Submitted to *the INFORMS Data Challenge Competition*

FOR INTERNAL USE ONLY (leave blank)
Submission Number:
Team Number:

Two-Stage Hurdle Models for Outage Forecasting

Shourya Bose, Tomas Kaljevic, and Yu Zhang*
Department of Electrical and Computer Engineering, University of California, Santa Cruz,
1156 High St., Santa Cruz, CA 95064, USA, zhangy@ucsc.edu

We present our solution to the INFORMS 2025 Data Challenge on forecasting hourly power outages across 83 Michigan counties for 24 and 48 hour horizons. Motivated by the zero inflation and burstiness of outage counts and the weak marginal correlations between individual weather covariates and outages, we develop a two-stage hurdle model. The first stage estimates the probability of a nonzero outage event, and the second stage predicts outage magnitude conditional on an event. Because weather covariates are not provided for the forecast window, We train a separate model TimesNet to forecast the weather features and construct a curated subset of those that are most informative for outage prediction. On the competition dataset, our approach outperforms SARIMAX, LSTM, and all-zeros prediction baseline on location-averaged RMSE, and attains a leaderboard RMSE of 189.

1. Introduction

The INFORMS 2025 Data Challenge tasks participants with forecasting county-level hourly outages across Michigan for 24 and 48 hour lookahead horizons. The dataset includes outage counts and multiple weather features over the three months preceding the forecast windows. Outage counts are both zero-inflated (the majority are zero) and bursty (nonzero events appear in short clusters). In addition, the Pearson correlation between any single weather feature and outages has small magnitude. These characteristics degrade the performance of conventional time-series models such as SARIMAX and LSTM.

Guided by findings in [Chapados, 2014], we adopt a two-stage hurdle design. The first stage predicts the probability that the outage is nonzero; the second stage predicts the outage magnitude conditional on a nonzero event. Because the competition does not provide weather covariates for the forecast

* Corresponding author.

windows, we train a dedicated model to forecast all weather features. On held-out data our model improves RMSE over LSTM, SARIMAX, and the baseline (all-zeros predictions), and achieves a leaderboard RMSE of 189.

2. Methodology

We next describe the training and inference pipeline. The code for our model is available at GitHub repository. Bold symbols (e.g., $\mathbf{a}$) denote vectors, and calligraphic symbols (e.g., $\mathcal{T}$) denote sets.

2.1. Competition Dataset

The dataset, derived features, and horizon lengths are summarized below.

- **Location and time indices:** Counties $i \in \{1, \dots, N\}$ with $N = 83$, and timesteps $t \in \mathcal{T}$ covering April 1 to June 30, 2023 (hourly resolution).
- **Outages and weather features:** Outage counts $o_{i,t} \geq 0$ and informative weather vectors $\mathbf{h}_{i,t} \in \mathbb{R}^{39}$ (a curated subset of 39 out of 109 total features, selected by variance and correlation [Bommert et al., 2022]; see Appendix A).
- **Horizon lengths:** Lookback window $L = 48$ and lookahead horizons $T \in \{24, 48\}$.

2.2. Dataset Splitting

We partition the timeline as $\mathcal{T} = \mathcal{T}_{\text{train}} \cup \mathcal{T}_{\text{val}} \cup \mathcal{T}_{\text{test}} \cup \mathcal{T}_{\text{hold}}$, where each split contains all locations $i \in \{1, \dots, N\}$. The sets $\mathcal{T}_{\text{val}}$, $\mathcal{T}_{\text{test}}$, and $\mathcal{T}_{\text{hold}}$ occupy the last days of the dataset, with the remainder in $\mathcal{T}_{\text{train}}$. Their sizes are

$$|\mathcal{T}_{\text{val}}| = L + mT \text{ for } m = 5, \quad |\mathcal{T}_{\text{test}}| = L + T, \quad |\mathcal{T}_{\text{hold}}| = \max\{48 - T, 0\},$$

where the holdout segment aligns the test windows for the 24-hour and 48-hour models. The competition forecast windows, denoted by $\mathcal{T}^{(24)}_{\text{forecast}}$ and $\mathcal{T}^{(48)}_{\text{forecast}}$, both begin immediately after $\mathcal{T}$.

