Advancing Temporal Forecasting: A Comparative Analysis of Conventional Paradigms and Deep Learning Architectures on Publicly Accessible Datasets

DATS 6501

Liang Gao

George Washington University

10 December 2024

Content

1 Overview

2 Methodology

3 Experiment results

4 Conclusion

5 Limitations

6 Future Work

List of Content

1 Overview
2 Methodology
3 Experiment results
4 Conclusion
5 Limitations
6 Future Work

Overview

- Time-series analysis is a critical area of research with applications spanning diverse fields such as finance, social science, and climate science.
- Numerous models have been developed to address specific problems in these domains.
- A noticeable lack of benchmarking studies comparing these models' performances on standardized datasets.
- The primary objective of this study is to evaluate and compare the performance of:
  - Classical models: AR, MA, ARMA, ARIMA
  - Modern techniques: LSTM, Bi-LSTM, Seq2Seq
  - State-of-the-Arts: Transformers

List of Content

1 Overview
2 Methodology
3 Experiment results
4 Conclusion
5 Limitations
6 Future Work

Data

- **Weather Station Beutenberg Dataset** Mnassri (2020): Contains meteorological measurements, such as temperature, humidity, and wind speed, recorded every 10 minutes. **temperature in Celsius**.
- **Power Consumption of Tetouan City** Salam and El Hibaoui (2018): Comprises the energy consumption data of Tetouan City, recorded at 10-minute intervals. **power consumption in Zone 1**.
- **Air Pollution Forecasting Dataset** Roy (2020): Hourly measurements of air pollutants along with meteorological features. **pollution called PM2.5 concentration**.

Models: Classical models (Box et al. (2015))

- **AR**: Forecast using a linear combination of past values of the variable. AR(n_a):

$$y(t) + a_1 y(t-1) + a_2 y(t-2) + \cdots + a_{n_a} y(t-n_a) = \epsilon(t)$$

- **MA**: Forecast using past forecast errors. MA(n_b:

$$y(t) = \epsilon(t) + b_1 \epsilon(t-1) + b_2 \epsilon(t-2) + \cdots + b_{n_b} \epsilon(t-n_b)$$

- **ARMA**: Combination of AR and MA models. ARMA(n_a, n_b)

$$y(t)+a_1 y(t-1)+a_2 y(t-2)+\cdots+a_{n_a} y(t-n_a) = \epsilon(t)+b_1 \epsilon(t-1)+b_2 \epsilon(t-2)+\cdots+b_{n_b} \epsilon(t-n_b)$$

- **ARIMA**: A generalization of the ARMA model with differencing. ARIMA(n_a, d, n_b):

$$\left(1 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a}\right)\left(1 - q^{-1}\right)^d y(t) = \left(1 + b_1 q^{-1} + \cdots + b_{n_b} q^{-n_b}\right) \epsilon(t)$$

Models: Classical models

Notation:

- $y(t)$ is the value of the time series at time t,
- $a_1, a_2, \ldots, a_{n_a}$ are the coefficients of the AR model,
- n_a is the order of the autoregressive model,
- $b_1, b_2, \ldots, b_{n_b}$ are the coefficients of the MA model,
- n_b is the order of the moving average model,
- $\epsilon(t)$ is a white noise normally distributed (WN $\sim (0, \sigma_\epsilon^2)$),
- d is the number of non-seasonal order differencing.

Models: Modern techniques

- **LSTM (Hochreiter and Schmidhuber 1997)**: Long Short-Term Memory: capture long-term dependencies in sequential data while mitigating the vanishing gradient problem
- **Bi-LSTM (Graves and Schmidhuber 2005(**: Bidirectional LSTM extends the LSTM by processing data bidirectionally, utilizing forward and backward LSTM to capture past and future context.
- **Seq2Seq (Sutskever, Vinyals, and Le 2014)**: Transform input sequences into output sequences of varying length.

State-of-the-art: Transformers

- The Transformers model, introduced by Vaswani et al. 2017, has significantly advanced deep learning with its attention mechanism and parallelized architecture. Unlike traditional recurrent models that process sequences sequentially, Transformers handle entire sequences simultaneously, making them more efficient at capturing long-term dependencies.

Analysis

- Time series domain knowledge:
  - Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF).
  - Generalized Partial Autocorrelation (GPAC).
- Optuna(Akiba et al. 2019): an open-source hyperparameter optimization framework.

