathcal{ACC}_{val}(M_{j+1}^{\tau}) - \mathcal{ACC}_{val}(M_{j+1}) \qquad (4)$$



Solution design

- RL agent design: a controller RNN
  - One-layer LSTM (with 100 hidden units)
  - DA policy τ = $\{ a_1, \ldots a_T \} = \{ t_1, \lambda_1, p_1, \ldots, t_N, \lambda_N, p_N \}$



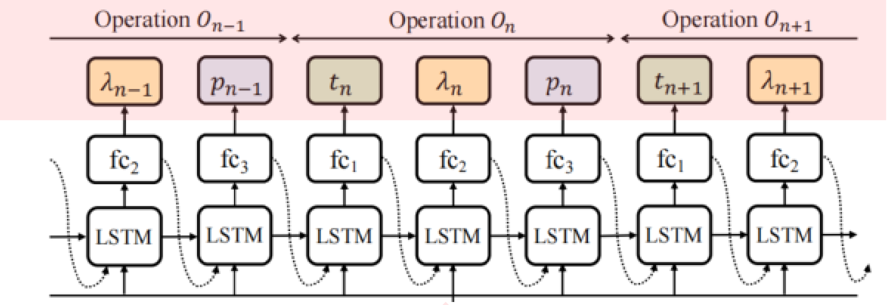


Solution design

- RL agent design: a controller RNN
  - One-layer LSTM (with 100 hidden units)
  - DA policy $\tau = \{a_1, \ldots a_T\} = \{t_1, \lambda_1, p_1, \ldots, t_N, \lambda_N, p_N\}$

- Challenges
  - Challenge 1: Representation of the *state*
  - Challenge 2: Execution of DA policy for the four different types of updating strategies F_{up}

2) F_{up} in the online BLF schemes have varying requirements regarding data size and diversity



1) the temporal dynamics of the observed data stream should be extracted

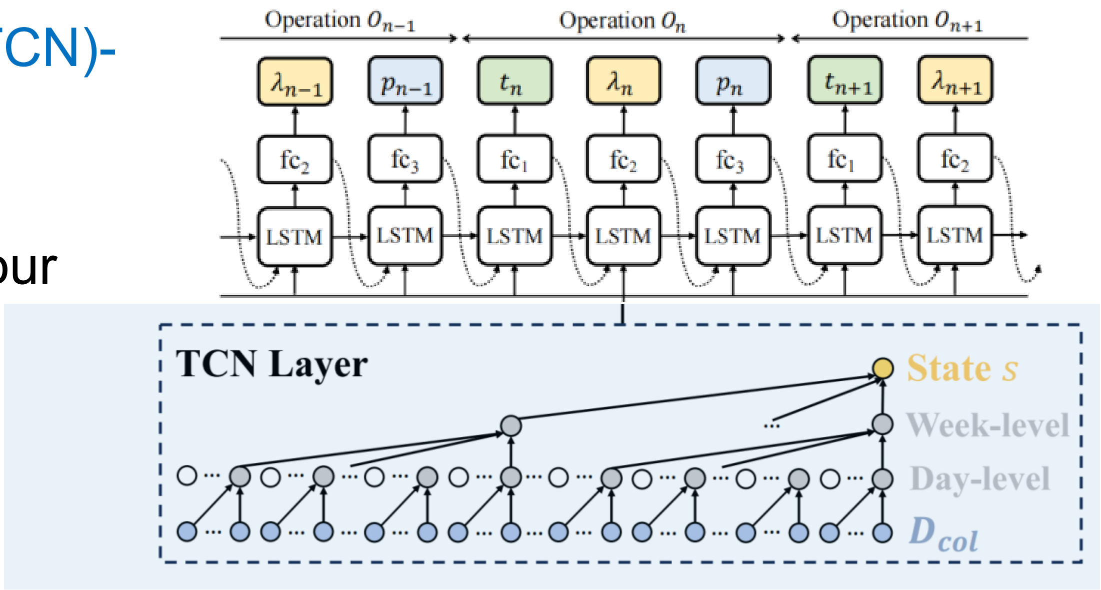
Solution design

- RL agent design: a controller RNN
  - One-layer LSTM (with 100 hidden units)
  - DA policy $\tau = \{ a_1, \ldots a_T \} = \{ t_1, \lambda_1, p_1, \ldots, t_N, \lambda_N, p_N \}$

$$D_\tau^j = \cup_{u=1}^{U} \cup_{v=1}^{V} \tau_u(D_{up}^j)$$

$$a_t : \begin{cases} = \mathrm{argmax}(\mathrm{softmax}(\mathrm{fc}_i(h_t))), & \text{// less diversity} \\ \sim \mathrm{Categorical}(\mathrm{softmax}(\mathrm{fc}_i(h_t))), & \text{// greater diversity} \end{cases}$$

- Challenges
  - Challenge 1: Representation of the *state*
  - Solution 1: temporal convolutional network (TCN)-based embedding
  - Challenge 2: Execution of DA policy for the four different types of updating strategies
  - Solution 2: Adaptable data transformation



RL training

- Applying PPO

Algorithm 1: Training design of AugPlug.

