# EXAMINE THE FLEXIBILITY OF ARTIFICIAL NEURAL NETWORK ARCHITECTURES IN DEMAND FORECASTING

**Nham Quoc Hung**

*The University of Texas at Austin*

**December 2024**

## ABSTRACT

Demand forecasting is one of the main applications of time series forecasting. To tackle this challenge, multiple modelling approaches have been developed over the years. Among those, traditional statistical methods based on parametric models such as autoregressive integrated moving average (ARIMA) and exponential smoothing (ETS) as well as supervised learning-based methods from non-parametric models such as support vector machine (SVM), K-nearest neighbors (KNN) and artificial neural networks (ANN) have been widely adopted. Given the complex nature of time series datasets and varying requirements of the prediction tasks, ANN and its variants have emerged as highly capable and adaptable architectures well-suited to solving many pain points above. In this paper, we seek to understand the challenges of demand forecasting and justify why ANN-based architectures are suitable for a wide range of demand forecasting problems through a sample application on the Romania's electricity consumption prediction dataset.

## 1. INTRODUCTION

Time series forecasting has traditionally been a fascinating domain across academia and industry alike. The ability to accurately and reliably predict the future has wide-ranging applications across multiple domains such as commercial sale projection, financial planning, disease control and anomaly detection (AltexSoft, 2022). Among those, deep neural networks have played a pivotal part in expanding the capability of traditional forecasting models due to their ability to learn complex patterns hidden in data while having less reliance on complex feature engineering steps (Lim & Zohren, 2021). At the same time, one major domain which has seen a lot of successes from the application of deep neural networks is demand forecasting (Abbasimehr, Shabani, & Yousefi, 2020). Across critical sectors such as urban planning, supply chain and energy production, predicting future demand is not just a matter of gaining a competitive advantage but also survival of the industry itself, as limited resources often need to be efficiently allocated in advance using historical data as the key indicator (Cash Flow Inventory, 2024).

Our research thus aimed to study and apply various neural network architectures to analyze their effectiveness when applied to demand forecasting on time series data. We also built a tree-based regression model alongside for comparisons. To support these goals, we examined the Kaggle dataset titled *Hourly Electricity Consumption and Production by Type in Romania for 5.5 years* (Comanita, 2024). This dataset contains not only the hourly demand and supply of electricity but also specific contributions from various sources of electricity generation such as nuclear, wind or solar, all of which were used as features in our machine learning and deep learning models.

Our key research questions were centered around the following:

1. Examine existing literature to understand the requirements of demand forecasting as well as strengths and weaknesses of various forecasting techniques
2. Perform exploratory data analysis on the dataset to collect key insights and engineer new meaningful features using traditional time series analysis techniques
3. Build multiple demand forecasting models utilizing traditional as well as neural network architectures to satisfy various electricity consumption prediction requirements
4. Evaluate different models' performances and use cases emphasizing on the flexibility of neural network architectures

## 2. LITERATURE REVIEW

Fundamentally, a time series is a sequence of data points often captured at fixed intervals over some duration of time (Bianchi, 2024). Depending on whether there is a single or multiple values at each time step, a prediction problem can be further categorized into univariate or multivariate (Geron, 2023). At the same time, the prediction horizon can also differ based on the nature of the task. With regards to energy demand forecasting, it is often divided into two main categorizations of short-term versus mid-/long-term (Meira de Oliveira & Cyrino Oliveira, 2018). Correspondingly, single-step and multi-step predictions are techniques required to achieve such forecasts (Brownlee, 2019). Due to the varying characteristics and requirements of each time series forecasting challenge, the methods used to tackle them need to be flexible yet still powerful enough to handle such complexities.

