Online Learning for Energy Consumption Forecasting in Heavy-Duty Electric Vehicles

Yuantao Fan[1,*,†], Zhenkan Wang[2,†] and Sławomir Nowaczyk[1]

[1]Center for Applied Intelligent Systems Research (CAISR), Halmstad University, Kristian IV:s väg 3, 301 18 Halmstad, Sweden
[2]Volvo Group Trucks Technology and Industrial Division, Götaverksgatan 10, 417 55 Göteborg, Göteborg, Sweden

Abstract

Accurate forecasting of auxiliary energy consumption in heavy-duty battery electric vehicles (HD-BEVs) is critical for energy-efficient operation, route planning, and charging optimization. However, varying driving conditions, payloads, and environmental factors cause concept drift, reducing the reliability of static, offline-trained models. To address this challenge, this paper presents a tailor-made online learning framework capable of continuously updating forecasting models in real time as new trip data becomes available. The proposed approach adapts to shifting consumption patterns while remaining computationally efficient for on-vehicle edge deployment. Experimental results on real-world HD-BEV data show that the proposed online learning strategy significantly outperform batch models in both accuracy and adaptability. Moreover, the proposed strategy achieves an effective trade-off between performance and computational cost, making it well-suited for real-time deployment in resource-limited environments.

Keywords

Online Learning, Energy Consumption Forecasting, Heavy-duty Battery Electric Trucks

1. Introduction

Optimizing the operation of electric commercial heavy-duty vehicles requires accurate energy consumption prediction in order to, for example, plan the route in the most efficient way and determine the best time and location for charging. Commercial vehicles operate across a wide range of scenarios and perform diverse types of tasks. In transportation operations, variations in ambient conditions, driver behavior, and route characteristics influence not only vehicle performance but also the internal dynamics of the driveline and auxiliary systems. The number of factors impacting key subsystems such as heaters, air compressors, and energy converters is extensive and often highly variable. Given this high degree of variability, AI systems designed to forecast energy consumption must be capable of adapting to dynamic and evolving conditions. This requires robust handling of concept drift, where the statistical properties of the features, or relationships between input features and the target variable change over time or across different sub-populations. To maintain forecasting accuracy in the presence of evolving data distributions, models must either be continuously updated through adaptive mechanisms or designed to be inherently robust to such heterogeneity. Moreover, delayed label availability often postpones the online learning process, particularly when longer forecasting horizons are employed, thereby hindering timely model adaptation. Last but not least, on-board computational resources are limited, which constrains the frequency and complexity of model updates that can be executed in real time. These challenges highlight the necessity for an adaptive, cost-efficient online learning framework capable of maintaining forecasting accuracy under practical resource constraints.

It is of interest to apply and evaluate the performance of online learning algorithms across various forecasting scenarios, particularly under conditions involving delayed label availability, where the ground truth is not immediately available for incremental model update. In industrial applications involving fleets of equipment, multi-stream learning presents a promising approach, where models are

1st Streaming Continual Learning Bridge at AAAI26, January 21, 2026, Singapore.
*Corresponding author.
†These authors contributed equally.
yuantao.fan@hh.se (Y. Fan); zhenkan.wang@volvo.com (Z. Wang); slawomir.nowaczyk@hh.se (S. Nowaczyk)
0000-0002-3034-6630 (Y. Fan); 0009-0007-7976-2874 (Z. Wang); 0000-0002-7796-5201 (S. Nowaczyk)

trained on multiple data streams, from different units or trip segments, and optimized to forecasting on a specific equipment. Last but not least, establishing a benchmark framework to compare the performance of batch versus online learners in this domain is of great interesting for guiding the selection and deployment of such online learning systems for real-world applications in forecasting energy consumptions.

This paper investigates online learning configurations and strategies for forecasting auxiliary energy consumption in HD-BEVs. The contributions of this paper including: i) a tailor-made online learning strategy for forecasting auxiliary energy consumption in HD-BEVs, aiming at adaptively update the model based on the streaming observations, while accounting for the computational constraints performing on, e.g., edge devices; and ii) an evaluation and comparison of multiple online learning configurations is conducted in terms of predictive accuracy and computational efficiency, using a real-world HD-BEV dataset collected from in-service operations. Moreover, several promising research directions are proposed for forecasting energy consumption under resource-constrained conditions.

