BART Overheat Prediction

Mentor

Yu Shen: Data Scientist @ BART

Undergraduate Researchers

Alaina Kolling, Ayumu Justin Ueda, Luca Butera, Tiange Chen, Yuwei Chen

Abstract

This study improves **overheating prediction** in BART train control rooms by replacing a currently deployed model (**F1: 0.35–0.52**) with a **Gated Recurrent Unit (GRU)** neural network, achieving improved **F1 scores of 0.48–0.77** for air-conditioned stations. These results highlight the GRU's ability to effectively capture **complex temporal patterns**.

Introduction and Objective

Overheating in BART train control rooms poses a **significant risk** to system reliability and train operational efficiency, leading to potential service delays. BART operates **62 train control rooms** across its network, with **17** stations using **forced constant ventilation** and **45** relying on **air conditioning** for cooling purposes. Current monitoring systems at BART lack the predictive capability needed to identify potential overheating events in advance and implement preventative measures to avoid them. This study aims to develop neural network-based models to forecast temperatures and detect early warning signs of overheating.

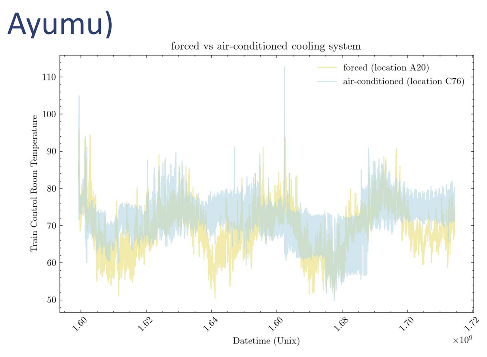
The **primary objectives a**re:

1. To develop two predictive models tailored to BART's cooling mechanisms:
   - One for stations using **forced constant ventilation** (forced)
   - One for stations relying on **air conditioning** (air-conditioned)
2. To accurately forecast if 2 hours ahead temperature is over **85**, **90** degree.
3. To enable **proactive cooling interventions** to minimize service disruptions

Dataset

We used **four years of time series data** on control room temperatures across 62 locations.

(Figure 1, Ayumu)



*Due to the difference in these two cooling systems' signals that can be seen in figure 1, we decided to build two classifiers for each cooling system.

*We completely removed 37 locations from air-conditioned group since those did not have any overheat points in the last four years.

We split this dataset into Dataset A (before 2024-04-30 23:52:01) and Dataset B (after the same time).

Validation results (**OOS**) shown in Table 1 and 2 are based on Dataset A, while test results (Test) come from Dataset B**,** which was never used for hyperparameter tuning. Thus, results on Dataset B reflect a "**real-world**" evaluation of the model's performance.

Regression vs Classification

Since overheated data points made up only 0.01–0.75% of all data, we initially used MSE loss, classifying points as 1 if the predicted temperature exceeded thresholds (85, 90) and 0 otherwise. However, **reducing MSE didn't consistently improve the F1-score**. Therefore, we switched to cross-entropy loss despite there being a limited number of overheated data points.

Our Approach

Models

The benchmark model currently deployed is a **Quasi-linear model** (Formula 1).

Formula 1:

$$\text{Output} = \sigma\left(T_{\text{current}} + \left(\frac{\sum_{i=1}^{8} \frac{T_i - T_{i-1}}{900}}{8}\right) \cdot 7200\right)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Where:

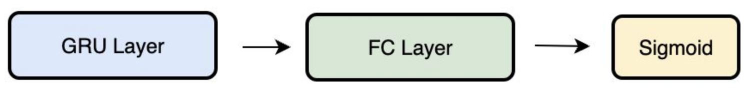
T_{current} : Current temperature,

$\frac{T_i - T_{i-1}}{900}$: First-order derivative of temperature (change over 900 seconds, or 15 minutes),

7200 : Scaling factor corresponding to 2 hours (in seconds).

Building on this benchmark model, we developed a neural network-based classifier with a **Gated Recurrent Unit (GRU)** as shown in Figure 2.