2.3. Feature Engineering

For each location-timestep pair (i, t), we construct the following feature vector:

$$\mathbf{x}_{i,t} := \texttt{concat}\Big(i,\ o_{i,t-L:t-1},\ \kappa_i,\ r_{i,t-1},\ \mathbf{h}_{i,t-1},\ \Delta\mathbf{h}_{i,t-1},\ \boldsymbol{\tau}_t,\ o_{i,t-1} \times \mathbf{h}^{(\text{top})}_{i,t-1}\Big), \tag{1}$$

where

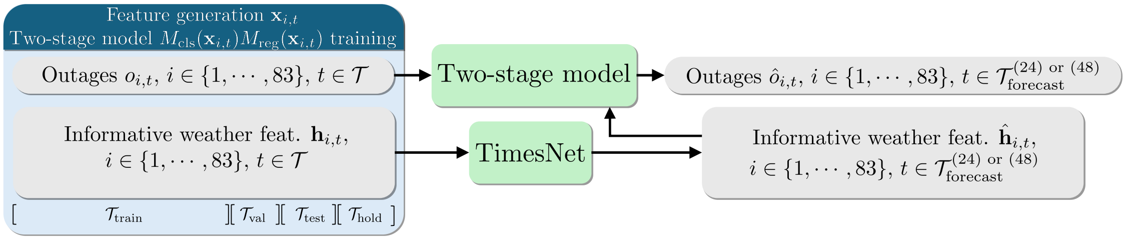

Figure 1 This figure presents the training and inference flow for the two-stage hurdle model.

- $\Delta \mathbf{h}_{i,t-1} := \mathbf{h}_{i,t-1} - \mathbf{h}_{i,t-2}$ is the first-order difference of the informative weather vector.
- $\mathbf{h}_{i,t-1}^{(\text{top})} \in \mathbb{R}^5$ are the top-five informative weather features with the highest training-set variance.
- $r_{i,t-1}$ is the run length since the last zero, defined recursively by

$$r_{i,t-1} = \begin{cases} 0, & \text{if } o_{i,t-1} > 0, \\ r_{i,t-2} + 1, & \text{if } o_{i,t-1} = 0. \end{cases}$$

- $\kappa_i := \frac{1}{|\mathcal{T}_{\text{train}}|} \sum_{t \in \mathcal{T}_{\text{train}}} o_{i,t}$ is the average outage level.
- $\boldsymbol{\tau}_t$ is the cyclical encoding of Hour of Day (`HoD`) and Day of Week (`DoW`), given as

$$\boldsymbol{\tau}_t := \left(\sin\left(\frac{2\pi \texttt{HoD}(t)}{24}\right), \cos\left(\frac{2\pi \texttt{HoD}(t)}{24}\right), \sin\left(\frac{2\pi \texttt{DoW}(t)}{24}\right), \cos\left(\frac{2\pi \texttt{DoW}(t)}{24}\right) \right).$$

Note that in our feature engineering, incorporating the first-order difference of the weather vector enables the model to capture temporal dynamics and short-term variability, leading to more accurate and robust forecasts. Additionally, cyclical encoding using sine and cosine functions preserves the natural periodicity of time features, ensuring that adjacent hours or days remain close in representation and thus promoting smoother model learning and continuity.

2.4. Two-Stage Hurdle Model

The first stage predicts the probability that the outage at (i,t) is nonzero given $\mathbf{x}_{i,t}$; the second stage forecasts the magnitude of $o_{i,t}$.

2.4.1. First-Stage Model We train a classifier M_{cls} using gradient-boosted decision trees GBDT (LightGBM [Ke et al., 2017]; Appendix B). The model produces raw probabilities

$$p_{i,t}^{\text{raw}} := \Pr\left[o_{i,t} > 0 \mid \mathbf{x}_{i,t}\right] = M_{\text{cls}}^{\text{raw}}(\mathbf{x}_{i,t}). \tag{2}$$

We fit $M_{\text{cls}}^{\text{raw}}$ on $\mathcal{T}_{\text{train}}$ with cross-entropy loss and then calibrate via isotonic regression on $\mathcal{T}_{\text{val}}$ (`IsotonicRegression` in Scikit-learn [Pedregosa et al., 2011]):

$$g^* := \arg\min_{g \text{ is monotone}} \sum_{i\, t \in \mathcal{T}_{\text{val}}} \left(z_{i,t} - g(p_{i,t}^{\text{raw}})\right)^2, \quad M_{\text{cls}}(\mathbf{x}_{i,t}) = p_{i,t}^{\text{final}} := g^*\left(p_{i,t}^{\text{raw}}\right) \in [0,1], \tag{3}$$

where $z_{i,t} \in \{0,1\}$ indicates whether $o_{i,t}$ is nonzero.