2. Related Work

Online time series forecasting [1, 2, 3, 4] is challenging due to streaming data and concept drift. Deep neural network based forecasters struggle to adapt to evolving environments while retaining prior knowledge [3, 4]. FSNet [3] addresses this by using per-layer adapters, which map contexts to transformation coefficients stored in associative memory for fast adaptation. Recent work emphasizes exploiting both temporal and cross-variable dependencies under a unified framework [4]. OneNet [4] builds independent models for these dependencies and combines them using Online Convex Programming (OCP), leveraging exponential gradient descent for long-term history and offline reinforcement learning for short-term adjustment. Empirical evaluations show OneNet outperforms FSNet and Informer [5] in online multivariate time series forecasting [4].

Streaming regressors are also used for data stream forecasting [6]. Many methods are discussed in [7]. Hoeffding Tree Regressor (HTR) adapts Hoeffding Tree (HT) for regression using Hoeffding bounds [8] and ADaptive sliding WINdow (ADWIN)[9] for drift detection. Fast Incremental Regression Trees with Drift Detection (FIRT-DD) uses variance-reduction split criteria and the Page-Hinckley test[10] for drift detection. In the above methods, upon drift detection, a background tree replaces the foreground tree. Ensemble methods such as Adaptive Random Forest Regressor (ArfReg)[11] and Self-Optimising K-Nearest Leaves (SOKNL)[12] outperform single-tree models [13]. SOKNL combines k Nearest Neighbors and ArfReg. Only the k trees with the smallest distances between the input instance and the centroid of the relevant leaf are used for the prediction. Both methods use FIRT-DD as base learners. ArfReg use ADWIN for drift detection and adaptation, while SOKNL contains explicit adaptation via centroid updates [12]. It is of interest to build a tailor-made online learning strategy for HD-BEVs, with light-weight and generic models (e.g. neural networks of a limited size) that can be configured and deployed on edge devices mounted on-board vehicle.

3. Problem Formulation

In this study, we consider the task of energy consumption forecasting for EVs using online learning. Let $\mathcal{X} \subseteq \mathbb{R}^d$ denote the input feature space (e.g., vehicle speed, ambient temperature, payload etc.), and $\mathcal{Y} \subseteq \mathbb{R}$ represent energy consumption or power output over a prediction horizon τ_{PH}.

A neural network parameterized by θ is used as the predictive model (regressor) $f_\theta : \mathcal{X} \to \mathbb{R}$, mapping the incoming streaming sample to the energy consumption in a future period of time τ_{PH}. Each trip is indexed by $m = 1, \ldots, M$, and provides a sequence of time-ordered samples $\mathcal{S}_m = \{(x_{m,t}, y_{m,t})\}_{t=1}^{T_m}$ where $x_{m,t} \in \mathcal{X}$ and $y_{m,t} \in \mathcal{Y}$ denote the input feature vector and the observed energy consumption at time step t. The objective is to design an online learning strategy that incrementally updates a regression model f_θ as streaming observations x_t arrive over time. The model parameters are updated based on a loss function $\ell(\hat{y}_t, y_t)$ computed using the true label $y_{t+\tau_{PH}}$, which becomes available after $t + \tau_{PH}$.

4. Online Learning for Energy Consumption Forecasting

In this study, we develop an online learning strategy specifically tailored for forecasting the energy consumption of auxiliary systems in HD-BEVs. Empirical observations indicate that, for most trips, the auxiliary energy consumption tends to be higher with more uncertainty during the initial phase of the journey, before stabilizing as the trip progresses. The proposed energy consumption forecasting framework consists of the following stages: i) training an initial model f_θ in a batch setting using historical datasets collected from in-service vehicle operations; ii) employing f_θ to forecast the auxiliary energy consumption over a future prediction horizon τ_{PH}; and iii) compute the loss $\ell(\hat{y}_t, y_t)$ once the true label become available and subsequently updating the regressor f_θ in an adaptive manner.

