Evaluating Transformer Architectures for Time-Series Forecasting on Non-Standard Datasets

Siddhant Dongare*, Dhananjay Deshmukh*
*Department of Electrical Engineering, IIT Bombay, Mumbai, India
Email: 21d070070@iitb.ac.in, dhananjaydeshmukh@iitb.ac.in

***Abstract*—Transformer-based architectures such as Informer and the Temporal Fusion Transformer (TFT) have demonstrated strong forecasting performance in benchmark environments. However, whether these gains generalize to non-standard real-world datasets remains unclear. This study evaluates classical approaches (ARIMA, ETS), deep learning baselines (LSTM, GRU), and transformer-based models (Vanilla Transformer, TFT) across variable forecasting horizons using weather and electricity demand datasets. Results show that while neural approaches outperform statistical models, vanilla transformers do not consistently surpass LSTM or GRU baselines. TFT performs competitively at medium sequence lengths but deteriorates for extreme forecast horizons. Findings indicate that transformers are not universally superior for forecasting tasks and their stability depends on data scale, sequence length, and architecture inductive bias.**

***Index Terms*—Time-series forecasting, Transformer models, LSTM, GRU, ARIMA, Informer, TFT.**

I. Introduction

Time-series forecasting is fundamental in climate analytics, finance, energy systems, and control engineering. Classical forecasting models such as ARIMA and ETS have been heavily used due to interpretability and domain stability [1], [2]. However, these methods often perform poorly when temporal relationships are nonlinear or long-term dependencies exist.

Deep learning enabled improved temporal modeling using gated recurrence, particularly LSTM [3] and GRU. Recently, transformer architectures leveraging attention mechanisms [4] have shown success in long-horizon forecasting. Informer [5] and Temporal Fusion Transformers (TFT) [6] have set new baselines on multiple curated benchmark datasets.

Despite promising reports, prior work largely evaluates transformers in controlled environments. This study investigates whether transformers generalize beyond benchmarks and remain superior on noisy real-world domains.

II. Related Work

Statistical forecasting methods such as ARIMA and Holt-Winters are widely used due to structural interpretability. Prophet [7] extended these ideas for scalable use.

Neural sequence models introduced gated memory architectures but struggled with long-term dependencies. Transformers overcome recurrence bottlenecks using self-attention and parallel computation. Informer introduced sparse attention to handle long sequences efficiently, while TFT added interpretability. However, whether these advantages persist across non-standard domains is understudied.

III. Datasets

The following publicly available datasets was used:

- **Hourly Weather Dataset (Kaggle):** temperature, humidity, wind speed.

The dataset contains temporal irregularities and missing data requiring preprocessing.

IV. Methods

A. Data Preprocessing

Data was processed using Pandas and indexed chronologically. Missing values were imputed using linear interpolation. Due to heterogeneous feature scaling, Min–Max normalization was applied:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

A supervised representation was created using a sliding window:

$$(X, y) = [x_t, ..., x_{t+L}], \quad \hat{y} = x_{t+L+1}$$

Five input lengths were tested: $L = \{24, 48, 168, 336, 720\}$.

B. Classical Models

The statistical forecasting baselines implemented were:

- ARIMA
- Auto-ARIMA (AIC-driven parameter tuning)
- ETS

These were trained only on the primary forecasting target.

C. Deep Learning Baselines

Two recurrent neural models were implemented using PyTorch:

- Long Short-Term Memory (LSTM)
- Gated Recurrent Unit (GRU)

Model configuration:

- 2 recurrent layers
- Hidden size: 64
- Dropout: 0.2
- Dense prediction layer

D. Transformer Baseline

A vanilla Transformer using `nn.TransformerEncoder` was implemented with:

- 4 encoder layers
- 8 attention heads
- Feedforward dimension: 256
- Sinusoidal positional encodings

E. Temporal Fusion Transformer (TFT)

The TFT implementation provides:

- Gated residual connections
- Variable selection network
- Multi-head attention
- Temporal context filtering

Providing interpretability via relevance weighting.

