Comprehensive Evaluation of LLMs for Time Series Forecasting

August 14, 2024

Problem Statement

Consider a multivariate time series dataset $\mathbf{X} \in \mathbb{R}^{T \times F}$, where T represents the number of time steps and F denotes the number of features. The primary objective is to develop a predictive model capable of forecasting a future sequence $\hat{\mathbf{X}}_{t+1:t+H} \in \mathbb{R}^{H \times F}$, where H is the prediction horizon, based on a given sequence of past observations $\mathbf{X}_{t-L+1:t} \in \mathbb{R}^{L \times F}$, where L denotes the input sequence length.

Formally, the task can be defined as finding a function $f : \mathbb{R}^{L \times F} \rightarrow \mathbb{R}^{H \times F}$ that minimizes the discrepancy between the predicted sequence $\hat{\mathbf{X}}_{t+1:t+H}$ and the actual future sequence $\mathbf{X}_{t+1:t+H}$. This problem is challenging due to the inherent complexity and high dimensionality of the time series data, as well as the potential presence of non-linear temporal dependencies and varying patterns across different time steps and features.

To address this problem, we employed a variety of deep learning models, each with distinct architectures and mechanisms for capturing temporal dependencies in the data:

- **DLINEAR:** The DLINEAR (Direct Linear) model is a fundamental approach to time series forecasting that simplifies the task by focusing exclusively on linear dependencies within the data. By assuming that the future values of the time series can be linearly extrapolated from past observations, DLINEAR effectively models the temporal relationships in the data without introducing the complexity of non-linear interactions. This model is particularly useful for scenarios where the underlying data exhibits strong linear trends or where computational simplicity is paramount. Despite its simplicity, DLINEAR can provide robust baseline performance for multivariate time series forecasting tasks.

- **FEDFORMER:** The FEDFORMER (Frequency Enhanced Decomposed Transformer) model is a sophisticated transformer-based architecture specifically designed for time series forecasting. It enhances the traditional transformer model by incorporating frequency domain analysis, which allows the model to efficiently capture both short-term and long-term dependencies in the data. FEDFORMER achieves this by decomposing the time series into components that are processed separately, enabling the model to focus on different frequency bands. This decomposition not only improves forecasting accuracy but also enhances the model's ability to generalize across different types of time series data. The efficient attention mechanisms employed by FEDFORMER make it well-suited for handling long sequences, a common challenge in time series forecasting.

- **INFORMER:** The INFORMER model is a variant of the transformer architecture that has been optimized for long-term time series forecasting tasks. Traditional transformers struggle with efficiency and scalability when applied to long sequences due to their quadratic complexity in attention mechanisms. INFORMER addresses these challenges by introducing a ProbSparse self-attention mechanism, which selectively focuses on the most informative parts of the input sequence, significantly reducing the computational burden. Additionally, INFORMER incorporates a distillation operation that compresses the input sequence, further enhancing efficiency. This model is particularly effective in scenarios where the forecasting horizon is extensive, and the computational resources are limited.

- **PATCHTST:** The PATCHTST (Patch Time Series Transformer) model innovatively applies patching techniques, commonly used in computer vision, to time series data. By dividing the input sequence into smaller, overlapping patches, PATCHTST allows the model to capture local temporal patterns more effectively. This approach mitigates the problem of long-range dependencies by enabling the model to process smaller segments of data sequentially, thereby reducing memory constraints and speeding up inference. The patches are then processed using transformer layers that aggregate the information to produce accurate forecasts. PATCHTST is particularly beneficial in applications where the time series data exhibits complex, localized patterns that are difficult to capture with traditional models.

- **TTM:** The Tensor-Train Model (TTM) leverages advanced tensor decomposition techniques to model multi-way temporal dependencies in time series data. By representing the multivariate time series as a high-dimensional tensor, TTM decomposes this tensor into a set of lower-dimensional components, capturing the interactions between different time steps and features more efficiently. This decomposition reduces the complexity of the model while preserving its ability to learn intricate patterns in the data. TTM is especially powerful in scenarios where the time series data has a high dimensionality, as it can effectively manage the curse of dimensionality and produce accurate forecasts with fewer parameters than traditional models.

Optimization Objective

The performance of these models is evaluated using a set of standard loss functions, which quantify the accuracy of the forecasts:

- **Mean Absolute Error (MAE):**

$$\text{MAE} = \frac{1}{H}\sum_{h=1}^{H}\left|\mathbf{X}_{t+h} - \hat{\mathbf{X}}_{t+h}\right|$$

  The MAE represents the average absolute difference between the actual and predicted values.

- **Mean Squared Error (MSE):**

$$\text{MSE} = \frac{1}{H}\sum_{h=1}^{H}\left(\mathbf{X}_{t+h} - \hat{\mathbf{X}}_{t+h}\right)^2$$

  The MSE penalizes larger errors more significantly, making it sensitive to outliers.

- **Root Mean Squared Error (RMSE):**

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{H}\sum_{h=1}^{H}\left(\mathbf{X}_{t+h} - \hat{\mathbf{X}}_{t+h}\right)^2}$$

  The RMSE provides an interpretable error metric in the same units as the data.