(Figure 2, Ayumu)



* As a side note, we opted not to use LSTM in this task, as our experiments showed that the GRU-based model outperformed the LSTM-based model (Luca).

Feature Engineering

The only feature engineering applied was the addition of two features, **velocity** and **acceleration** (Formula 2).

Formula 2. Velocity and acceleration calculations:

$$v = \frac{T_{\text{current}} - T_{\text{prev}}}{900}, \quad a = \frac{v_{\text{current}} - v_{\text{prev}}}{900}$$

Where:

T_{current} : Current temperature,

T_{prev} : Temperature 15 minutes earlier,

v_{current} : Current velocity,

v_{prev} : Previous velocity.

*Yuwei explored adding environmental data from the OpenMeteo API, but since it would require real-time API calls during deployment, potential network issues could disrupt consistent inference. Therefore, we decided not to use these features.

*Ayumu attempted to incorporate location data (longitude and latitude) using methods from Paper **[5]** and **[6]**, but these approaches were unsuccessful.

Preprocessing Methods

- Overheat data points were **augmented** by adding Gaussian noise to the duplicated overheat data, as shown in Formula 3 and proposed by Alaina, increasing their proportion to 4%.

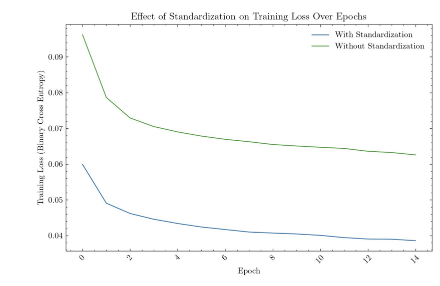
Formula 3. $X_{\text{duplicated}} + 0.01 \cdot \text{Noise}, \quad \text{where Noise} \sim \mathcal{N}(0, 1)$

- Ayumu applied **standardization** to the data, leading to faster training convergence, as illustrated in Figure 3.

*No data cleaning was performed because no significant difference in the model's performance was observed.

*Yuwei tried other augmentation methods like Synthetic Minority Oversampling Technique (SMOTE), but we did not see significant differences in results. **[4]**

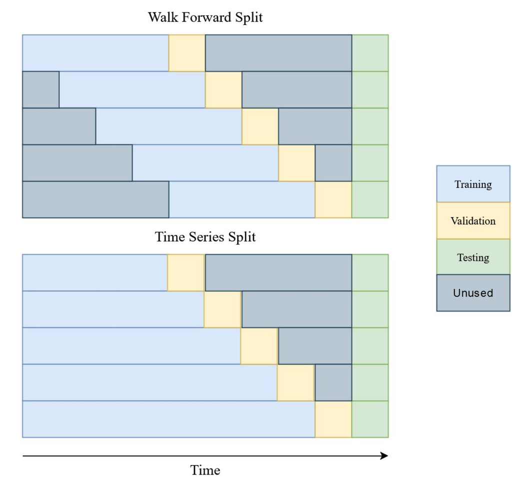
(Figure 3, Luca)



Validation Methods

Traditionally, either **Time Series Split** or **Walk Forward Split** (illustrated in Figure 4), as suggested by Luca, is used to cross-validate time series models.

(Figure 4, Yuwei)



However, we encountered two challenges:

- GRU generally require a large amount of data, but both methods reduce the size of the training dataset.
- We could not ensure the inclusion of all instances of overheat across the subsets of validation, as each fold exhibited variability in the data.

Given the constraints, an **out-of-sample (OOS)** validation was the only viable option. Data was divided into **train** (before 2023-01-01), **validation** (2023-01-01 to 2023-09-01), and **test** (after 2023-09-01) based on arbitrary date ranges.

*Ayumu found a paper **[1]** in which the author proposed that, in the case of a purely autoregressive model, standard K-fold Cross-Validation (CV) can be applied, provided the models being evaluated have uncorrelated errors. However, this approach yielded inconsistent results in our case.