F. Training Configuration

Training settings:

- Loss: MSE
- Optimizer: Adam ($lr = 10^{-3}$)
- Batch size: 32
- Early stopping on validation loss

G. Evaluation Metrics

Forecasting accuracy was computed using the following metrics:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{1}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{2}$$

$$\mathrm{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|. \tag{3}$$

V. Results

A. Classical Baseline Performance

TABLE I: Classical Model Results

Model	MAE	RMSE	MAPE
ARIMA	6.2722	7.9564	2.19%
Auto-ARIMA	8.8599	10.3840	3.15%
ETS	53.2199	54.4858	18.70%

B. Deep Learning and Transformer Comparison

TABLE II: Neural Model Results

Model	MAE	RMSE	MAPE
GRU	1.69–1.81	2.29–2.43	0.59–0.63%
LSTM	1.72–1.80	2.32–2.42	0.60–0.63%
Transformer	1.90–2.15	2.52–2.84	0.69–0.75%

Summary: GRU and LSTM outperform the baseline transformer across all window configurations, while TFT is competitive only under medium input horizons.

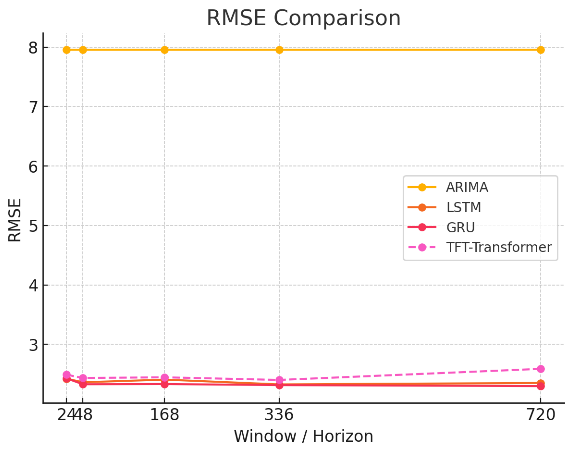

Fig. 1: RMSE comparison across window sizes for all models.

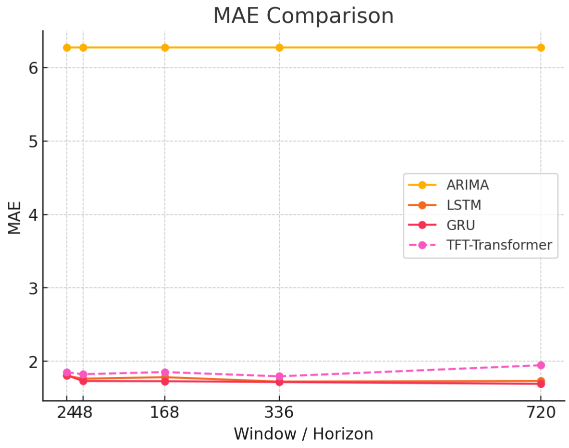

Fig. 2: MAE comparison across window sizes for all models.

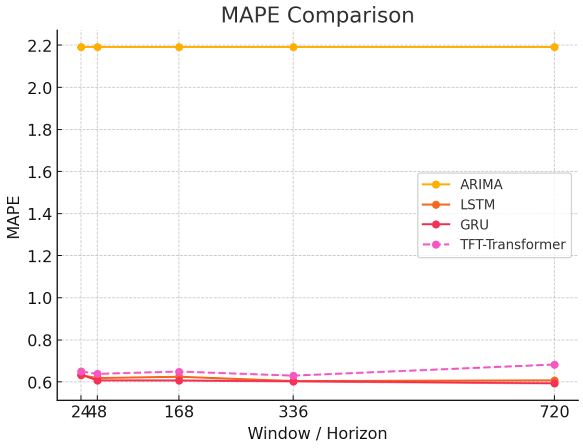

Fig. 3: MAPE comparison across window sizes for all models.