The proposed adaptive online learning strategy, specifically designed for this application, is based on the following mechanisms/principles: i) training activation: learning is triggered for P epochs whenever the prediction error exceeds a predefined threshold ξ, which is determined empirically by selecting a value corresponding to a certain upper percentile of the distribution of historical training losses; ii) Adaptive learning rate: a weighting function $w(t)$ (see candidate functions in Figure 1) is applied to adjust the learning rate dynamically according to $\eta_t = \eta_0 \cdot w(t)$, where η_0 denotes the initial learning rate and t represents the number of instances since the learning mechanism was last triggered; iii) Enhanced learning in early stage: during the initial phase of each trip, when the auxiliary systems exhibit higher and more variable energy consumption due to vehicle initialization, the number of training epochs P is doubled to promote rapid model adaptation; and iv) Stabilized learning phase: once the vehicle enters a more stable consumption regime, the model is updated less frequently, using a buffer containing the most recent Q samples to maintain adaptation while reducing computational cost. Pseudo-code of the proposed learning strategy is available in Algorithm 1.

Prequential evaluation was employed to assess and validate the performance of the proposed online learning strategies. Standard regression metrics, including the mean absolute error (MAE) and mean squared error (MSE), were utilized for quantitative comparison. Furthermore, given that the overall objective is to estimate the vehicle's total auxiliary energy consumption for the remainder of each trip, the accumulated error (AccErr) metric was introduced to evaluate deviations in aggregated consumption across the entire trip, defined as $AccErr = \frac{1}{M} \sum_{i \in M} | \sum_{j=1}^{T_M} \frac{y_j}{T_M} - \sum_{j=1}^{T_M} \frac{\hat{y}_j}{T_M} |$, where T_M denotes the number of instances within the M-th trip. Finally, the cost-efficiency of the proposed online learning framework is analyzed by examining the trade-off between predictive accuracy and computational resource usage, e.g., the CPU time required.

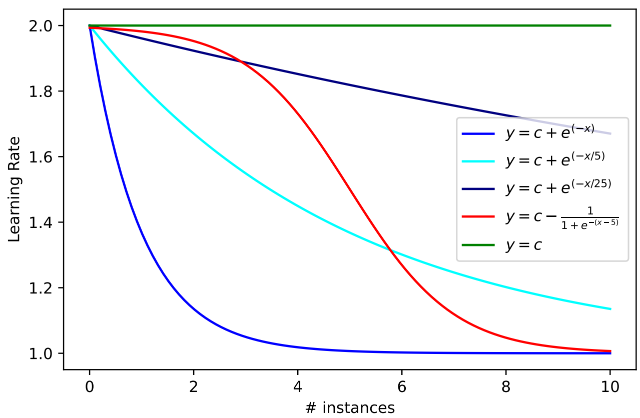


Figure 1: learning rate weighting function over instances when triggered

Algorithm 1 Online Learning for Auxiliary Energy Consumption Forecasting of HD-BEVs

Input: Historical time-series data $\{(x_{tr}, y_{tr})\}_{t=1}^{T}$ from previous trips; Streaming time-series data $\{x_t\}_{t=1}^{T}$ with delayed labels $\{x_t\}_{t=1}^{T}$, available at $t+1$
Parameter: MLP regressor f_θ, with parameters θ; adaptive weighting function $w(t)$; K cycle for reaching a steady consumption state; default
Output: Updated model parameters θ_T and consumption forecasts $\{\hat{y}\}_{t=1}^{T}$