Input: The building dataset $\{\mathcal{D}\}$. The BLF model M and its update strategy $\mathcal{F}_{up}$.
Output: The controller π_θ.

```
Initialize D_train ← ∅;
for D ∈ {D} do
    Obtain samples {(M_j, D^j_col, D^j_up)} by deploying M on D;
    D_train ← D_train ∪ {(M_j, D^j_col, D^j_up)};
for i = 1, ..., #Episodes do
    for (M_j, D^j_col, D^j_up) ∈ D_train do
        /* Step 1: prepare updating set */
        Obtain state s through embedding D^j_col with TCN;
        {τ_u}^U_{u=1} ← π_θ(s);
        D^j_τ ← ∪^U_{u=1} ∪^V_{v=1} τ_u(D^j_up);          // Eq. 6
        /* Step 2: update BLF model */
        M_{j+1} ← F_up(M_j, D^j_up) ;
        M^τ_{j+1} ← F_up(M_j, D^j_τ);
        R ← ACC_val(M^τ_{j+1}) − ACC_val(M_{j+1});
        /* Step 3: update the RL agent */
        θ ← θ − η∇_θ L_PPO(θ);                              // Eq. 7
```

Where: line 1 is $\mathcal{D}_{train} \leftarrow \emptyset$; line 4 is $\mathcal{D}_{train} \leftarrow \mathcal{D}_{train} \cup \{(M_j, D^j_{col}, D^j_{up})\}$; line 8 is $\{\tau_u\}_{u=1}^{U} \leftarrow \pi_\theta(s)$; line 9 is $D^j_\tau \leftarrow \cup_{u=1}^{U} \cup_{v=1}^{V} \tau_u(D^j_{up})$; line 10 is $M_{j+1} \leftarrow \mathcal{F}_{up}(M_j, D^j_{up})$; line 11 is $M^\tau_{j+1} \leftarrow \mathcal{F}_{up}(M_j, D^j_\tau)$; line 12 is $R \leftarrow \mathcal{ACC}_{val}(M^\tau_{j+1}) - \mathcal{ACC}_{val}(M_{j+1})$; line 13 is $\theta \leftarrow \theta - \eta\nabla_\theta \mathcal{L}_{PPO}(\theta)$.

AutoDA model adoption

(The additional codes needed: highlight in red)

- The input features of BLF model:
  - (1) mechanical features, e.g., history power
  - (2) meteorological features, e.g., outdoor temperature;
  - (3) and time features
- Integrating to the existing online BLF schemes
  - Little efforts needed to equip an online BLF scheme with the proposed AutoDA model.

Adaptation \ Learning mode	Retrain	Fine-tune
Periodically	SVM (Energy Reports. 2023) Ensemble (SUSTAIN ENERGY GRIDS. 2021)	HMM (Trans.Power Syst. 2021) LSTM (Applied Energy. 2021)
Triggered	Random Forest (Sustainability. 2022) KNN (Sensors. 2020)	Autoencoder (Applied Energy. 2022) RNN (Applied Energy. 2020)

The file of **inference.py (from RF[28])**

```
import AugPlug
import numpy, sklearn, ...
err_threshold = 0.25  # follow this paper
def get_raw_data(): ...
def data_preprocessing(): ...
def predict(): ...
def cal_WAPE():...
# F_up function (trigger + retrain)
def triggered_detection(pred, data_stream):
    error = cal_WAPE(pred, data_stream)
    return (error > err_threshold)
def retrain(updating_set):
  LF_model = KNeighborsRegressor(n_neighbors=3)
  LF_model.fit(updating_set)
  return LF_model

if __name__ == '__main__':
  with open('LF_model.pkl', 'rb') as r:
     LF_model = pickle.load(r)
  pred_list = []
  # LF prediction and online learning
  while raw_data := get_raw_data():
    data_stream = data_preprocessing()
    pred = predict(LF_model, data_stream)
    if len(pred_list) < 30*24:
      continue
    if triggered_detection(pred_list[-1], data_stream):
      # original updating set
      updating_set = data_stream[-30*24: ]
      updating_set = AugPlug(updating_set,
                                   raw_data, err)
      LF_model = retrain(updating_set)
    pred_list += pred
```