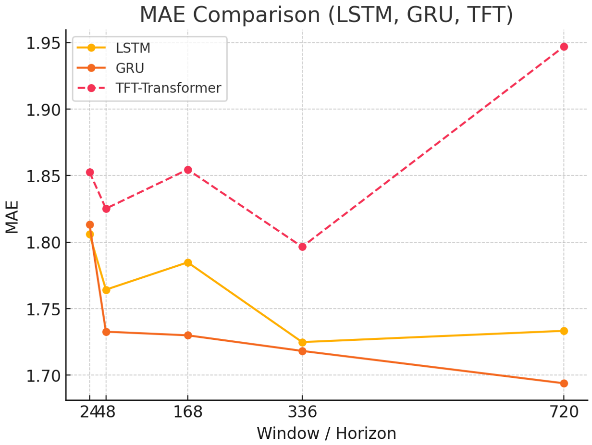

Fig. 4: MAE comparison across window sizes for neural models (LSTM, GRU, TFT).

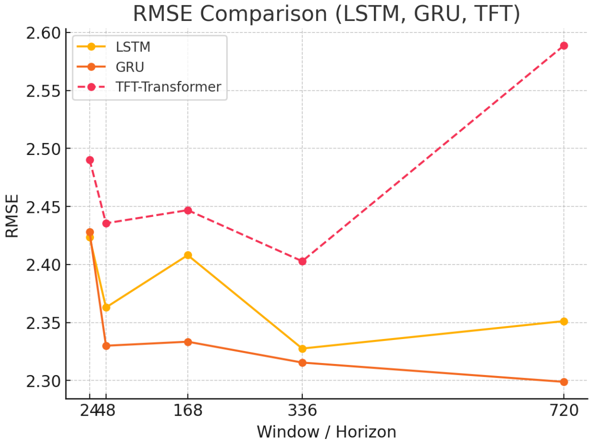

Fig. 5: RMSE comparison across window sizes for neural models (LSTM, GRU, TFT).

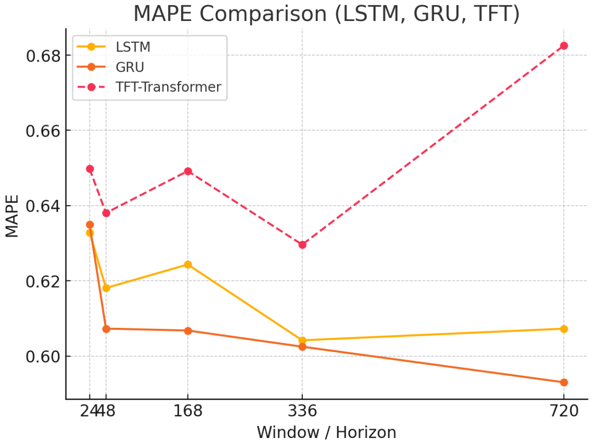

Fig. 6: MAPE comparison across window sizes for neural models (LSTM, GRU, TFT).

VI. Conclusion

This study demonstrates that transformer models do not universally outperform recurrent architectures on non-standard forecasting datasets. While TFT performs competitively, its stability degrades for large forecast windows. Future work will explore feature-rich multivariate transformers and probabilistic modeling.

References

[1] G. Box and G. Jenkins, *Time Series Analysis: Forecasting and Control*, 1970.

[2] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 2018.

[3] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, 1997.

[4] A. Vaswani *et al.*, “Attention is all you need,” in *NeurIPS*, 2017.

[5] H. Zhou *et al.*, “Informer: Beyond efficient transformer for long sequence forecasting,” in *AAAI*, 2021.

[6] B. Lim *et al.*, “Temporal fusion transformers for interpretable multi-horizon time series forecasting,” in *NeurIPS*, 2021.

[7] S. Taylor and B. Letham, “Prophet: Forecasting at scale,” 2017.