- **Mean Absolute Percentage Error (MAPE):**

$$\text{MAPE} = \frac{100\%}{H}\sum_{h=1}^{H}\left|\frac{\mathbf{X}_{t+h} - \hat{\mathbf{X}}_{t+h}}{\mathbf{X}_{t+h}}\right|$$

  The MAPE measures the average percentage error, giving insight into the relative prediction accuracy.

- **Mean Squared Percentage Error (MSPE):**

$$\text{MSPE} = \frac{100\%}{H}\sum_{h=1}^{H}\left(\frac{\mathbf{X}_{t+h} - \hat{\mathbf{X}}_{t+h}}{\mathbf{X}_{t+h}}\right)^2$$

  The MSPE, similar to the MAPE, focuses on the percentage error but with a quadratic penalty for larger deviations.

The ultimate goal is to identify the model and hyperparameters that yield the best performance across these metrics, thereby providing an accurate and reliable forecasting solution for the multivariate time series data.

Model	MAE	MSE	RMSE	MAPE	MSPE
DLINEAR	**0.312**	**0.216**	**0.466**	**4.7495**	**2275**
FEDFORMER	4.474	7.439	8.8	21.33	5424912
INFORMER	1.254	2.4338	1.56	3.1481	36090
PATCHTST	0.402	0.336	0.58	1.701	13098.73
TTM	0.63	0.54	0.7348	N/A	N/A

Table 1: Performance metrics of various models on the custom dataset.

Input and Output Sequences

- **Input Sequence:** The length of the input sequence is defined by the hyperparameter L, where $\mathbf{X}_{t-L+1:t} \in \mathbb{R}^{L\times F}$ represents the past L time steps used for prediction. - **Output Sequence:** The goal is to predict the next H time steps, where $\hat{\mathbf{X}}_{t+1:t+H} \in \mathbb{R}^{H\times F}$ is the predicted sequence.

Experimental Settings

In this study, we evaluated the performance of several deep learning models on a custom multivariate time series dataset. The input to each model consisted of a sequence of time series data, where the sequence length was defined by the `seq_len` hyperparameter, set to 386 time steps. The models were tasked with predicting the target feature over a future sequence length defined by the `pred_len` hyperparameter, which was set to 96-time steps. The target feature in this dataset was identified as `System`, and the frequency of the data was hourly, denoted by `freq = 'h'`.

The experiments were conducted using a single GPU (A100). The models were trained for 30 epochs with a batch size of 32. The initial learning rate was set to 0.05 and adjusted according to the `type1` strategy during training. Early stopping was implemented with a patience of 3 epochs to prevent overfitting. After training, each model was tested on the test set. The GPU memory was cleared after testing to ensure efficient use of resources.

The models evaluated in this study included DLinear, FEDFormer, Informer, PatchTST, and TTM. Each model was configured with specific hyperparameters. For DLinear, a label length of 48 (`label_len = 96`), and it was trained to predict a sequence of 96 time steps (`pred_len = 96`). Similarly, FEDFormer, Informer, PatchTST, and TTM were configured with the same input, label, and prediction lengths, with their respective checkpoints saved in their designated directories.

Results and Discussion

Table 1 presents the performance metrics of several deep learning models on the custom multivariate time series dataset. The models evaluated include DLinear, FEDFormer, Informer, PatchTST, and TTM. The performance of these models was measured using five key metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and Mean Squared Percentage Error (MSPE). These metrics provide a comprehensive understanding of the models' accuracy and error characteristics.

The DLinear model demonstrated strong performance across all metrics, with a particularly low MAE of 0.312 and an MSE of 0.216. The corresponding RMSE of 0.466 indicates that the model's predictions are quite close to the actual values on average. The model also achieved a MAPE of 4.7495, showing a relatively low average percentage error, and an MSPE of 2275, reflecting moderate sensitivity to larger errors.

In contrast, the FEDFormer model exhibited significantly higher errors, with an MAE of 4.474 and an MSE of 7.439, which led to a high RMSE of 8.8. The MAPE and MSPE for FEDFormer were also elevated, at 21.33 and 5424912, respectively. These results suggest that while FEDFormer may capture certain patterns in the data, it struggles with overall accuracy and is particularly sensitive to large deviations from the true values.

The Informer model showed a balanced performance, with an MAE of 1.254 and an MSE of 2.4338. Its RMSE of 1.56 and MAPE of 3.1481 indicate that the model maintains reasonable accuracy, although it is less precise compared to DLinear. The MSPE of 36090 suggests that Informer handles larger errors better than FEDFormer, but still with some difficulty.

PatchTST performed well, achieving an MAE of 0.402 and an MSE of 0.336. With an RMSE of 0.58 and a low MAPE of 1.701, PatchTST demonstrated its ability to accurately forecast the target values with minimal errors. Its MSPE of 13098.73 further indicates that the model is robust against larger deviations, outperforming both FEDFormer and Informer in this regard.

Lastly, the TTM model was evaluated, yielding an MAE of 0.63 and an MSE of 0.54. The RMSE of 0.7348 suggests moderate accuracy, though not as strong as DLinear or PatchTST. Due to the nature of the data and model, MAPE and MSPE were not applicable (N/A) for this model, highlighting some limitations in its application or the evaluation methodology used.