Supplementary Materials For
"Boosted Multi-Task Learning for Inter-District Collaborative Load Forecasting"

Haizhou Liu, Xuan Zhang*, Hongbin Sun* and Mohammad Shahidehpour

The goal of the Supplementary Materials is to elaborate on two extended topics that are only briefly mentioned in the manuscript: the generalizability of the proposed framework to other Machine Learning (ML) algorithms, and the load forecasting curves on the test set of other districts.

A. Generalizability of the Framework to Other ML Algorithms

In the paper, we have demonstrated the generalizability of the framework to other distributions, especially the Gaussian Mixed Model (GMM). In this section, we intend to further demonstrate the generalizability of the framework to other ML algorithms.

Theoretically speaking, as long as the ML algorithm is boosted in nature like GBRTs, the proposed framework can be generalized to it. This is because the essence of the framework is to perform epoch-wise regressor training, regardless of the form of regressors. Of course, the built-in federated learning framework has to be seamlessly adapted to the new algorithm, which poses a new challenge for us.

We here select AdaBoost, a more classical ML algorithm, to demonstrate the generalizability of the framework. AdaBoost also works by iteratively boosting regressors, but the regressors employed are "decision stumps", threshold-based bifurcation functions based on one feature:

$$f(\boldsymbol{x}) = \begin{cases} y_L, & \boldsymbol{x}_i \leq \text{threshold} \\ y_R, & \boldsymbol{x}_i > \text{threshold} \end{cases} \tag{S1}$$

where f is the decision stump function; $\boldsymbol{x}$ denotes the input feature vector; y_L and y_R are the two possible output values for the decision stump; subscript i denotes the i-th feature.

Evidently, decision stumps are much weaker regressors than modern regression trees, so the prediction capability of the AdaBoost might also be inferior to that of GBRT (this is understandable as AdaBoost was invented much earlier). To maximally remedy for the weakness of decision stumps, AdaBoost dynamically adds weights to samples in each training epochs, such that samples with worse predictions are paid more attention to.

In order for the framework to be generalized to AdaBoost, we can also use the criterion of information gain to determine the optimal threshold for the decision stump, and the leaf-value formula to compute y_L and y_R. The different weights of samples can also be seamlessly integrated into the decision function by making adding weights directly to the first-order and second-order gradients:

$$\widetilde{g}_i^{\mu} \to w_i g_i^{\mu}, \widetilde{h}_i^{\mu} \to w_i h_i^{\mu}, \widetilde{g}_i^{\sigma'} \to w_i g_i^{\sigma'}, \widetilde{h}_i^{\sigma'} \to w_i h_i^{\sigma'} \tag{S2}$$

where g_i^μ and h_i^μ are the first-order and second-order gradients on the mean μ of sample i; $g_i^{\sigma'}$ and $h_i^{\sigma'}$ are the first-order and second-order gradients on the transformed standard deviation σ' of sample i. The tilde symbol $\sim$ indicates the corresponding weighted gradients.

As Equation (S2) cleverly incorporates weights into transformed gradients, the FederBoost framework can now be easily generalized to AdaBoost, which is elaborated in Framework S1 below:

Framework S1: Adapted Federated Learning Based on AdaBoost

Notations:

$\mathcal{B}$: the sample set belonging to bin b;
A_L, A_R: subsets of the sample set belonging to its left and right child node, respectively;
$\widetilde{g}_n^\mu$, $\widetilde{h}_n^\mu$: 1st- and 2nd-order weighted gradients of the NLL on μ;
$\mathcal{G}_\mathcal{B}^{\mu,i}$,$\mathcal{H}_\mathcal{B}^{\mu,i}$: 1st- and 2nd-order encrypted, weighted gradient sum of samples of District i in bin $\mathcal{B}$ on μ;
Enc, Dec: encryption and decryption algorithms;
I_k: the set of remaining districts in epoch k;
λ: the regularization parameter.