Initialization: Train base model f_{θ_0} offline on historical data in a traditional batch-learning setting
Set initial time step $t = 0$, threshold ξ for loss, and learning rate $\eta(t)$
for each new trip $i = 1, 2, \ldots, N$ **do**
  Reset local time index $t = 0$
  while new data point x_t arrives **do**
    Predict energy consumption $\hat{y}_t = f_\theta(x_t)$
    if label y_t has arrived for x_t **then**
      for $j = 1, 2, \ldots, P$ epochs **do**
        Update forecast $\hat{y}_t$, and compute instantaneous loss $L_t = \ell(\hat{y}_t, y_t)$
        if $t > K$ **then**
          Reduce update frequency
        end if
        if $L_t > \xi$ **then**
          Compute weighted learning rate $\eta_t = \eta_0 \cdot w(t)$
          Update model parameters: $\theta \leftarrow \theta - \eta_t \nabla_\theta L_t$
        end if
      end for
    end if
    Increment time: $t \leftarrow t + 1$
  end while
  Store updated parameters θ_i for next trip initialization
end for
return consumption forecasts $\{\hat{y}\}_{t=1}^{T}$ and parameters θ_T at time T

5. Experiment Results

For the experiments conducted in this exploratory study, a four-layer feedforward neural network was employed for the forecasting task, consisting of hidden layers with 128, 64, 32, and 16 neurons, respectively; MSE was employed as the loss function for training the network; Adam optimizer was selected as the optimizer with a learning rate of 0.008, determined through hyper-parameter tuning; the model was implemented in PyTorch, and all experiments were executed on a server equipped with an NVIDIA Tesla V100.

5.1. HD-BEVs Dataset

The dataset consists of signals transmitted through the controller area network within a heavy-duty battery electric truck, including parameters such as speed, acceleration, road inclination and other vehicle operating signals, from its delivery operations over a few weeks. The battery capacity of the truck is 540 kWh (including 6 modules) by Lithium Nickel-Cobalt-Aluminum Oxide (NCA) technology. The time-series data were segmented into trips from in-service delivery tasks, based on expert suggested rules. The segmented trips represent individual journeys partitioned into distinct segments resulting from interruptions such as vehicle stops, route alterations, or discontinuities in data logging process. Only segments corresponding to the driving mode are retained for analysis, defined as periods during which the vehicle's batteries are discharging and it is not connected to an external charging source.

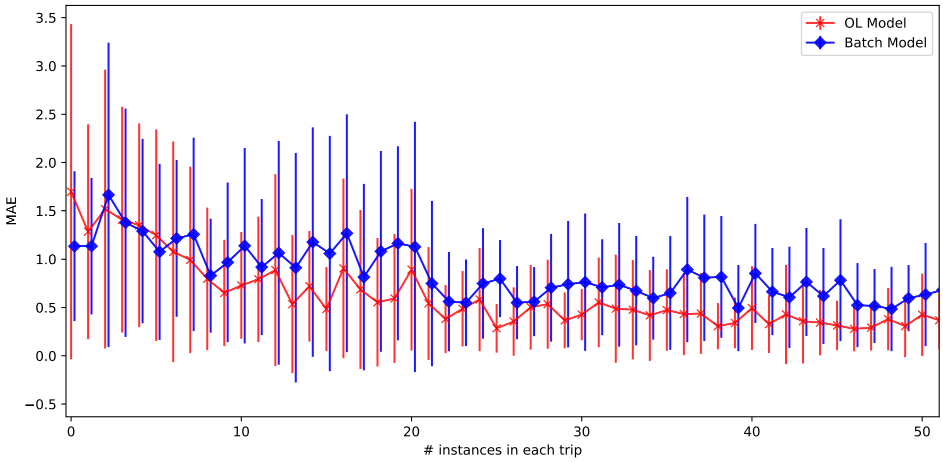


Figure 2: MAE vs. numbers of instances within each trip; error bars corresponding to standard deviations of errors from all trips in the testing set; forecasting errors are higher during the initial periods compared to later stages of the trips.