List of Content

1 Overview
2 Methodology
3 Experiment results
4 Conclusion
5 Limitations
6 Future Work

Experiment setting

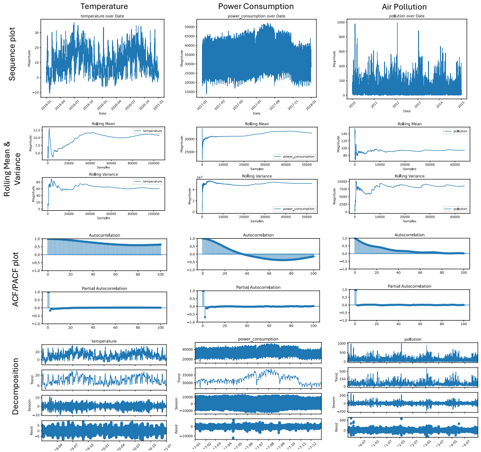
- Optuna: max n_a: 20, max n_b:10, trails: 30
- Hyperparameters for LSTM, Bi-LSTM, Seq2Seq, and Transformers:

Hyperparameter	**Value**
Sequence length	6
Batch Size	128
Optimizer	Adam
Training Epochs	100
Learning Rate	0.001

- Data was normalized to ensure fair comparison across datasets before calculating evaluation metrics using min-max scaling:

$$\mathsf{X}_{\text{normalized}} = \frac{\mathsf{X} - \mathsf{X}_{\min}}{\mathsf{X}_{\max} - \mathsf{X}_{\min}}. \tag{1}$$

Exploratory Data Analysis (EDA)



Exploratory Data Analysis (EDA)

- Strength of Trend and Seasonality for Each Dataset:

	Strength of Trend (%)	**Strength of Seasonality (%)**
Weather	94.33	74.79
Air Pollution	86.01	40.67
Power Consumption	92.41	98.64

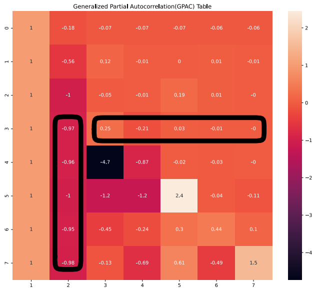
Domain knowledge vs. Optuna

- For AR and MA models, we use ACF and PACF plots to find the order
- For ARMA and ARIMA, we use GPAC.

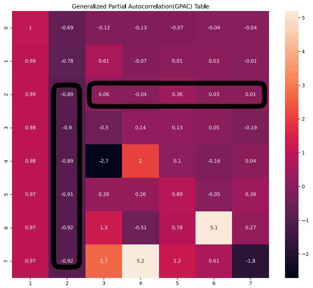
Domain knowledge vs. Optuna

- GPAC for three stationary raw datasets to find potential order for ARMA process:

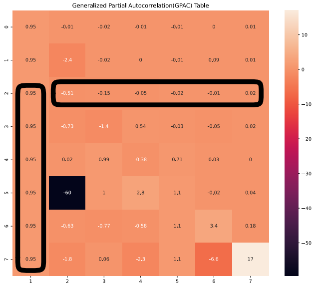
Temperature



Power Consumption



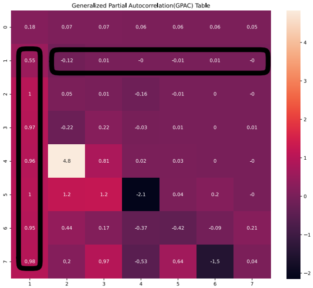
Air Pollution



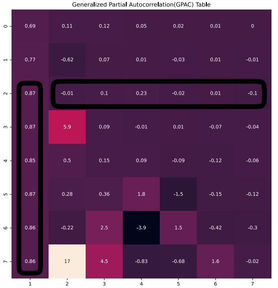
Domain knowledge vs. Optuna

- GPAC for three non-seasonal first-order differencing data to find potential order for ARMA process.

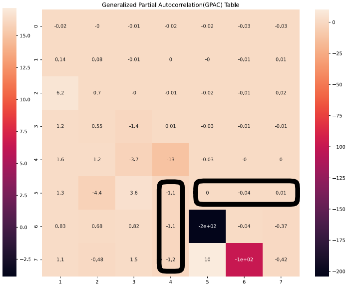
Temperature



Power Consumption



Air Pollution



Domain knowledge vs. Optuna

	Domain Order	MSE	Optuna Order	MSE
AR				
Temperature	(2)	0.038986	(4)	0.038950
Power Consumption	(2)	0.059108	(10)	0.059213
Air Pollution	(1)	0.019484	(5)	0.019482
MA				
Temperature	–	–	(10)	0.038870
Power Consumption	–	–	(10)	0.059218
Air Pollution	–	–	(10)	0.019472
ARMA				
Temperature	(2,3)	0.038971	(8,2)	0.038892
Power Consumption	(2,2)	0.059206	(4,5)	0.059281
Air Pollution	(1,2)	0.019483	(2,6)	0.019458
ARIMA				
Temperature	(1,1,1)	0.049111	(7,1,6)	0.044493
Power Consumption	(1,1,2)	0.055292	(11,1,6)	0.055068
Air Pollution	(4,1,5)	0.019482	(13,1,10)	0.019479