2.4.2. Second-Stage Model We train a regressor M_{reg} using the Tweedie objective in LightGBM (Appendix C). Training uses only samples with $o_{i,t} > 0$. The Tweedie family interpolates between Poisson ($\rho = 1$) and Gamma ($\rho = 2$). Training of M_{reg} is performed via maximum-likelihood estimation with $\rho = 1.55$, as determined from validation results. Inference uses the conditional expectation. The resulting *autoregressive one-step ahead forecasting model* is

$$\boxed{\hat{o}_{i,t} = \mathbb{E}\left[o_{i,t} \mid o_{i,t} > 0, \mathbf{x}_{i,t}\right] = M_{\text{cls}}(\mathbf{x}_{i,t}) \times M_{\text{reg}}(\mathbf{x}_{i,t}).} \tag{4}$$

We emphasize that a single two-stage model is trained jointly across all locations rather than fitting separate models per county.

2.4.3. Forecasting Informative Weather Features Because $\mathbf{h}_{i,t}$ is needed for $t \in \mathcal{T}_{\text{forecast}}^{(24)}$ and $t \in \mathcal{T}_{\text{forecast}}^{(48)}$, we introduce a weather forecaster $W : \mathbb{R}^{L^w \times 109} \to \mathbb{R}^{T^w \times 109}$ that predicts the next T^w steps of $\mathbf{h}_{i,t}$ given the previous L^w steps. Following our prior experience [Bose et al., 2024], we use TimesNet [Wu et al., 2023]. We train two models with $(L^w, T^w) = (120, 24)$ and $(120, 48)$, each shared across locations, and use them to populate $\mathbf{h}_{i,t}$ on $\mathcal{T}_{\text{forecast}}^{(24)}$ and $\mathcal{T}_{\text{forecast}}^{(48)}$.

2.5. Results

With $\mathbf{h}_{i,t}$ available for the forecast windows, we apply (4) to produce outage forecasts. We train separate $\left(M_{\text{cls}}, M_{\text{reg}}\right)$ pairs for $T \in \{24, 48\}$. The evaluation metric is location-averaged RMSE:

$$\texttt{RMSE} = \frac{1}{N} \sum_{i=1}^{N} \sqrt{\frac{1}{T} \sum_{t=1}^{T} \left(o_{i,t} - \hat{o}_{i,t}\right)^2}, \quad T \in \{24, 48\}. \tag{5}$$

For the 24 hour model, $\mathcal{T}_{\text{test}}$ spans 1:00 AM June 28 to 12:00 AM June 29, 2023; for the 48 hour model it spans 1:00 AM June 28 to 12:00 AM July 1, 2023. Table 1 reports losses on $\mathcal{T}_{\text{test}}$ against baselines.

Table 1 **This table reports Root Mean Square Error (RMSE) for each method. For each lookahead horizon T, the lowest RMSE is shown in bold and the second-lowest is underlined.**

Method	24h RMSE	48h RMSE
Baseline (all-zeros predictions)	43.61	70.49
SARIMAX $((p,d,q) = (2,0,2))$	37.87	<u>63.52</u>
LSTM (with sparsity clustering)	<u>33.84</u>	72.09
Two-stage hurdle (ours)	**19.86**	**47.88**