Over time, a wide range of modeling methods have been developed to solve time series problems such as demand forecasting. According to Parmezan et al. (2019), these methods can be categorized into parametric and non-parametric models based on their underlying assumptions of the data distribution. Parametric methods consist of traditional statistical models such as ARIMA (Box, Jenkins, Reinsel, & Ljung, 2015) and exponential smoothing (Brown, 1959; Holt, 1957, Winters, 1960), each of which rely on certain parameters related to autoregression *(p)*, integration *(d)* and moving average *(q)* or level $(\alpha)$, trend $(\beta)$ and seasonality $(\gamma)$ to make accurate predictions. Due to this property, the successes of these models are heavily dependent on deep insights and mathematical understanding of the practitioner on a given problem. Furthermore, in certain cases they often perform less well due to assumptions that the relationship between a time series' underlying components is purely linear (Abbasimehr, Shabani, & Yousefi, 2020). On the other hand, non-parametric methods based on machine learning and deep learning models such as support vector machine (SVM), K-nearest neighbors (KNN) and artificial neural networks (ANN) do not make strong prior assumptions on the dataset. Instead, these models usually require the time series forecasting task to be formulated as a supervised learning problem whereby historical observations at earlier time steps become inputs to predict labels which are subsequent time steps (Brownlee, 2020). From there, they tend to work well with nonlinearity while not requiring an in-depth domain know-how. In certain use cases, hybrid models which combine the strengths of both techniques have also emerged with success (Babu & Reddy, 2014).

Among non-parametric models, ANN is growing in popularity (Lim & Zohren, 2021). Their flexible architectures and ability to encode complex temporal insights into intermediate forms of representation allow for adaptation to different learning requirements. Convolutional neural networks (CNN) are a specific variant of ANN capable of learning from historical data windows for predictions using 1-D causal and dilated convolutional layers (Oord et al., 2016). Similarly, recurrent neural networks (RNN) have long been applicable to modelling sequence data which time series naturally fall under (Petnehazi, 2019). However, it tends to suffer from the vanishing gradient problem which severely hampers training over extended sequences (Bengio et al., 1994). RNN's trouble in dealing with long-term memory has been greatly overcome with the introduction of long short-term memory (LSTM) which relies on certain gates to regulate information flow through memory cells selectively (Greff et al., 2017).

The flexibility from an ANN-based architecture also lies in various other aspects. The input and output layers can be adjusted to model both univariate and multivariate time series prediction with minimal changes to other components (Geron, 2023). Depending on the formulation of the prediction task, ANN can support both regression on continuous outputs and classification on discrete targets through the appropriate activation function (Lim & Zohren, 2021). For multiple-step as opposed to single-step forecasting, ANN can support with relative ease either through autoregressive (Salinas et al., 2020) or sequence-to-sequence (Wen et al., 2018) architectures.

However, despite their adaptability, architectures built purely from neural networks may suffer from certain limitations such as high computational complexity and overfitting (Makridakis, Spiliotis, & Assimakopoulos, 2018). As a result, new hybrid architectures combining both parametric and non-parametric models have emerged (Smyl, 2020). While not examining these architectures directly, we were thus inspired to apply traditional statistical time series analysis techniques in the feature engineering process to further enhance the models under study.

## 3. DATA AND VARIABLES

Our dataset consists of hourly Consumption and Production of electricity in Romania across more than 5 years from 2019-01-01 to 2024-04-01. In this dataset, Production exceeding Consumption signifies electricity export while the reverse implies import (Comanita, 2024). At the same time, Production is further split into several categories ranging from Nuclear, Wind, Hydroelectric, Oil and Gas, Coal, Solar to Biomass. All columns share the same measurement in megawatts (MWs).