Results

Table 1: Model Performance with 85 degree threshold

Model	OOS (%)	Test (%)	Cooling System
Quasi-linear model	86.0	**92.0**	forced
GRU	**92.0**	90.0	forced
Quasi-linear model	52.0	51.0	air-conditioned
GRU	**77.0**	**66.0**	air-conditioned

Table 2: Model Performance with 90 degree threshold

Model	OOS (%)	Test (%)	Cooling System
Quasi-linear model	78.0	82.0	forced
GRU	**88.0**	**84.0**	forced
Quasi-linear model	35.0	44.0	air-conditioned
GRU	**48.0**	**54.0**	air-conditioned

Explanation: - **OOS**: Out-of-Sample validation (holdout validation) - **Test**: Performance on the unseen dataset. - **Metrics**: F1-Score

GRU Model Parameters: input size = 3, hidden size = 16, GRU layer number = 2, window size = 16, batch size = 64, epoch = 30, optimizer = AdamW, learning rate = 1×10^{-3}.

Conclusion

1. For the 85°F threshold, GRU and Quasi-linear models performed similarly due to this data's periodic cycle and many 85°F points.
2. However, for the 90°F threshold and all thresholds in air-conditioned systems, our GRU model consistently outperformed.

This result suggests that our GRU model becomes increasingly effective as the threshold rises and/or when the cooling system is air-conditioned.

Discussion

Hyperparameter Tuning

Due to computational and time constraints, as well as the research-oriented nature of this project, **we did not prioritize hyperparameter optimization**.

Data Augmentation

Although Yuwei explored data augmentation methods **[4]**, further investigation is needed, particularly since the ultimate goal is to predict a threshold of 95. The amount of data available for the **95-degree threshold is extremely limited**, necessitating more robust augmentation techniques.

Model Architecture

Our mentor, Yu, discovered a paper presenting an MLP and Transformer-based model **[2]**. Yuwei later implemented the model, while Tiange and Ayumu experimented with Temporal Convolutional Networks and 1D convolutions as alternatives to GRU layers. However, none of these approaches were thoroughly explored, leaving their full potential untapped.

References

[1] Bergmeir, C., Hyndman, R. J., & Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis, 120*, 70–83. https://doi.org/10.1016/j.csda.2017.11.003

[2] Chen, S.-A., Li, C.-L., Yoder, N., Arik, S. O., & Pfister, T. (2023). *TSMixer: An all-MLP architecture for time series forecasting*. arXiv. https://arxiv.org/abs/2303.06053

[3] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). *Empirical evaluation of gated recurrent neural networks on sequence modeling*. arXiv. https://arxiv.org/abs/1412.3555

[4] Ilbert, R., Zhu, X., Wang, Y., & Brown, T. (2024). Data augmentation for multivariate time series classification: An experimental study. *2024 IEEE 40th International Conference on Data Engineering Workshops (ICDEW)*, 128–139. https://doi.org/10.1109/icdew61823.2024.00023

[5] Li, D., Chen, Y., Li, Z., Zhang, X., & Zhang, J. (2021). Integrating static and time-series data in deep recurrent models for oncology early warning systems. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 913–936. https://doi.org/10.1145/3459637.3482441

[6] Liu, Q., You, Q., Min, Y., Li, L., & Zhang, X. (2022). Machine learning crop yield models based on meteorological features and comparison with a process-based model. *Artificial Intelligence for the Earth Systems, 1*(4). https://doi.org/10.1175/aies-d-22-0002.1

Contributions

This poster represents a collaborative effort:

- Abstract, Introduction and Objective, and References were written by Alaina.
- Dataset, Regression vs. Classification, Results, Conclusion, Discussion (Yuwei contributed), and Our Approach were written by Ayumu.

The experiments presented in the Results section were conducted by both Alaina and Ayumu. Prior to identifying the optimal architecture, Luca and Yuwei contributed significantly to the modeling efforts. Additional contributions are credited throughout the text, with specific ideas attributed to individuals, such as "(name)" to recognize their input.

Berkeley
UNIVERSITY OF CALIFORNIA