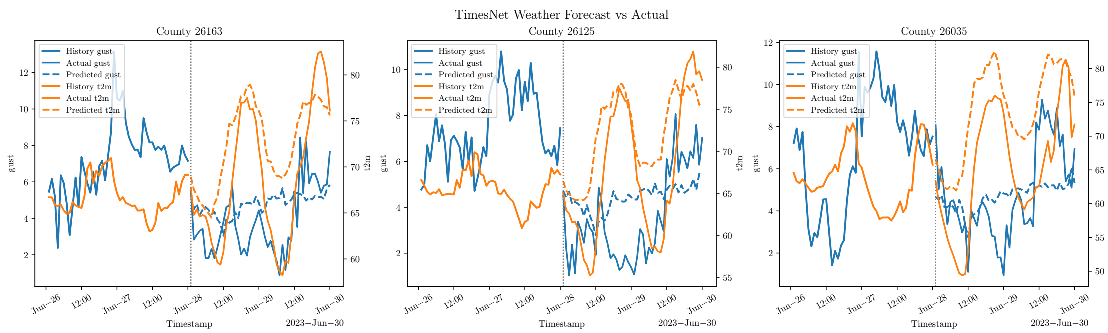

Figure 2 **This figure shows TimesNet forecasts for two weather features (`gust` and `t2m`) across three locations.**

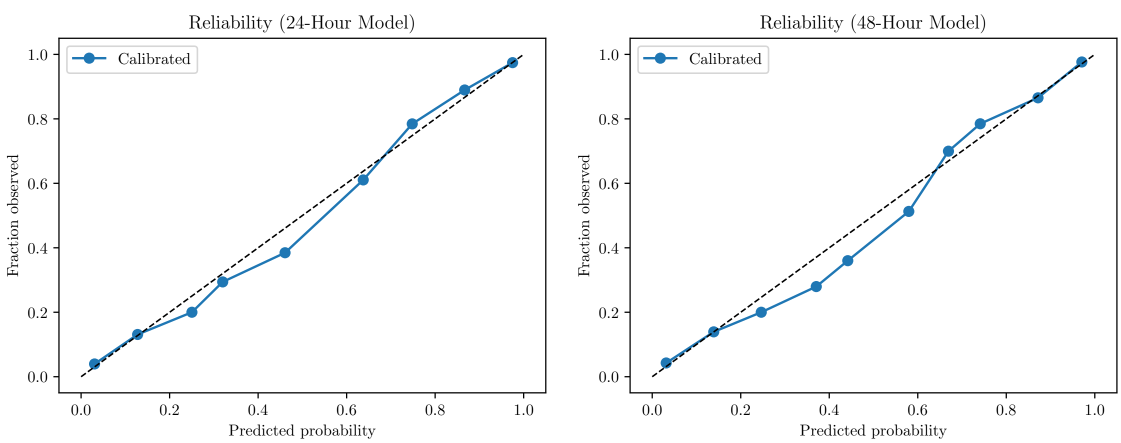

Figure 3 **This figure shows reliability curves illustrating the effect of isotonic regression for the 24-hour and 48-hour models.**

3. Conclusion

We proposed a two-stage hurdle framework for zero-inflated outage forecasting with informative weather covariates. The first stage estimates the probability of a nonzero event; the second stage predicts magnitude conditional on an event. We also forecast the required weather covariates over the