In total, there are 46002 distinct hourly measurements across 9 columns over the entire period. Figure 1 displays summary statistics across all data columns.

| | Consumption | Production | Nuclear | Wind | Hydroelectric | Oil and Gas | Coal | Solar | Biomass |
|---|---|---|---|---|---|---|---|---|---|
| count | 46002.000000 | 46002.000000 | 46002.000000 | 46002.000000 | 46002.000000 | 46002.000000 | 46002.000000 | 46002.000000 | 46002.000000 |
| mean | 6587.924286 | 6518.938220 | 1291.157167 | 792.376440 | 1857.216295 | 1171.946611 | 1193.154015 | 156.718686 | 55.852202 |
| std | 1043.513924 | 986.662774 | 236.568151 | 675.836308 | 692.555331 | 434.757973 | 320.464170 | 229.514634 | 14.236465 |
| min | 3889.000000 | 3315.000000 | 562.000000 | -26.000000 | 175.000000 | 198.000000 | 279.000000 | 0.000000 | 17.000000 |
| 25% | 5773.000000 | 5814.000000 | 1347.000000 | 236.000000 | 1347.000000 | 858.000000 | 962.000000 | 0.000000 | 45.000000 |
| 50% | 6552.000000 | 6462.500000 | 1383.000000 | 592.000000 | 1747.000000 | 1211.000000 | 1172.000000 | 2.000000 | 57.000000 |
| 75% | 7321.750000 | 7176.000000 | 1405.000000 | 1205.750000 | 2265.000000 | 1511.000000 | 1406.000000 | 281.000000 | 67.000000 |
| max | 9615.000000 | 9886.000000 | 1457.000000 | 2811.000000 | 4434.000000 | 2141.000000 | 2537.000000 | 1137.000000 | 89.000000 |

Figure 1: Summary statistics across all original data columns

Based on these available features, our goal was to perform a wide range of demand forecasting tasks on electricity consumption such as univariate, multivariate, short-term and longer-term predictions to examine and verify the adaptability of ANN architectures on time series data. Such applications can potentially benefit multiple stakeholders such as power grid operators, energy producers, government agencies (Dupont, 2023) and utility companies (Miraki et al., 2024). While short-term forecasts are expected to be more accurate and allow for more responsive adjustments, longer-term forecasts can be critical to allow for more strategic planning such as development and policy making (Meira de Oliveira & Cyrino Oliveira, 2018). The success of these forecasts carries wide-ranging impacts from enhancing grid management (Miraki et al., 2024), ensuring adequate supply (Hu et al., 2024), preventing blackouts and disruptions of economic activities (Elamin & Fukushige, 2018) to promoting sustainable energy initiatives (Williams & Short, 2020).

## 3.1 EXPLORATORY DATA ANALYSIS

Exploratory data analysis was first taken to examine the data and variables. As shown in Figure 2, except for the DateTime column, the dataset contains 9 other numerical columns that need to be analyzed before embarking on the modeling process. At the same time, the DateTime column is of a fixed hourly frequency over a 5-year period, hence rendering the remaining columns as distinct time series data.

| DateTime | Consumption | Production | Nuclear | Wind | Hydroelectric | Oil and Gas | Coal | Solar | Biomass |
|---|---|---|---|---|---|---|---|---|---|
| 2019-01-01 00:00:00 | 6352 | 6527 | 1395 | 79 | 1383 | 1896 | 1744 | 0 | 30 |
| 2019-01-01 01:00:00 | 6116 | 5701 | 1393 | 96 | 1112 | 1429 | 1641 | 0 | 30 |
| 2019-01-01 02:00:00 | 5873 | 5676 | 1393 | 142 | 1030 | 1465 | 1616 | 0 | 30 |
| 2019-01-01 03:00:00 | 5682 | 5603 | 1397 | 191 | 972 | 1455 | 1558 | 0 | 30 |
| 2019-01-01 04:00:00 | 5557 | 5454 | 1393 | 159 | 960 | 1454 | 1458 | 0 | 30 |

Figure 2: DateTime and other 9 numerical columns

Figure 3 then gave an overview of all the original time series in distinct colors. We further zoomed in to study the correlation across different variables.

Figure 4 is a correlation matrix across every pair of variables. As shown, Consumption has a strong positive relationship with Production, along with a moderately positive relationship with Oil and Gas and Coal. This is perhaps because electricity supply must be responsive to any changes in demand while Oil and Gas and Coal were the main resources used by the Romanian government to adjust their electricity production in response to consumption.

