Indian Institute of Technology, Ropar

Project: - Load Forecasting using Lag-Llama

Students	Entry Numbers
G. Satya Prakash	2021EEB1172
N. Praneeth	2021EEB1189
Y. Pratheek	2021EEB1224
J. Uday Singh	2021EEB1179
R. Jayaprakash	2021EEB1202

1. Introduction:

In the domain of energy systems, load forecasting serves as a cornerstone for effective grid management and resource planning, encompassing both short-term and long-term horizons. Our study focuses on the application of the Lag-LLAMA model to the task of load forecasting, spanning various time scales. Leveraging its adaptive model aggregation and machine learning capabilities, Lag-LLAMA offers a robust framework for capturing the complex dynamics of electricity demand, ranging from immediate fluctuations to longer-term trends.

As we delve into the intricacies of load forecasting with the Lag-LLAMA model, our objective is to provide comprehensive insights into its performance across different forecasting horizons. From short-term predictions to long-term projections, we aim to assess the model's accuracy, reliability, and scalability. Through meticulous data preprocessing, model development, and evaluation, we endeavor to unlock the full potential of the Lag-LLAMA model in addressing the diverse challenges of load forecasting in modern energy systems. By shedding light on its capabilities and limitations, we strive to contribute to the advancement of load forecasting methodologies and facilitate informed decision-making in the management of power grids.

1.1. Background:

Electricity demand forecasting plays a crucial role in the efficient operation and planning of power systems. Short-term load forecasting, which involves predicting electricity consumption over a horizon ranging from a few hours to several days, is particularly important for optimizing generation, transmission, and distribution resources in real-time.

1.2. Problem Statement:

The primary objective of this study is to develop and evaluate load forecasting models based solely on historical load data without the incorporation of additional variables.

2. Literature Survey:

- Machine Learning Based Short-Term Load Forecasting for Smart Meter Energy Consumption Data in London Households

  https://ieeexplore.ieee.org/document/9501104

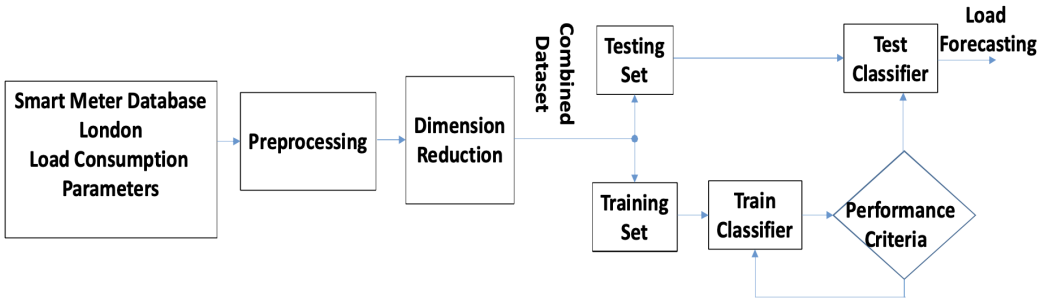
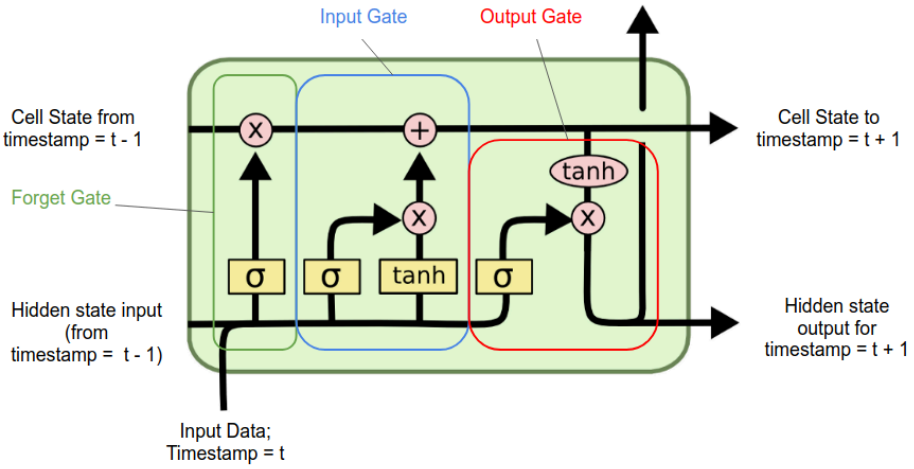


Fig.1. The proposed system block diagram.