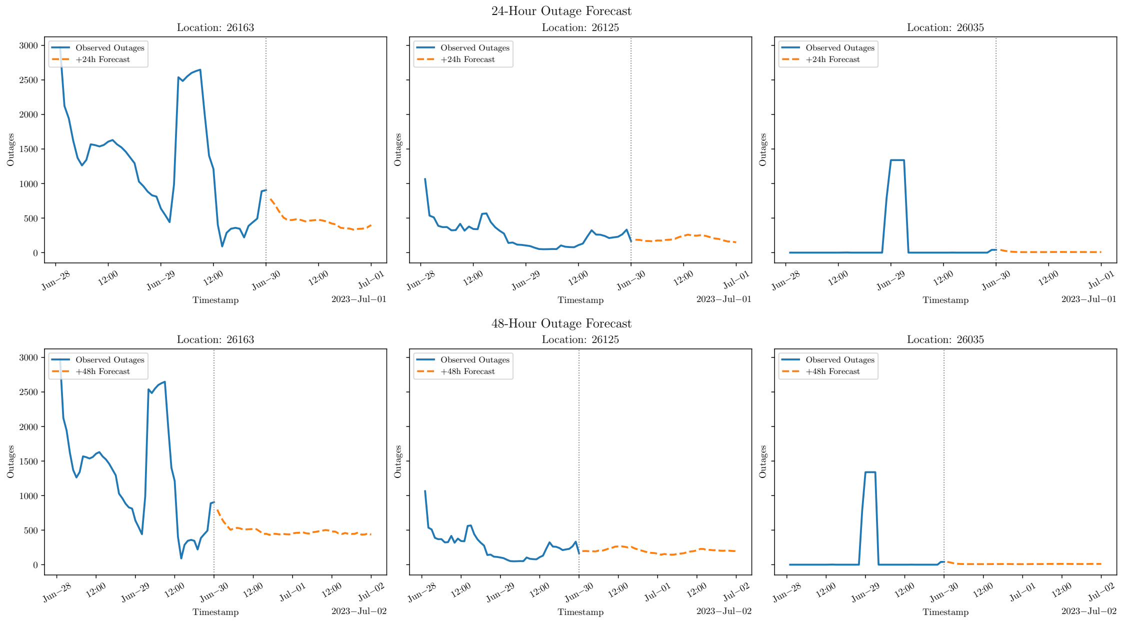

Figure 4 This figure presents the forecasted outages generated by the two-stage hurdle model for the 24 hour and 48 hour lookahead horizons at three representative locations.

competition windows using TimesNet. The combined approach improves RMSE over strong time-series baselines and achieves a leaderboard RMSE of 189 on the INFORMS 2025 Data Challenge.

4. Team Members

This work was conducted by Tomas Kaljevic, Shourya Bose, and Yu Zhang in the Department of Electrical and Computer Engineering at the University of California, Santa Cruz. Correspondence: `tkaljevi@ucsc.edu`, `shbose@ucsc.edu`, `zhangy@ucsc.edu`.

References

[Bommert et al., 2022] Bommert, A., Welchowski, T., Schmid, M., and Rahnenführer, J. (2022). Benchmark of filter methods for feature selection in high-dimensional gene expression survival data. *Brief Bioinform*, 23(1).

[Bose et al., 2024] Bose, S., Li, Y., Sant, A. V., Zhang, Y., and Kim, K. (2024). From RNNs to foundation models: An empirical study on commercial building energy consumption. In *NeurIPS Workshop on Time Series in the Age of Large Models*.

[Chapados, 2014] Chapados, N. (2014). Effective Bayesian modeling of groups of related count time series. In Xing, E. P. and Jebara, T., editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1395–1403, Bejing, China. PMLR.

[Ke et al., 2017] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Édouard Duchesnay (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830.

[Wu et al., 2023] Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., and Long, M. (2023). Timesnet: Temporal 2D-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*.

Appendix A: Informative Weather Features and Sparsity Clustering

The competition dataset contains 109 total weather features, many of which are constant or weakly informative. We curate 39 informative features: `wx_sde`, `wx_sdwe`, `wx_gust`, `wx_fricv`, `wx_lai`, `wx_hail_1`, `wx_hail`, `wx_refc`, `wx_max_10si`, `wx_unknown_3`, `wx_unknown`, `wx_cnwat`, `wx_gh_3`, `wx_ustm`, `wx_snowc`, `wx_refd_1`, `wx_veril`, `wx_tcc_1`, `wx_refd`, `wx_lcc`, `wx_gh_5`, `wx_tcc`, `wx_sdlwrf`, `wx_pcdb`, `wx_sbt114`, `wx_veg`, `wx_unknown_5`, `wx_sbt124`, `wx_mcc`, `wx_gh`, `wx_pwat`, `wx_vstm`, `wx_pres`, `wx_lsm`, `wx_hcc`, `wx_gh_4`, `wx_vis`, `wx_orog`, `wx_cape`. The choice is guided by correlation with outages (Figure 5) and variance (Figure 6).

Our initial approach clustered counties according to their outage sparsity, defined as the proportion of zero outages, and trained a separate LSTM model for each cluster. As shown in Figure 7, counties were grouped into three sparsity levels: low (<40%), medium (40–80%), and high (>80%). This clustering strategy yielded better performance than training a single LSTM across all locations. However, it still failed to fully capture the zero-inflated and bursty characteristics of the dataset.