Figure 5 further confirms the strong positive relationship after resampling hourly Consumption and Production by day, plotting their simple moving average across a 30-day window and then stacking them together. We could observe that these time series share some similarities in pattern such as peaking around the same period each year. Figure 6 presents the data distribution across all available variables. Consumption, Production and Coal roughly follow a normal distribution.

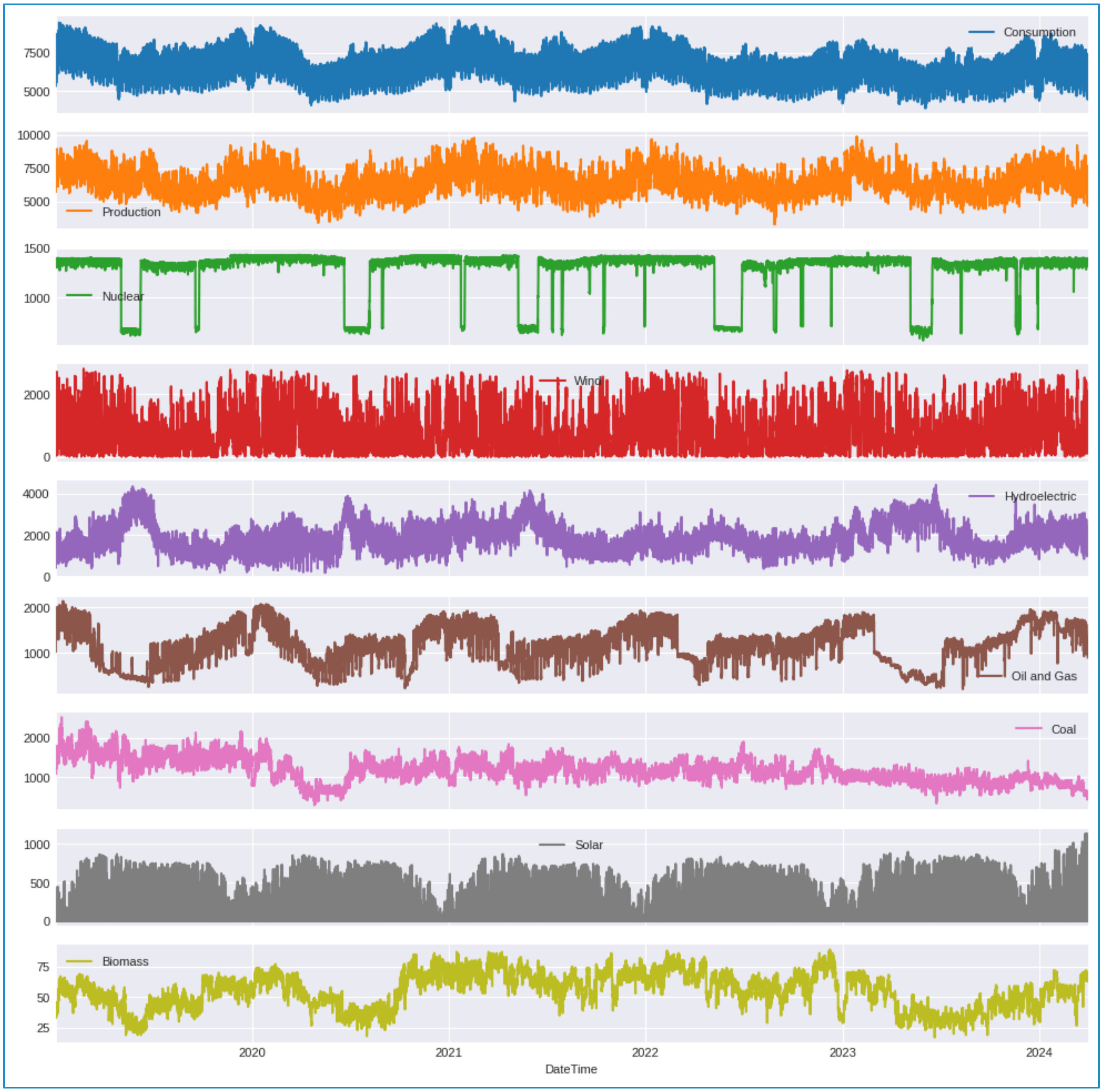