- Short-term Load Forecasting on Smart Meter via Deep Learning

  https://ieeexplore.ieee.org/abstract/document/9000185



- Probabilistic Individual Short-Term Load Forecasting Using Conditional Variational Autoencoder

  https://ieeexplore.ieee.org/abstract/document/10252364

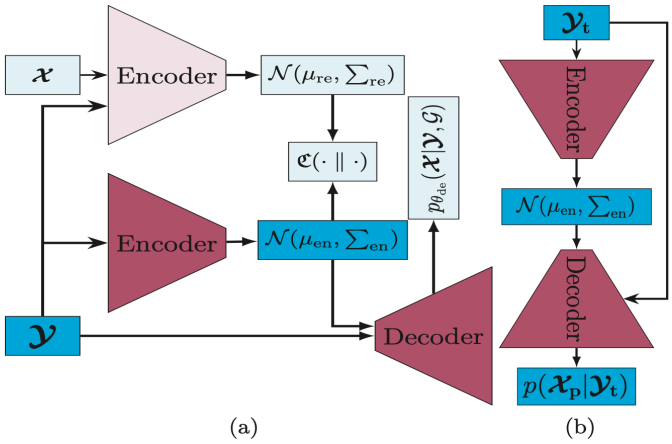


Fig. 1. The architecture of the proposed CVAE: (a) during training; (b) during test.

3. Methodology:

3.1. Model Selection:

In our study, we have opted to utilize foundational models such as Lag-LLAMA for short-term load forecasting. Foundation models are machine learning models trained on extensive and diverse datasets, enabling their application across a wide range of use cases. These models have revolutionized artificial intelligence, serving as the backbone for various generative AI applications, including ChatGPT.

Lag-LLAMA stands out as the first open-source foundation model specifically designed for time series forecasting tasks. Built upon extensive research and experimentation, Lag-LLAMA offers a versatile and scalable framework for predicting time-dependent patterns in data. Its architecture and training methodology are detailed in the research paper available at https://arxiv.org/abs/2310.08278.

By leveraging Lag-LLAMA, we aim to harness the power of foundational models to achieve accurate and reliable short-term load forecasting results. Its adaptability and performance make it a compelling choice for addressing the forecasting challenges in our study.

3.2. Description of Lag-Llama Model:

The "Lag-Llama" model is a sophisticated approach to time series forecasting, using a transformer-based architecture optimized for univariate probabilistic forecasting. Here's a detailed look at the mathematical and structural elements of the model:

Mathematics of Lag-Llama:

- The foundation of Lag-Llama is probabilistic forecasting where future values of a time series are predicted by estimating parameters of a probability distribution based on past values and covariates.
- It leverages the chain rule of probability to construct the likelihood of future values given historical data, modeled as:

$$p_\phi(x^i_{C+1:C+P} \mid x^i_{1:C}, \mathbf{c}^i_{1:C+P}; \theta) = \prod_{t=C+1}^{C+P} p_\phi(x^i_t \mid x^i_{1:t-1}, \mathbf{c}^i_{1:t-1}; \theta).$$

Here,

$x_{i,1:C}$	:lagged values used as inputs
$c_{i,1:C+P}$	:covariates across the forecast period
P	:no. of future values
ϕ	:the parameters of a parametric distribution.
C	:sub-sample fixed context window size
θ	:neural network parameters

Architecture:

Lag Features and Tokenization:

- The model processes input data by forming "tokens" from lagged observations. Lags are chosen based on the time granularity relevant to the forecasting (e.g., daily, weekly).
- Each token combines lagged values with additional date-time features like day of the week or hour, which help the model understand temporal patterns.