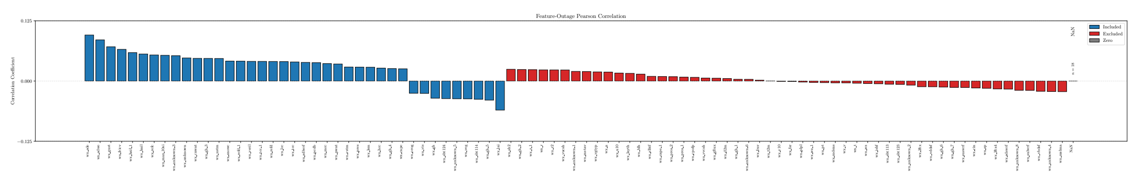

Figure 5 This figure reports correlations between weather features and outage counts (zoom for detail).

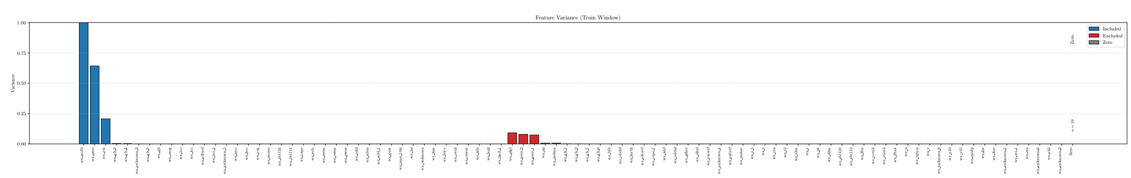

Figure 6 This figure reports normalized variances of weather features (zoom for detail).

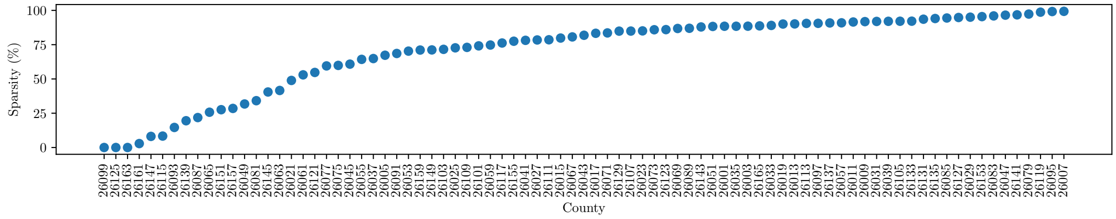

Figure 7 This figure illustrates the outage sparsity percentage across all counties in the dataset (used for LSTM model only).

Appendix B: GBDT Configuration for $M_{\mathbf{cls}}$

```python
import lightgbm as lgb
classifier_params = {
    "objective": "binary",
    "learning_rate": 0.005,
    "num_leaves": 1000,
    "max_depth": 12,
    "min_child_samples": 100,
    "min_sum_hessian_in_leaf": 5.0,
    "subsample": 0.85,
    "subsample_freq": 1,
    "colsample_bytree": 0.9,
    "feature_fraction_bynode": 0.8,
    "reg_alpha": 0.03,
    "reg_lambda": 0.03,
    "n_estimators": 10000,
    "random_state": master_seed,
    "scale_pos_weight": scale_pos_weight,
    "verbosity": -1,
}
if linear_leaves:
    classifier_params["linear_tree"] = True
classifier = lgb.LGBMClassifier(**classifier_params)
```

Appendix C: Tweedie GBDT Configuration for $M_{\mathbf{reg}}$

```python
import lightgbm as lgb
regressor_params = {
    "objective": "tweedie",
    "tweedie_variance_power": 1.55,
    "learning_rate": 0.005,
    "num_leaves": 1000,
    "max_depth": 12,
    "min_child_samples": 150,
    "min_sum_hessian_in_leaf": 10.0,
    "subsample": 0.85,
    "subsample_freq": 1,
    "colsample_bytree": 0.9,
    "feature_fraction_bynode": 0.8,
    "reg_alpha": 0.03,
    "reg_lambda": 0.03,
    "n_estimators": 12000,
    "random_state": master_seed,
    "verbosity": -1,
}
if linear_leaves:
    regressor_params["linear_tree"] = True
regressor = lgb.LGBMRegressor(**regressor_params)
```