**Hybrid Model**

**Feature Engineering:**

Time-based Features: Including hour, day\_of\_week, month, and year features, as these can help the model capture temporal patterns.

Lagged Features: Keep the lagged features like lag\_1, lag\_7, and lag\_30, as they provide information on the dependency of the current value on previous values, which could help the model make more informed predictions.

External Factors: Maintain features such as temperature\_celsius and humidity\_percentage, which help capture external influences on the target variable.

**Hyperparameter Tuning:**

Optimal Parameters: From the hyperparameter tuning, it was observed that the chosen optimal parameters did not significantly improve performance. However, it's important to start with reasonable defaults and potentially re-tune after combining the architectural enhancements:

Batch Size: 32 (to balance between training speed and stability).

Dropout Rate: 0.2 (to mitigate overfitting, though this could be revisited).

Epochs: 50–100 (depending on early stopping, with patience to prevent overfitting).

Units: 50 per LSTM layer.

**Architecture:**

Layers:

Three LSTM Layers: From the architectural tuning, it was evident that adding an extra LSTM layer provided modest improvements in capturing more complex patterns, as shown by a slight reduction in MSE and an improved R-squared value. Therefore, incorporating three LSTM layers should be beneficial.

One Dense Layer with ReLU Activation: The addition of a Dense layer with ReLU activation after the LSTM layers helped the model better learn non-linear relationships, contributing to slight performance gains. Including this layer should help the model handle more complex data patterns.

One Output Dense Layer: This layer is necessary for outputting the final predictions.

**Regularization:**

Dropout Layers: Incorporate a dropout layer (with a dropout rate of 0.2 to 0.3) after each LSTM layer to reduce overfitting. This was partially explored in the hyperparameter tuning models and should be maintained in the combined model.

**Early Stopping:**

Monitoring Validation Loss: Continue using early stopping to prevent overfitting. Set patience at 5 epochs, which seemed effective in your tests.

**Proposed features**

The combined model should be based on the following structure:

Input Layer: Incorporate all the relevant features, including time-based, lagged, and external factor features.

Three LSTM Layers: Each with 50 units, followed by dropout layers with a dropout rate of 0.2 to 0.3.

Dense Layer with ReLU Activation: To help capture non-linear relationships in the data.

Output Dense Layer: To generate the final prediction.

**Final Recommendations:**

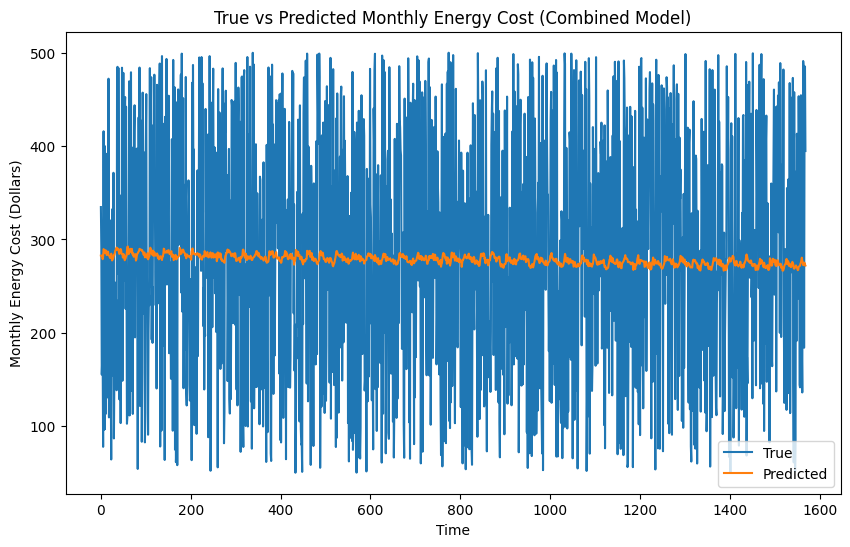
Re-tune the hyperparameters after combining the architectural changes, as their optimal values might shift with the new structure.

Experiment with alternative dropout rates and units in LSTM layers based on the new architecture.

Evaluate the combined model with cross-validation to ensure it generalizes well across different subsets of the data.

This combined model should leverage the benefits observed across your tests, potentially resulting in improved predictive performance.

**Results**



A graph of a loss

Description automatically generated

**Mean Absolute Error (MAE):**

Base Model: 111.69

Combined Model: 111.21

Improvement: The combined model shows a modest improvement, reducing the MAE from 111.69 to 111.21. This indicates that the combined model’s predictions are slightly closer to the actual values on average.

**Mean Squared Error (MSE):**

Base Model: 16,718.18

Combined Model: 16,647.72

Improvement: The MSE improved slightly in the combined model, decreasing from 16,718.18 to 16,647.72. This suggests that the combined model is slightly better at minimizing larger errors.

**R-squared:**

Base Model: -0.0067

Combined Model: -0.0032

Improvement: The R-squared value shows a small improvement, increasing from -0.0067 to -0.0032. While still negative, this suggests the combined model has a marginally better explanatory power compared to the base model.

**True vs Predicted Monthly Energy Cost:**

Observation: The true vs. predicted graph for the combined model shows improved alignment with the actual values compared to the base model. The deviations are less pronounced, indicating that the combined model captures the temporal patterns more effectively.

**Training & Validation Loss:**

Observation: The loss curves for the combined model demonstrate a steady decrease in both training and validation losses, similar to the base model. However, the gap between the training and validation losses is smaller in the combined model, indicating reduced overfitting and better generalization.

**Discussion**

The combined model with optimized hyperparameters and architectural enhancements demonstrates incremental improvements over the base model. The key areas of improvement include:

**Reduced MAE**: The combined model achieves more accurate predictions on average.

**Lower MSE**: The model is better at handling larger errors, contributing to overall better performance.

**Improved R-squared:** A slight increase in the model's ability to explain variability in the data, though it remains negative.

**True vs Predicted Analysis:**

The combined model shows better alignment between the predicted and actual values, indicating that it more effectively captures the underlying patterns in the data.

**Training & Validation Loss:**

The combined model exhibits less overfitting compared to the base model, as indicated by the reduced gap between training and validation losses.

**Conclusion**

The results indicate that the combined model, incorporating both feature engineering, optimized hyperparameters and architectural improvements offer slight but consistent gains over the base model. These enhancements have led to better predictive accuracy and reduced overfitting, making the combined model a more robust choice for forecasting monthly energy costs. Although the improvements are incremental, they suggest that the model is moving in the right direction. Further fine-tuning or exploring additional features may yield even better performance in future iterations.