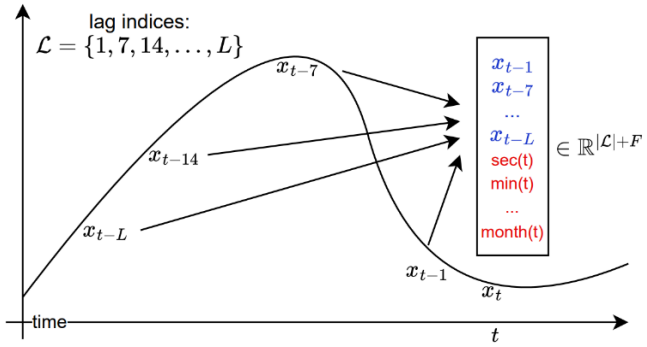


Figure 1: *For a time series, we depict the tokenization at the timestep t of the value xt which contains lag features constructed using an example set of lag indices L, where each value in the vector is from the past of xt (in blue), and F possible temporal covariates (date-time features) constructed from timestamp t (red).*

Decoder-only Transformer:

- Lag-Llama employs a decoder-only transformer, which uses positional encodings to maintain the temporal context of inputs. The architecture is designed to process sequences of lagged tokens and predict future values.

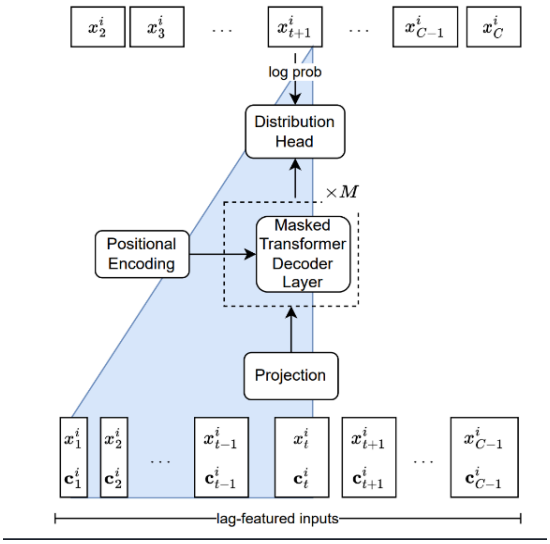


Figure2:*The* ***Lag-Llama architecture****. Lag-Llama learns to output a distribution over the values of the next time step based on lagged input features. The input to the model is the token of a univariate time series i at a given timestep, x^i_t, constructed as described. Here, we use c^i_t to refer to all additional covariates used along with the value at a timestep t, which include the |L| lags, F date-time features, and summary statistics. The inputs are projected through M masked decoder layers. The features are then passed through the distribution head and trained to predict the parameters of the forecast distribution of the next timestep.*

Training and Evaluation:

- The model is pretrained on a diverse dataset comprising various time series domains, enabling it to learn generalized features applicable across different datasets.
- It employs a distribution head at the output, which is trained to predict the parameters of a chosen probability distribution (e.g., Gaussian, Student's t-distribution) for the forecast.

Normalization and Scaling:

- Implements robust standardization by adjusting for the median and scaling according to the interquartile range, ensuring the model's robustness to outliers in the data.

4. Implementation of Model for short term load forecasting:

4.1. Data Collection:

4.1.1. Data Sources:

The data used in this study were sourced from the Smart Meter Data for Mathura and Bareilly, provided by the Council on Energy, Environment and Water (CEEW) and available on Kaggle [1]. The dataset contains smart meter readings collected from residential and commercial consumers in the cities of Mathura and Bareilly, covering the year 2019.

4.1.2. Description of the Dataset:

Data Format: The dataset is provided in CSV (Comma-Separated Values) format, which is commonly used for storing tabular data.
Variables: Each row in the dataset represents a single meter reading and contains various variables, including:

- Timestamp: Date and time of the meter reading
- Meter ID: Unique identifier for each smart meter
- Consumption (kWh): Electricity consumption recorded by the smart meter

Coverage: The dataset covers a wide range of smart meters installed in residential and commercial properties across Mathura and Bareilly, providing a representative sample of electricity consumption patterns in these cities.

4.2. Data Preprocessing:

- *Data Aggregation:*
  Individual meter readings were aggregated to consolidate load demand within specific geographical areas, providing a comprehensive overview of electricity consumption patterns.
- *Temporal Resolution Adjustment:*
  The dataset was transformed into data with a ***9-minute granularity.***
  This adjustment facilitated the exploration of forecasting models operating at different temporal resolutions, providing flexibility in analysis and model selection.
- *Conversion to Power Values:*
  Energy readings were converted into power values to reflect the instantaneous electricity demand accurately.
- *Quality Control Measures:*
  Stringent quality control procedures were implemented to detect and rectify anomalies such as missing values, outliers, or inconsistencies.