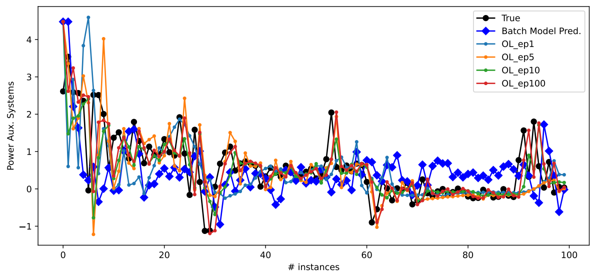
5.2. Comparison Forecasting Sequences

Figure. 3 presents a comparison of forecasting sequences obtained under different online learning settings from four trips, demonstrating how online training epochs and strategies influence the model's ability to learn and generalize in dynamic conditions. Figures. 3(a), 3(c), 3(e), and 3(g) illustrate the performance of online learning when trained with varying numbers of epochs per update step (e.g., 1, 5, 10, and 100). Across all trips, the general trend shows that the increasing number of epochs, resulted in smother and more stable predictions compared to batch-trained model. The first three trips (i.e., A to C) illustrated high power consumption at the beginning of each trip. In these cases, the online learning models were updated to capture the sharp initial variations, with some delay in the adaptation, whereas the batch-trained model fail to produce sound forecasts, and performs worse compared to the online learning methods. Conversely, Trip D displays relatively low consumption at the start, representing a unique case amongst the trips, the online methods were able to adapt in a few epochs but the batch model fail to produce reliable predictions. It is observed in Figure. 2, for both online learning methods and the batch model produce higher errors the first 20 instances, while the error of the subsequent samples (in which the consumption stabilizes) were lower. Figures 3(b, d, f, h) illustrates the forecasting sequences of different online learning configurations, including variations in buffer size, learning rate adaptation, and exponential weighting. The results indicate that models update with adaptive learning strategies achieve better tracking of short-term variations in the consumption.

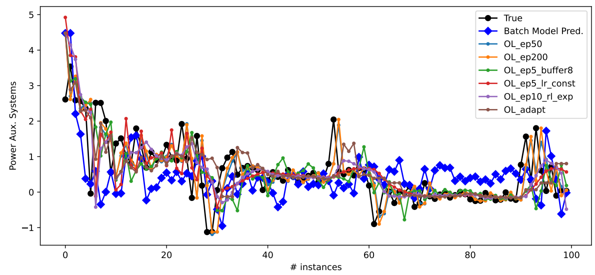
5.3. Performance Comparison

Table 1 summarizes the performance of various online learning configurations in terms of MAE, MSE, AccErr, and CPU Time. The results shows the trade-offs between predictive accuracy and computational efficiency w.r.t. training and inference time.

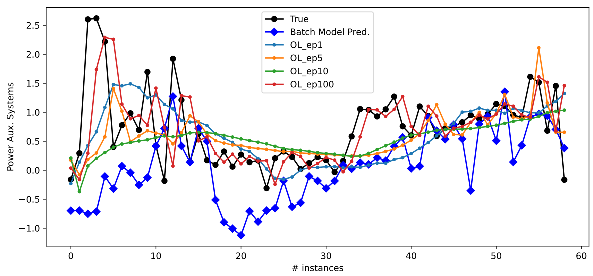
The baseline batch-trained model exhibits the highest MAE and MSE, indicating limited capability to handle evolving data streams. In contrast, all online learning configurations achieve notable improvements across all accuracy metrics, with MAE reductions of up to 30% compared to the batch model. Increasing the number of training epochs generally enhances prediction accuracy but with significantly higher CPU time. The experiments with buffer-based and adaptive learning rate strategies demonstrate comparable accuracy to OL models with different training epochs but with much lower CPU times, indicating better efficiency for real-time applications. The proposed ECF strategy (i.e. OL_ECF) allocates more computational resources for online learning during the initial phase; afterwards, the training was carried out for every fours incoming samples, with a reduced number of iterations and a small buffer



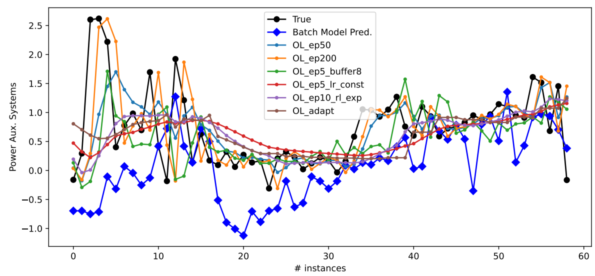
(a) Trip A: online learning with different epochs