In each epoch *k*:

(Each district in I_k performs)
Compute $\mathcal{G}_\mathcal{B}^{\mu,i}, \mathcal{H}_\mathcal{B}^{\mu,i}$ for bins of local features.
Transfer $\mathcal{G}_\mathcal{B}^{\mu,i}, \mathcal{H}_\mathcal{B}^{\mu,i}$ to Active District.
(Active District performs)
Compute bin statistics $\sum_{n\in\mathcal{B}} \widetilde{g}_n^\mu, \sum_{n\in\mathcal{B}} \widetilde{h}_n^\mu$ for bins of local features.
Compute bin statistics $\sum_{n\in\mathcal{B}} \widetilde{g}_n^\mu, \sum_{n\in\mathcal{B}} \widetilde{h}_n^\mu$ for bins of global features.
Add up bin statistics into $\sum \widetilde{g}_n^\mu, \sum \widetilde{h}_n^\mu$.
For each candidate threshold condition on μ:
    Add up bin statistics into $\sum_{n\in A_L} \widetilde{g}_n^\mu, \sum_{n\in A_L} \widetilde{h}_n^\mu, \sum_{n\in A_R} \widetilde{g}_n^\mu, \sum_{n\in A_R} \widetilde{h}_n^\mu$ from bin statistics.
End For
Compute the optimal threshold condition with the maximum information gain.
Compute the optimal $y_L = -\frac{\sum_{n\in A_L} \widetilde{g}_n^\mu}{\sum_{n\in A_L} \widetilde{h}_n^\mu + \lambda}$ and $y_R = -\frac{\sum_{n\in A_R} \widetilde{g}_n^\mu}{\sum_{n\in A_R} \widetilde{h}_n^\mu + \lambda}$.
Inform districts in I_k of the optimal results.
Repeat 1-13, but switch to the σ' components instead.

Framework 2 is accordingly modified, albeit slightly, into Framework S2, to better adapt to the AdaBoost algorithm.

We then perform a case study on the prediction performance of the AdaBoost-based multi-task learning framework. Three case studies are prepared for comparison: (a) the proposed framework with simultaneous withdrawal, (b) the proposed framework with dynamic withdrawal, and (c) single-district training without multi-task learning. The NLL loss statistics are presented in Table S1.

One can reach the following conclusions from the table:

1) The NLL losses of the multi-task learning framework, regardless of the withdrawal mechanism adopted, outperform those of the single-district models where multi-task learning is not involved. Therefore, **the multi-task learning framework can indeed be generalized to other boosting-based algorithms, and the advantage in prediction accuracy still holds.**
2) Between the two withdrawal mechanisms, the simultaneous withdrawal mechanism is still better in terms of the mean and maximum NLL loss across districts (in fact, it is better in 8 out of 9 test districts). How-

Framework S2: Collaborative Multi-Task Learning Based on AdaBoost

Notations:

I_k: the set of remaining districts in epochs k;
$S_k^{\mu}, S_k^{\sigma'}$: the decision stump constructed in epoch k on μ and σ';
$G_{(k)}$: the global model updated after epoch k;
$L_{(k)}^{G}(\cdot)$: the loss of the global model on the dataset $(\cdot)$;
$D_i, \cup_{i\in I} D_i$: the load dataset for District i and for all districts;
G, H_i: the global model and the local model for District i;
F_i^{μ}, F_i^{σ}: the final model on μ and σ.

In each epoch *k*:

(Stage 1):
Districts in I_k execute **Framework S1** to compute $(S_k^{\mu}, S_k^{\sigma'})$.
Update global model $G = G_{(k)}$ by boosting the decision stumps.
Districts in I_k calculate $\mathcal{L}_{(k)}^{G}(D_i)$ (or $\mathcal{L}_{(k)}^{G}(\cup_{i\in I} D_i)$, depending on the withdrawal mechanism) to determine I_{k+1}.
(Stage 2):
Districts $\{i | i \notin I_k, i \in I_{k-1}\}$ perform local training until the sign of overfitting appears, to obtain the local model H_i.
Combine G and H_i via model addition, to get $(F_i^{\mu}, F_i^{\sigma})$.

Table SI: NLL Loss of the AdaBoost Algorithm with/without the Generalized Multi-Task Learning Framework.

Case	Mean NLL Across Districts	Maximum NLL Across Districts	NLL on District 11 (Unseen)
Simultaneous Withdrawal	-0.4021	-0.0513	-0.4600
Dynamic Withdrawal	-0.2717	0.0947	-0.5623
Without Multi-Tasking	0.0279	0.6127	-0.2825

ever, in terms of NLL loss on the new district, it is outperformed by the dynamic withdrawal mechanism. This is probably because the sample weights introduced in AdaBoost change the dynamics within the framework, which our qualitative analysis in the main text did not take into account.

B. Load Forecasting Curves of Other Districts

In Fig. 9 of the main paper, we demonstrate the prediction performances of the proposed framework by plotting the forecasting curves of District 1, using both the simultaneous and dynamic withdrawal mechanisms. To demonstrate the performance is consistent on all districts under study, in Fig. S1, we plot the load forecasting curves of 8 other districts. Note that the 10-th district, namely District 7, is not plotted due to the lack of a proper test set.