4.3. Model Evaluation and Optimization:

We embarked on the implementation phase by first subjecting the Lag-LLAMA model to **zero-shot testing**, a crucial step aimed at evaluating its initial performance without any parameter adjustments or optimization. For this assessment, we made predictions for various time horizons, including 2, 10, and 40 points into the future. This diversified approach enabled us to gauge the model's predictive capabilities across different forecast horizons, providing valuable insights into its inherent strengths and weaknesses.

Following the zero-shot testing phase, we proceeded to **fine-tune** the Lag-LLAMA model to enhance its predictive accuracy and refine its performance for short-term load forecasting. Fine-tuning involved a meticulous process of parameter optimization and configuration adjustments, guided by the observations and insights gleaned from the initial zero-shot testing results. Leveraging advanced optimization techniques, we systematically adjusted model parameters to optimize its performance across the selected forecast horizons.By iteratively fine-tuning the Lag-LLAMA model based on the observed results.

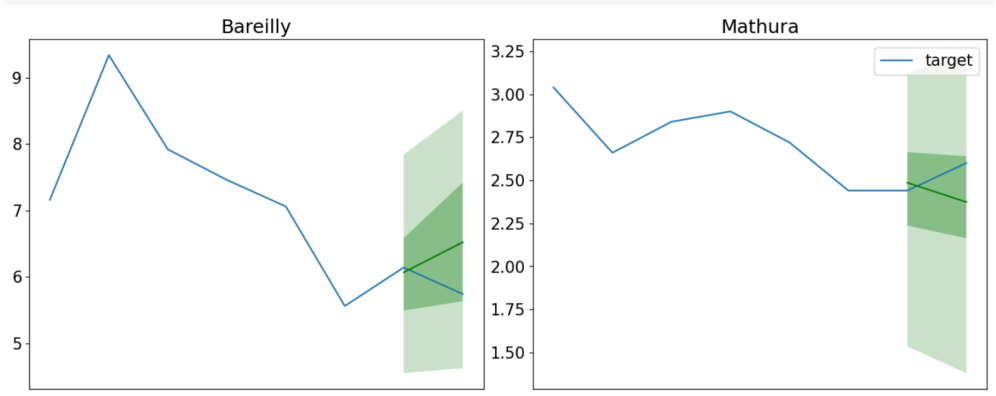
Overall, the combination of zero-shot testing and fine-tuning served as a comprehensive strategy to assess and optimize the Lag-LLAMA model for the task at hand. Through meticulous experimentation and iterative refinement, we aimed to leverage the full capabilities of the model and achieve accurate and reliable short-term load forecasts.

4.4. Results:

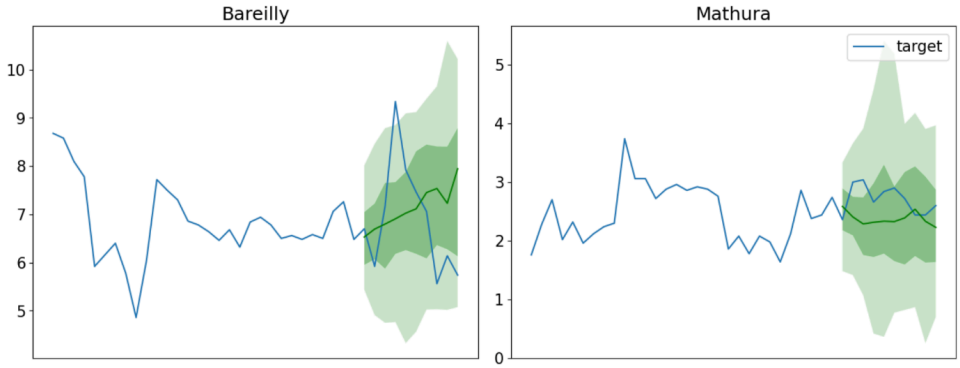
Here the interval between the consecutive points is 9 minutes.