The file of **inference.py (from LSTM[12])**

```
import AugPlug
import numpy, torch, ...
def get_raw_data(): ...
def preprocessing(): ...
def predict(): ...
class buffer_mechanism(): ...
# F_up function (periodically + fine-tune)
def update(model, updating_set):
  for epoch in range(total_iters):
    for x, y in updating_set:
      loss = criterion(model(x), y)
      optimizer.zero_grad()
      loss.backward()
      optimizer.step()

if __name__ == '__main__':
  model = torch.load('BLF_model.pt')
  buffer = buffer_mechanism()
  # BLF model forecasts then updates
  while raw_data := get_raw_data():
    inference_set = preprocessing(raw_data)
    pred, err = predict(model, inference_set)
    buffer.append_data(inference_set)
    # original updating set
    updating_set = buffer.output()
    # replace with augmented updating set
    updating_set = AugPlug(updating_set,
                            raw_data, err)
    update(model, updating_set)
```

Evaluation

- Online BLF schemes & Building Datasets
  - Four online BLF schemes: BLF_1 to BLF_4
  - 500+ buildings (two-year length)
- Metrics
  - BLF accuracy: CV-RMSE
  - Online A/B testing metrics: the proportion of the ineffective updates conducted
- Baselines:
  - 1) Generative model: TimeGAN [1], to generates more diverse time-series data.
  - 2) Concept drift adaptation method: DDGDA [2], directly forecasts the future data distribution.

Adaptation \ Learning mode	Retrain	Fine-tune
Periodically	SVM (Energy Reports. 2023) Ensemble (SUSTAIN ENERGY GRIDS. 2021)	HMM (Trans.Power Syst. 2021) LSTM (Applied Energy. 2021)
Triggered	Random Forest (Sustainability. 2022) KNN (Sensors. 2020)	Autoencoder (Applied Energy. 2022) RNN (Applied Energy. 2020)

[1] Time-series generative adversarial networks, NeurIPS, 2019
[2] DDG-DA: Data Distribution Generation for Predictable Concept Drift Adaptation, AAAI, 2022

Evaluation

- Overall performance
  - The accuracy improvement is 29% for the tested online BLF schemes.
  - Reduce the ratio of ineffective update from 31% to 17%

Table 5: The CVRMSE (lower is better) of the day ahead load forecasting results on 15 buildings. Comparisons across the default online BLF scheme and the scheme equipped with AugPlug and the baselines.

Online BLF models	**Methods**	*Education*			*Public*			*Assembly*			*Office*			*Lodging*		
		B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_{12}	B_{13}	B_{14}	B_{15}
BLF_1	Original	71.48	33.93	32.11	43.53	34.23	49.19	60.27	26.82	43.95	42.97	38.17	44.48	49.75	34.89	40.77
	+ TimeGAN	65.13	37.05	27.75	44.88	35.97	45.31	52.11	27.27	44.89	41.36	34.63	42.85	48.12	38.01	42.66
	+ DDG-DA	53.35	27.43	24.13	39.22	26.81	41.19	46.75	23.53	37.18	38.93	35.07	37.66	43.61	**28.92**	39.34
	+ ***AugPlug***	**41.47**	**22.86**	**23.35**	**33.57**	**23.31**	**36.77**	**33.77**	**18.88**	**32.59**	**28.83**	**25.02**	**31.09**	**31.92**	29.58	**29.43**
BLF_2	Original	54.13	18.53	21.78	23.65	22.01	38.25	46.13	15.03	29.18	28.42	24.86	27.41	36.39	23.24	25.07
	+ TimeGAN	50.24	22.14	19.06	22.31	22.33	36.11	46.94	14.44	26.02	27.55	22.45	26.16	35.32	31.17	26.47
	+ DDG-DA	37.38	16.03	**18.22**	17.35	18.03	31.29	45.29	13.76	27.45	28.54	24.87	23.98	31.22	21.03	25.08
	+ ***AugPlug***	**27.55**	**14.87**	18.71	**16.83**	**15.41**	**28.44**	**32.27**	**13.51**	**23.65**	**25.28**	**18.06**	**23.09**	**19.67**	**16.71**	**20.99**
BLF_3	Original	71.26	35.81	31.26	43.81	36.26	52.41	58.66	33.54	44.59	43.85	42.82	47.31	50.06	38.08	43.47
	+ TimeGAN	56.12	39.12	26.85	40.81	40.54	54.79	51.35	32.14	47.47	40.76	38.44	43.89	48.59	43.78	43.67
	+ DDG-DA	49.34	30.38	**22.74**	40.38	36.42	48.26	43.18	29.93	41.24	35.11	34.51	38.08	47.66	36.86	43.74
	+ ***AugPlug***	**36.22**	**24.27**	24.59	**34.59**	**26.41**	**41.21**	**39.78**	**26.13**	**34.35**	**32.64**	**29.56**	**35.63**	**34.46**	**32.73**	**32.09**
BLF_4	Original	66.52	31.49	22.96	27.58	31.52	26.35	61.62	28.11	32.67	29.42	23.13	24.15	52.52	30.18	42.01
	+ TimeGAN	48.81	30.28	45.89	33.38	29.86	28.92	41.52	22.78	28.36	34.05	26.88	24.06	40.87	30.56	40.78
	+ DDG-DA	44.98	28.26	24.73	26.07	25.89	21.64	37.01	21.72	26.09	30.86	25.14	21.39	30.22	22.17	36.77
	+ ***AugPlug***	**20.12**	**16.39**	**20.31**	**17.21**	**25.32**	**13.44**	**32.38**	**18.16**	**19.58**	**26.28**	**17.54**	**20.23**	**25.12**	**13.17**	**20.33**