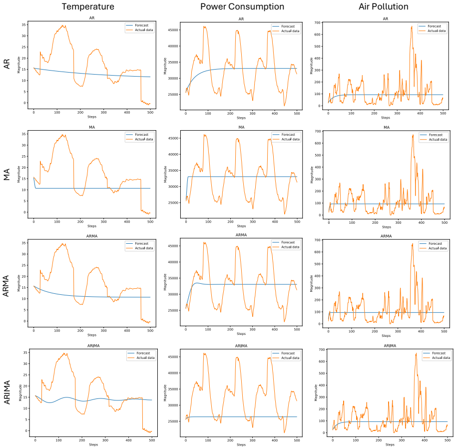
Domain knowledge vs. Optuna

- The minor differences in MSE suggest that while domain knowledge is valuable, Optuna offers a systematic and automated alternative, particularly when prior knowledge is insufficient or when determining an appropriate order is challenging.

Classical models results

- **Model Order Selection:**
  - Selected the order with the lowest MSE from both domain knowledge and Optuna results.
  - In cases with multiple GPAC patterns, the order with the lowest MSE was chosen.
- **Findings:**
  - The performance of classical models was not very satisfactory, which is expected given their simplicity and reliance on basic approaches.
  - These models primarily serve as baseline references for comparison.

Classical models results



Classical models results

	MSE	RMSE	MAE
AR			
Temperature	0.038950	0.197359	0.161282
Power consumption	0.059108	0.243121	0.204268
Air pollution	0.019482	0.139578	0.101504
MA			
Temperature	0.038870	0.197156	0.160967
Power consumption	0.059218	0.243347	0.204480
Air pollution	0.019472	0.139542	0.101515
ARMA			
Temperature	0.038892	0.197210	0.161051
Power consumption	0.059206	0.243322	0.204456
Air pollution	0.019458	0.139493	0.101845
ARIMA			
Temperature	0.044493	0.210933	0.178464
Power consumption	0.055068	0.234666	0.193346
Air pollution	0.019479	0.139569	0.101566

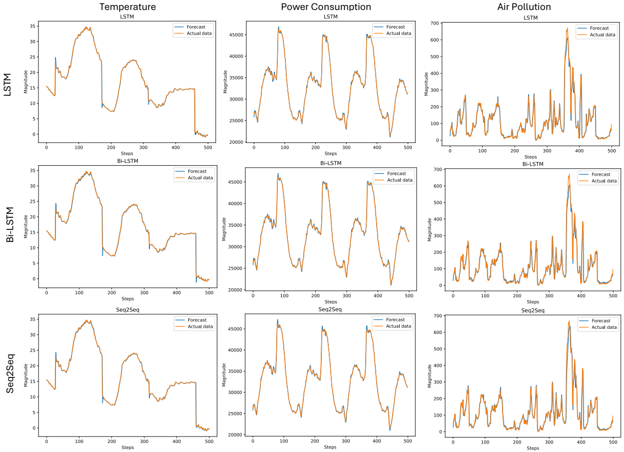
Modern techniques results

- **Training Setup:**
  - All models were trained with identical hyperparameters.
  - This ensures direct comparability across models and isolates the impact of each model's architecture.
- **Findings:**
  - The modern techniques significantly outperformed classical models, with lower MSE values and superior accuracy.
  - Among the models, LSTM, despite its simpler architecture, achieved the lowest MSE, indicating its effectiveness for the datasets in this study.

Modern techniques results

	MSE	RMSE	MAE
LSTM			
Temperature	0.000127	0.011277	0.004017
Power consumption	0.000175	0.013225	0.008923
Air pollution	0.001260	0.035502	0.018459
Bi-LSTM			
Temperature	0.000131	0.011437	0.004310
Power consumption	0.000179	0.013362	0.009071
Air pollution	0.000261	0.035516	0.019128
Seq2Seq			
Temperature	0.000128	0.011307	0.004099
Power consumption	0.000226	0.015046	0.009872
Air pollution	0.000247	0.035312	0.018662