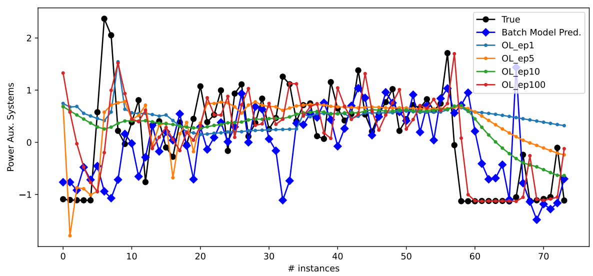
(b) Trip A: online learning with different settings



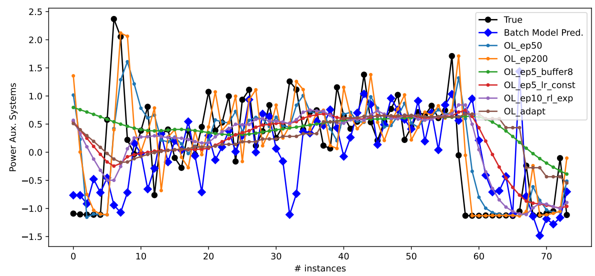
(c) Trip B: online learning with different epochs



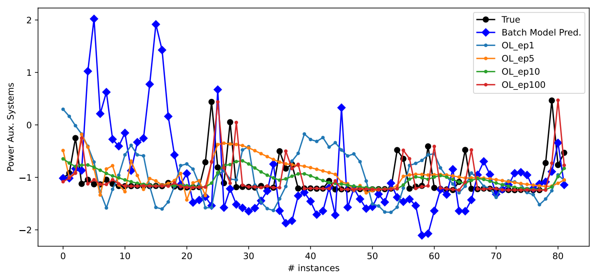
(d) Trip B: online learning with different settings



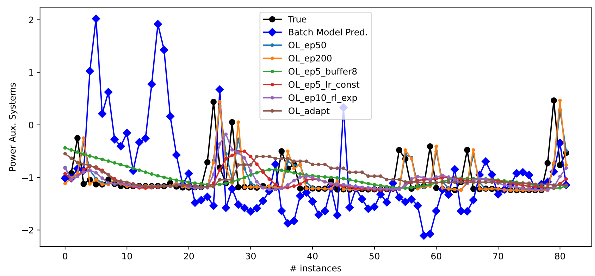
(e) Trip C: online learning with different epochs



(f) Trip C: online learning with different settings



(g) Trip D: online learning with different epochs



(h) Trip S: online learning with different settings

Figure 3: Comparison of forecasting sequences with different online learning settings/strategies; enabling online learning consistently improves the forecasting performance

of four samples. Among all tested variants, the proposed approach achieves a balanced performance, with low MAE, MSE and AccErr, and moderate computational cost (23.52 ± 0.95s), significantly lower than OL model trained with 10 epochs per instance. Figure. 4 illustrates the trade-off between accuracy (MAE) and computational cost (CPU time) across different online learning configurations. The results show a general trend where the number of training epochs improves accuracy but at the expense of higher computation time. The proposed approach (i.e., OL_ECF) attains good accuracy while being computationally efficient, compared to other configurations.