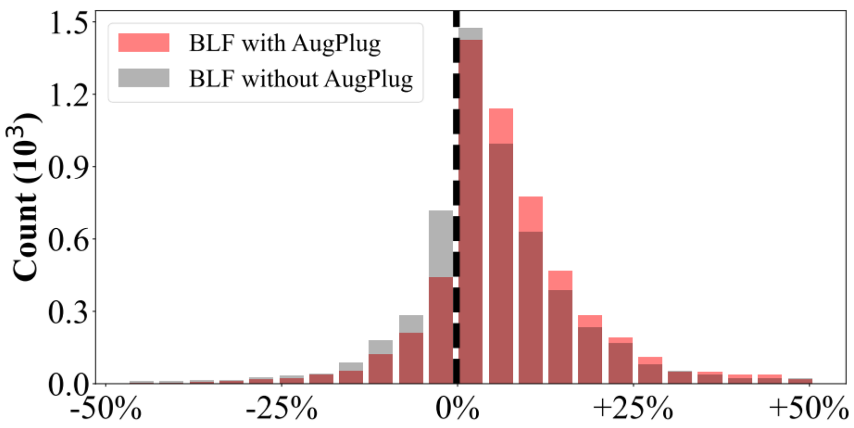
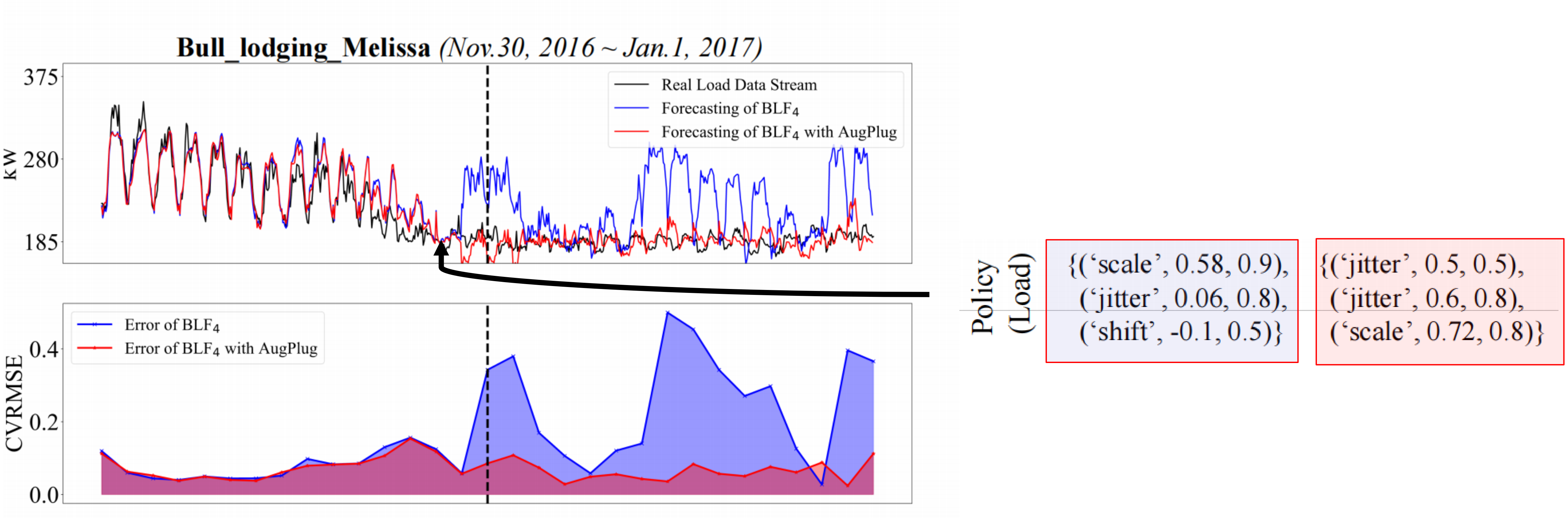