Modern techniques results



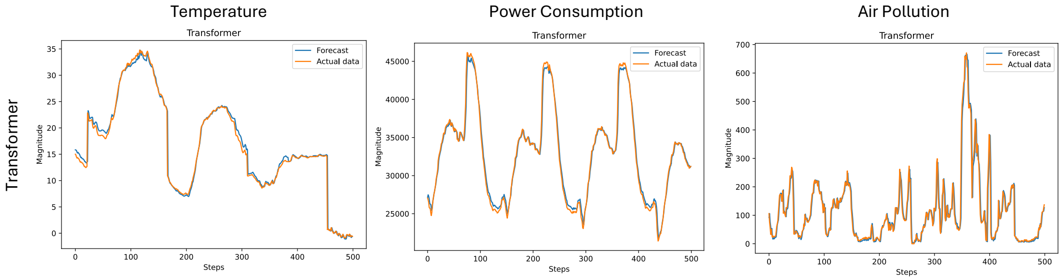
State-of-the-art results

- **Training Setup:**
  - Used the same hyperparameters as the modern techniques, to ensure consistency.
- **Findings:**
  - The Transformer model outperforms the classical models.
  - However, it does not exceed the performance of the three modern techniques (LSTM, BiLSTM, Seq2Seq).
  - This suggests that for these three datasets, the complex Transformer architecture may not be necessary for optimal performance.

State-of-the-art results

	MSE	RMSE	MAE
Temperature	0.000599	0.024466	0.016530
Power consumption	0.000467	0.021602	0.015685
Air pollution	0.001370	0.037011	0.019526

State-of-the-art results



List of Content

1 Overview
2 Methodology
3 Experiment results
4 Conclusion
5 Limitations
6 Future Work

Conclusion

- **Models Evaluated:**
  - Classical Models: AR, MA, ARMA, ARIMA
  - Modern Techniques: LSTM, BiLSTM, Seq2Seq
  - State-of-the-Art: Transformers
- **Performance Comparison:**
  - Modern techniques and Transformers consistently outperformed classical models.
  - LSTM, with its simplest architecture, achieved the lowest MSE.
- **Insights:**
  - Even the simplest deep learning architectures are sufficient to handle these datasets effectively.

List of Content

1 Overview
2 Methodology
3 Experiment results
4 Conclusion
5 Limitations
6 Future Work

Limitations

- **Classical Models:**
  - Simple configurations were selected based on available information, which may not represent the optimal performance.
  - Further exploration could lead to better-performing classical models.
- **GPAC:**
  - Limited the number of rows and columns, which may have prevented us from identifying the correct patterns.
- **Optuna Hyperparameter Tuning:**
  - Maximum potential order and number of trials were constrained, limiting the discovery of more suitable models.
- **Modern Techniques and Transformers:**
  - All models were trained with a fixed sequence length and common hyperparameters, which may not be optimal for every model or dataset.

List of Content

1 Overview
2 Methodology
3 Experiment results
4 Conclusion
5 Limitations
6 Future Work

Future Work

- **SARIMA Models:**
  - Implement Seasonal Autoregressive Integrated Moving Average (SARIMA) models Box et al. 2015 to capture seasonality more effectively.
- **Box-Jenkins Methodology:**
  - Apply the advanced Box-Jenkins methodology Box et al. 2015 for more comprehensive performance comparisons.
- **Model Variations:**
  - Explore other model variations to further refine the findings and broaden the scope of the research.
- **Expanding to Multivariate Time Series:**
  - Currently, the analysis focuses on univariate time series. Future work could extend the models to multivariate time series analysis to capture relationships between multiple variables over time.

References I

- Akiba, Takuya et al. (2019). "Optuna: A next-generation hyperparameter optimization framework". In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631.
- Box, George EP et al. (2015). "Time series analysis: forecasting and control". In.
- Graves, Alex and Jürgen Schmidhuber (2005). "Framewise phoneme classification with bidirectional LSTM and other neural network architectures". In: *Neural networks* 18.5-6, pp. 602–610.
- Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780.
- Mnassri, Baligh (2020). *Weather Station Beutenberg Dataset*. Accessed: 2023-12-11. URL: https://www.kaggle.com/datasets/mnassrib/jena-weather-dataset/data.

References II

- Roy, Rupak (2020). *Air Pollution Forecasting - LSTM Multivariate*. Accessed: 2024-12-02. URL: https://www.kaggle.com/datasets/rupakroy/lstm-datasets-multivariate-univariate.
- Salam, Abdulwahed and Abdelaaziz El Hibaoui (2018). *Power Consumption of Tetouan City*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5B034.
- Sutskever, Ilya, Oriol Vinyals, and Quoc V Le (2014). "Sequence to Sequence Learning with Neural Networks". In: *Advances in Neural Information Processing Systems*. Vol. 27, pp. 3104–3112.
- Vaswani, Ashish et al. (2017). "Attention is all you need". In: *Advances in Neural Information Processing Systems*.