Regarding the plot itself, we wish to explain that in some districts, the final model might improve little from the global model, which leads to their plots overlapping with each other, in terms of either the lines (indicating the mean) or the shaded areas (indicating the standard deviation).

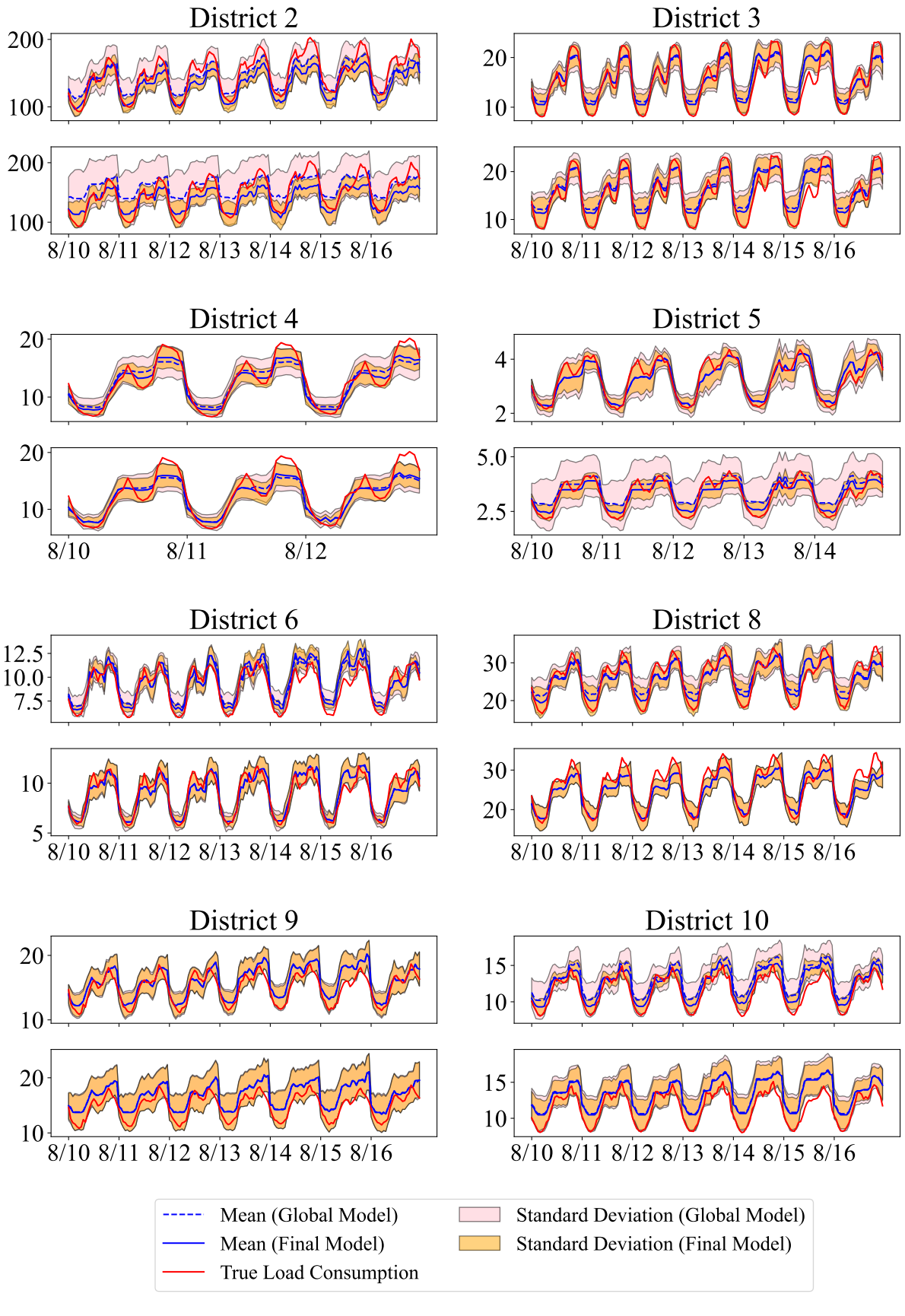

Figure S1: District load forecasting curves for 8 other districts in the training set. Horizontal axis: date. Vertical axis: load (MW). Each district contains two load forecasting subplots, generated by the simultaneous (up) and dynamic (down) withdrawal mechanisms, respectively.