Figure 8: Accuracy improvement (with vs without AugPlug).

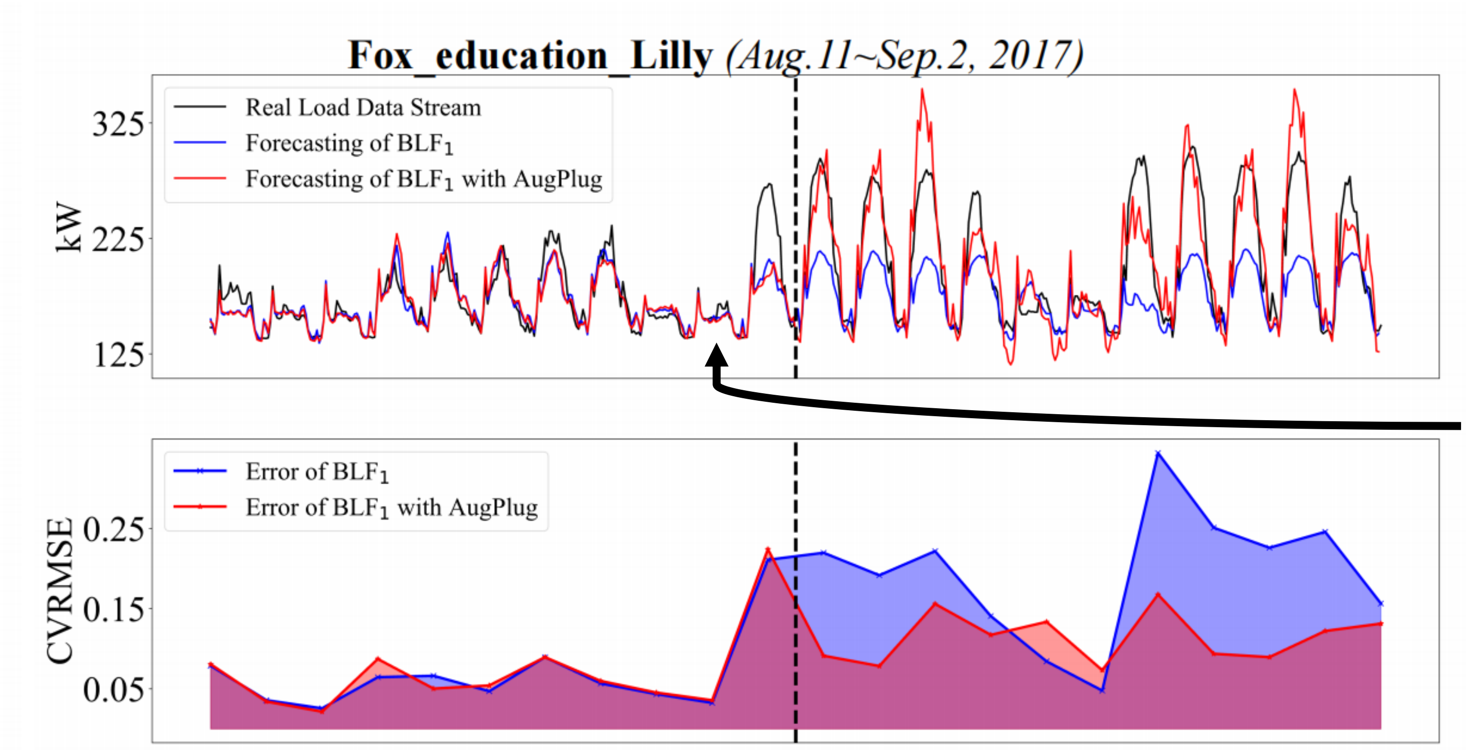
Evaluation

- Data augmentation policy analysis
  - Case 1: for BLF scheme 4 (Autoencoder (Applied Energy. 2022)), univariate input feature (load)