Online Learning Config.	MAE	MSE	AccErr	CPUTime
N/A Batch Model	0.76 ± 0.03	1.10 ± 0.07	0.29 ± 0.04	1.38 ± 0.41
OL ep1	0.62 ± 0.03	0.84 ± 0.05	0.2 ± 0.04	6.79 ± 0.75
OL ep5	0.6 ± 0.02	0.81 ± 0.04	0.16 ± 0.03	19.47 ± 2.41
OL ep10	0.55 ± 0.01	0.73 ± 0.02	0.09 ± 0.0	34.64 ± 0.5
OL ep100	0.54 ± 0.0	0.66 ± 0.0	0.03 ± 0.0	303.59 ± 7.94
OL ep5 buffer4	0.59 ± 0.01	0.81 ± 0.03	0.13 ± 0.02	16.81 ± 0.64
OL ep5 buffer64	0.64 ± 0.03	0.88 ± 0.1	0.13 ± 0.01	17.73 ± 0.5
OL ep10 buffer4	0.57 ± 0.0	0.77 ± 0.0	0.05 ± 0.0	32.9 ± 1.01
OL ep10 buffer64	0.66 ± 0.0	0.9 ± 0.01	0.22 ± 0.01	38.75 ± 2.3
OL ep5 lr const	0.56 ± 0.0	0.78 ± 0.02	0.06 ± 0.0	20.05 ± 0.8
OL ep5 lr decay exp	0.56 ± 0.0	0.78 ± 0.03	0.06 ± 0.01	20.66 ± 1.09
OL ep5 lr decay exp div5	0.56 ± 0.0	0.77 ± 0.03	0.06 ± 0.01	20.61 ± 1.06
OL ep5 lr decay exp div25	0.56 ± 0.0	0.77 ± 0.03	0.06 ± 0.0	20.74 ± 1.09
OL ep5 lr decay logists	0.56 ± 0.0	0.77 ± 0.02	0.06 ± 0.0	20.49 ± 1.19
OL ep10 lr const	0.54 ± 0.0	0.77 ± 0.12	0.05 ± 0.01	36.22 ± 1.56
OL ep10 lr decay exp	0.56 ± 0.06	7.13 ± 19.15	0.07 ± 0.04	36.77 ± 1.71
OL ep10 lr decay exp div5	0.55 ± 0.05	4.99 ± 15.93	0.06 ± 0.03	36.01 ± 1.93
OL ep10 lr decay exp div25	0.55 ± 0.04	3.95 ± 13.91	0.06 ± 0.03	35.9 ± 1.97
OL ep10 lr decay logists	0.55 ± 0.04	3.3 ± 12.51	0.06 ± 0.03	36.2 ± 2.06
OL ECF (proposed)	0.55 ± 0.01	0.76 ± 0.02	0.07 ± 0.01	23.52 ± 0.95