Zero-shot:

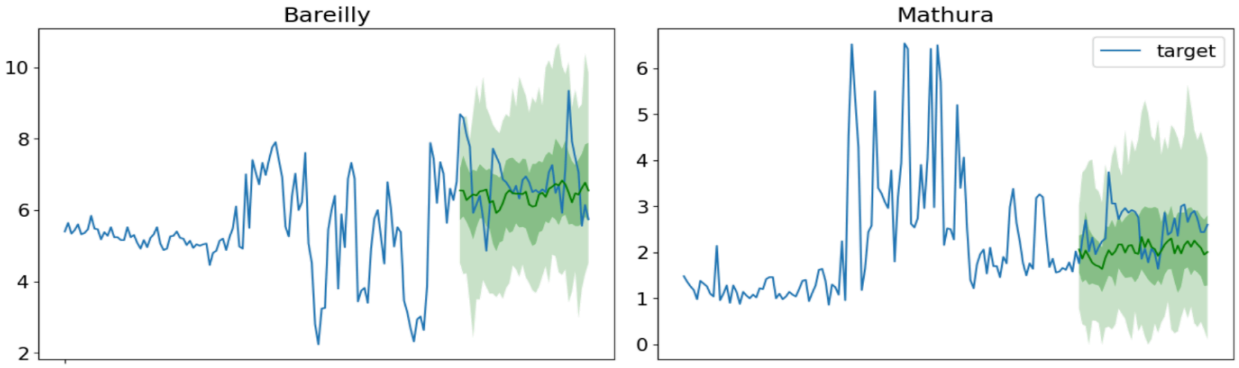
2 point prediction:



10 point prediction:

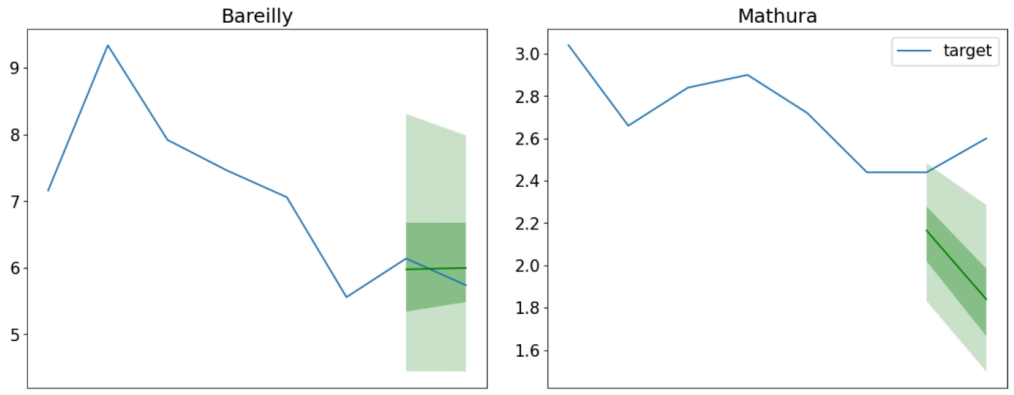


40 point prediction:

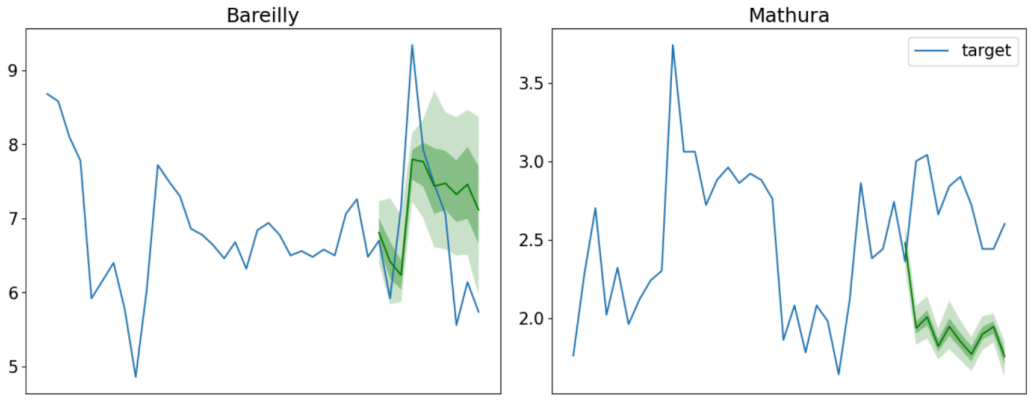


Fine tuning:

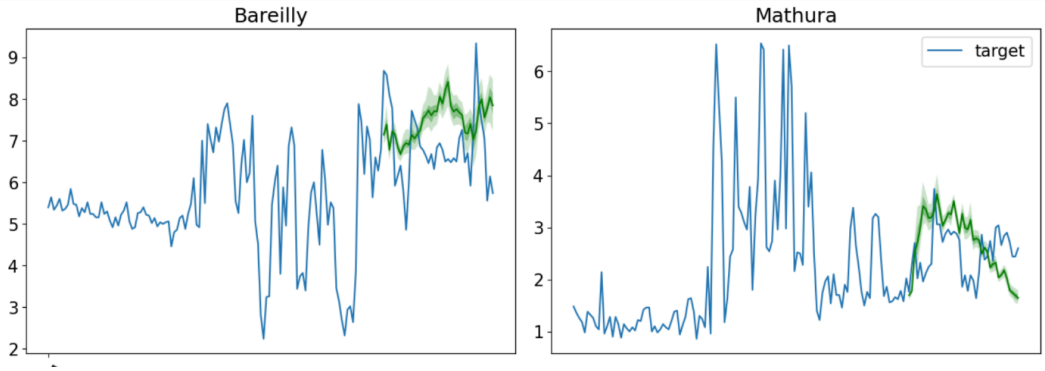
2 point prediction:



10 point prediction:



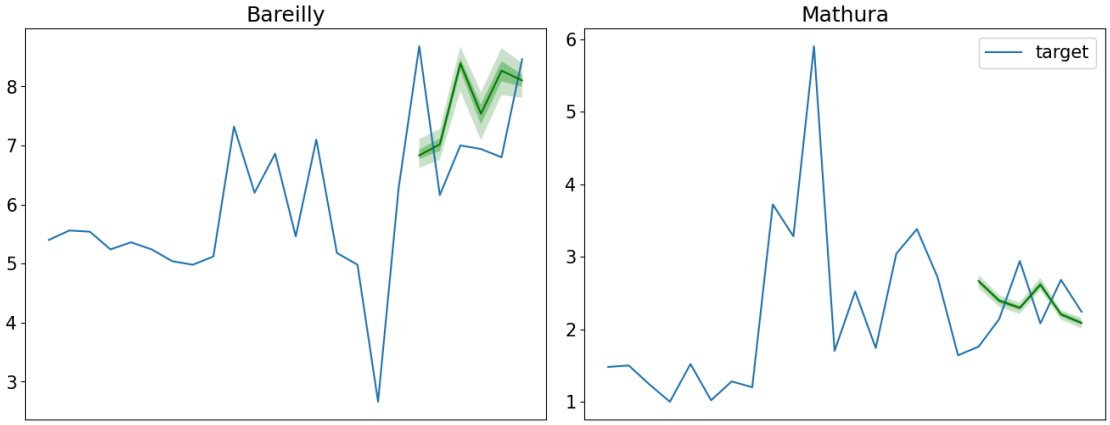
40 point prediction:



Prediction length(9min)	Finetuneing	MSE	CRPS
x2	Zero-shot	0.1707	0.0605
x10	Zero-shot	0.8879	0.1175
x40	Zero-shot	0.6831	0.1162
x2	Fine-tuned	0.1868	0.0825
x10	Fine-tuned	0.8969	0.1522
x40	Fine-tuned	0.9866	0.1655

- We can observe that, even after fine tuning, comparatively the error did not get any better.
- We can see that the model performs well for the short-term forecasting (2 point prediction) and the error increases as we try to predict more points because there can be spikes in the data if we consider more points.
- In order to make the model perform better for long-term forecasting(40 point prediction), we can feed the Hourly data to the model so that the sudden spikes are reduced in the data.

Six point prediction for hourly data:(Fine tuned)



MSE: 0.8643344839413961
CRPS: 0.15354048379619636

Here, we can see that the error gets reduced for the same time frame prediction when compared to the 40 point fine tuned prediction.

So we can say that, When the interval of the data and the prediction length is comparable to the input data time framework, we get better results.

5. Limitations and Future Work:

We've noted that fine-tuning the models leads to a decline in model performance and reduces the variance of the predicted load distribution. Therefore, future efforts could focus on exploring improved fine-tuning methods, possibly enhancing the loss function to boost prediction accuracy.

6. Conclusion:

In the course of this project, our utilization of the time-series foundational model Lag-LLAMA effectively enabled load prediction with a probabilistic approach. Despite our efforts to fine-tune the model, its performance did not meet our expectations. This outcome underscores the importance of ongoing refinement in our methodologies, suggesting the need for further exploration and enhancement to achieve desired levels of predictive accuracy.