Evaluation

- Data augmentation policy analysis
  - Case 2: for BLF scheme 1 (SVM (Energy Reports. 2023)), Two input features (historical load + outdoor temperature)



Policy (Load)

{('scale', 1.4, 0.6), ('smooth', 7, 0.8), ('scale', 0.86, 0.4)}
{('scale', 1.8, 0.7), ('smooth', 3, 0.5), ('smooth', 9, 0.9)}

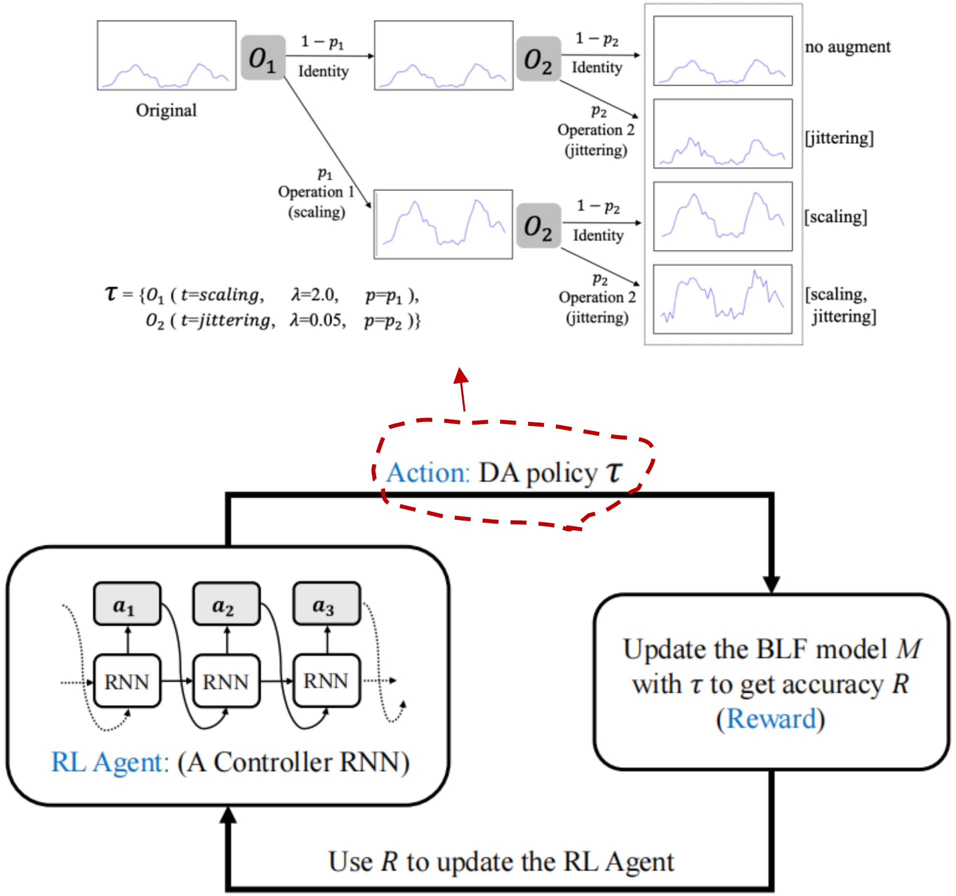
Policy (Temp.)

{('shift', 0.3, 0.8), ('smooth', 2, 0.3), ('scale', 1.4, 0.8)}
{('shift', 0.3, 0.8), ('scale', 0.86, 0.4), ('shift', 0.1, 0.5)}

Summary

- We investigate the effectiveness of online updates in existing online BLF schemes, and we show a significant proportion of updates have negative effects.

- We introduce the framework of *AutoDA (automated data augmentation)*, based on which to develop a DA model to search the suitable data augmentation policies.



AugPlug: An Automated Data Augmentation Model to Enhance Online Building Load Forecasting

Thank you!
Q&A

Yang Deng, Rui Liang, Yaohui Liu, Jiaqi Fan, and Dan Wang
Department of Computing,
The Hong Kong Polytechnic University