Table 1
Performance comparison on HD-BEVs dataset

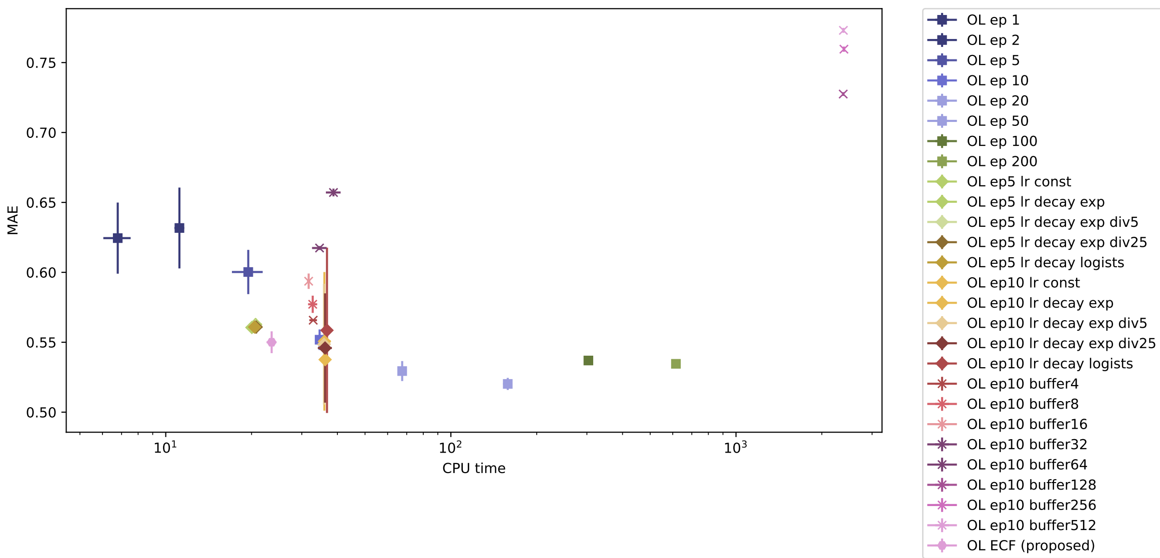


Figure 4: Performance and resource trade-offs, MAE vs. CPU time

5.4. Discussion

The online learning strategy achieves its best performance when the model is trained for at least 50 epochs and updated at every time step, resulting in a computational cost of more than 158 CPU time units. However, reducing the number of epochs to 5 and applying a learning rate decay function decreases the computational cost to approximately 20 CPU time units, with only a slight increase of about 3% in MAE error. Therefore, the proposed adaptive online learning strategy provides a substantial reduction in computation time with minimal loss in accuracy. Considering the trade-off between training time, computational resources, and accuracy improvement, our online learning strategy is to use 5 epochs for online training after initialization of the vehicle. In contrast, employing 50 epochs during the initial 10–20 time steps (or instances) is beneficial to ensure adequate model adaptation at the beginning of operation.

5.5. Conclusion and Future Work

This paper presented a exploratory study on online learning strategies for forecasting auxiliary energy consumption in HD-BEVs. The proposed framework enables adaptive model updates in response to streaming data, while accounting for the computational constraints of onboard edge devices. The experimental results demonstrated that the proposed online learning models substantially outperform the batch-trained baseline in both mean absolute error and adaptability across multiple driving trips.

Future research includes the following directions: i) developing adaptive mechanisms that dynamically adjust the training schedule (including model update frequency, buffer sampling strategies, and weighting functions) based on learning efficiency and data variability; ii) exploring alternative time-series forecasting models and advanced online learning techniques to further enhance predictive performance and adaptability; and iii) investigating hybrid online–offline learning paradigms that support long-term knowledge retention while maintaining rapid responsiveness to short-term operational dynamics.

Author Declaration on GenAI

During the preparation of this work, the authors used ChatGPT, Grammarly in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the authors reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] O. Anava, E. Hazan, S. Mannor, O. Shamir, Online learning for time series prediction, in: Conference on learning theory, PMLR, 2013, pp. 172–184.

[2] C. Liu, S. C. Hoi, P. Zhao, J. Sun, Online arima algorithms for time series prediction, in: Proceedings of the AAAI conference on artificial intelligence, volume 30, 2016.

[3] Q. Pham, C. Liu, D. Sahoo, S. Hoi, Learning fast and slow for online time series forecasting, in: The Eleventh International Conference on Learning Representations, 2023.

[4] Q. Wen, W. Chen, L. Sun, Z. Zhang, L. Wang, R. Jin, T. Tan, et al., Onenet: Enhancing time series forecasting models under concept drift by online ensembling, Advances in Neural Information Processing Systems 36 (2024).

[5] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: AAAI, volume 35, 2021, pp. 11106–11115.

[6] Y. Sun, H. M. Gomes, B. Pfahringer, A. Bifet, Real-time energy pricing in new zealand: An evolving stream analysis, arXiv preprint arXiv:2408.16187 (2024).

[7] A. Choudhary, P. Jha, A. Tiwari, N. Bharill, A brief survey on concept drifted data stream regression, SocProS (2021) 733–744.

[8] W. Hoeffding, Probability inequalities for sums of bounded random variables, The collected works of Wassily Hoeffding (1994) 409–426.

[9] A. Bifet, R. Gavalda, Learning from time-changing data with adaptive windowing, in: SIAM (SDM), SIAM, 2007, pp. 443–448.

[10] E. Ikonomovska, J. Gama, S. Džeroski, Learning model trees from evolving data streams, Data Min. Knowl. Discov. 23 (2011) 128–168.

[11] H. M. Gomes, J. P. Barddal, L. E. B. Ferreira, A. Bifet, Adaptive random forests for data stream regression., in: ESANN, 2018.

[12] Y. Sun, B. Pfahringer, H. M. Gomes, A. Bifet, Soknl: A novel way of integrating k-nearest neighbours with adaptive random forest regression for data streams, Data Min. Knowl. Discov. 36 (2022) 2006–2032.

[13] H. M. Gomes, J. Montiel, S. M. Mastelini, B. Pfahringer, A. Bifet, On ensemble techniques for data stream regression, in: 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–8.