Figure 3: Overview of all original time series

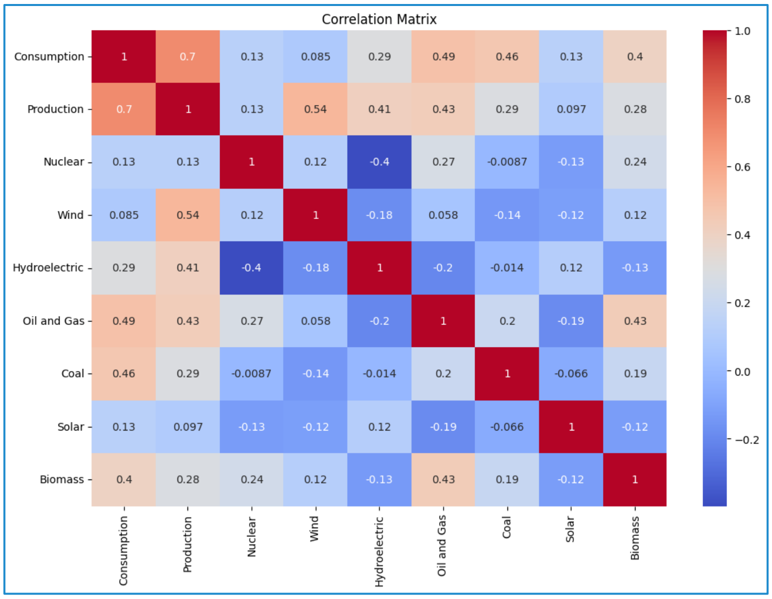


Figure 4: A correlation matrix across all variable pairs

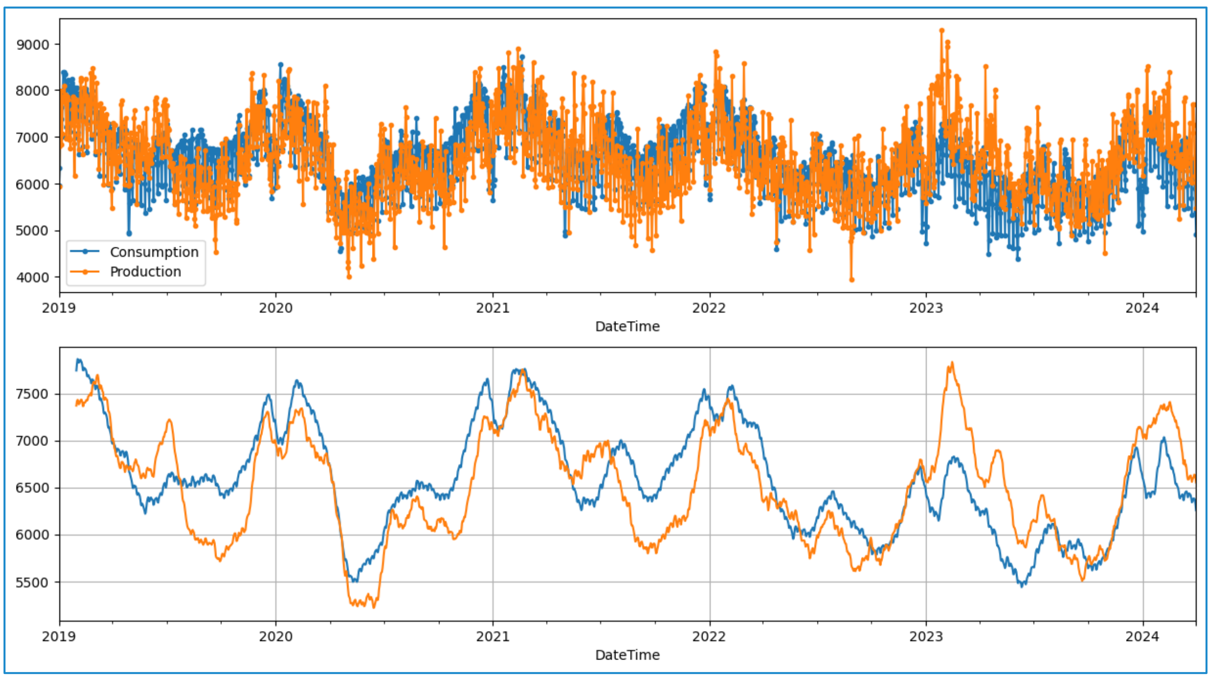


Figure 5: Daily and 30-day moving average for Consumption and Production

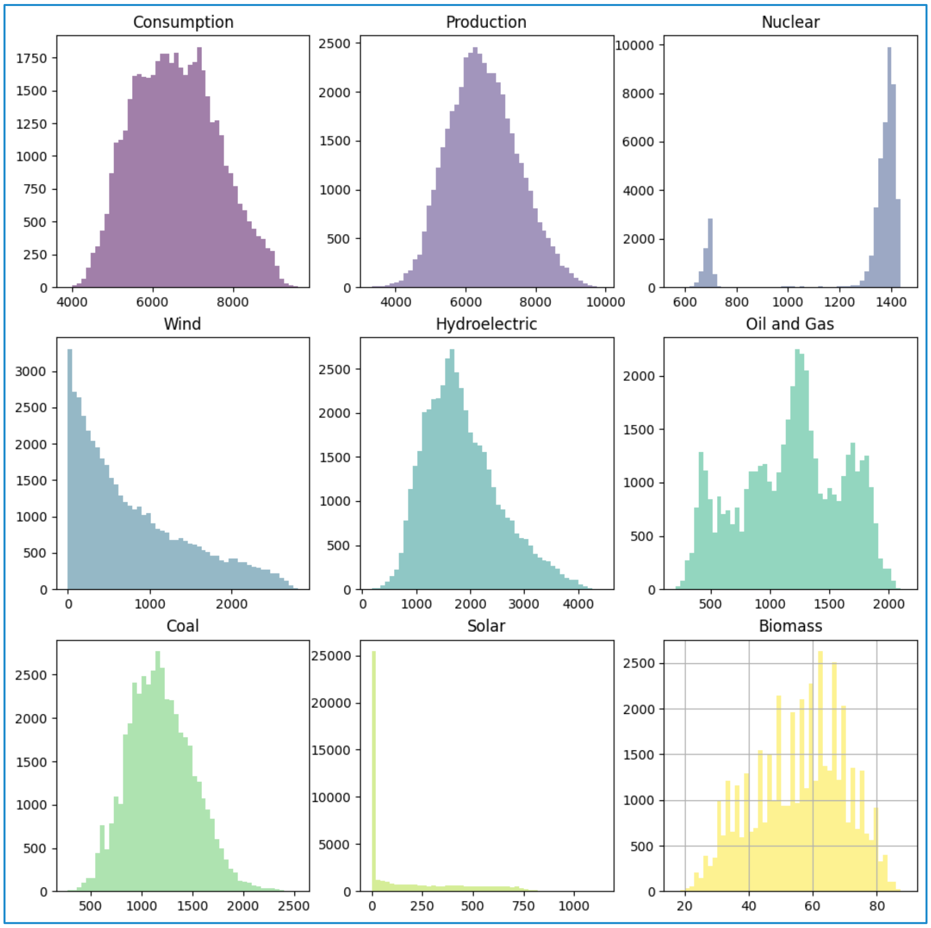


Figure 6: Data distribution across all variables

We then zoomed in to our dependent variable Consumption for further time series-related analysis. Stationary check is a common technique to prepare a time series for traditional forecasting models (Bianchi, 2024). Many of those models are only effective conditioned on a strong assumption that the underlying time series is stationary. A time series is said to be stationary when its statistical properties are independent of the time of observation. In other words, it should not contain any trend or seasonality as they could render the statistics of different sets of observations inconstant when they are sampled from different points in time. We applied the following methods to check for characteristics such as seasonality and trend:

- Visualize average hourly, daily and monthly Consumption
- Visualize Consumption's autocorrelation graph
- Visualize Consumption's seasonal decomposition graphs

Figure 7 indicates some potential insights into Consumption patterns given hourly, daily and monthly aggregations. The hourly average plot suggests that peak daily Consumption is at 9AM and 8PM while its trough is at 3AM. This can be explained as most people start work, reach home and go to bed following the above time points. At the same time, weekdays seem to maintain much higher Consumption levels than weekends, potentially due to work-related activities. Throughout the year, January shows the highest Consumption while May and September display the lowest. This can be explained by stronger demand for heating during winter and more moderate demand during warmer months.

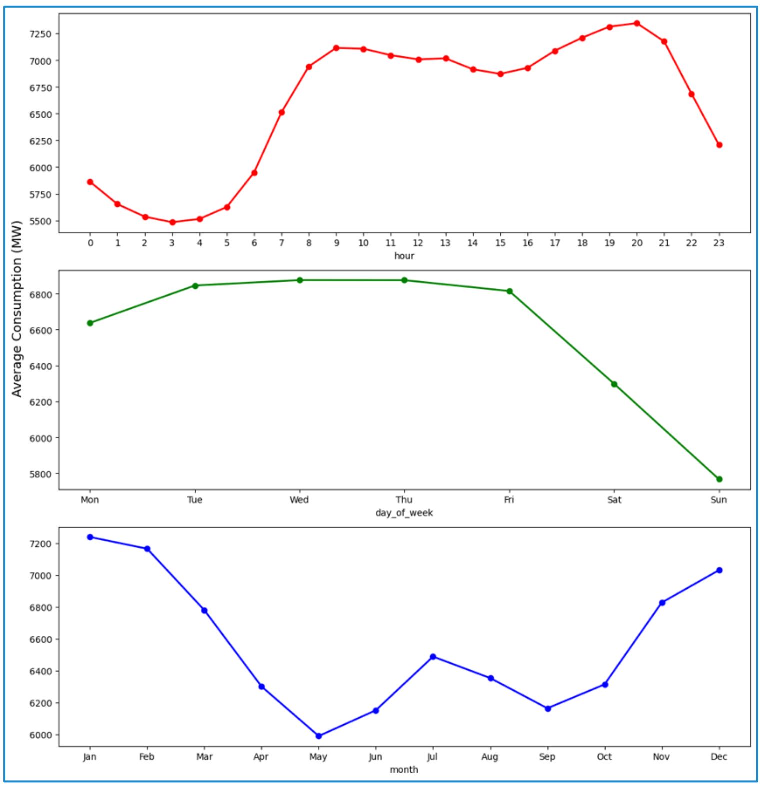


Figure 7: Hourly, daily and monthly average Consumption

Figure 8 showcases seasonality in greater detail by using autocorrelation function (Hyndman & Athanasopoulos, 2021). Autocorrelation measures the degrees of correlation between a time series' current and one of its past measurements specified by a *lag* parameter (Flores et al., 2012). In our case, the autocorrelation function from the *statsmodels* package (Seabold, Skipper & Perktold, 2010) can compute and plot autocorrelation values for different lag hours to identify seasonality through its peaks and troughs. As observed, Consumption has a strong daily and weekly seasonality due to its hourly and daily autocorrelation values peaking at every

24 hours and 7 days respectively. Meanwhile, it has no monthly and a weak yearly (12-month lag) seasonality.

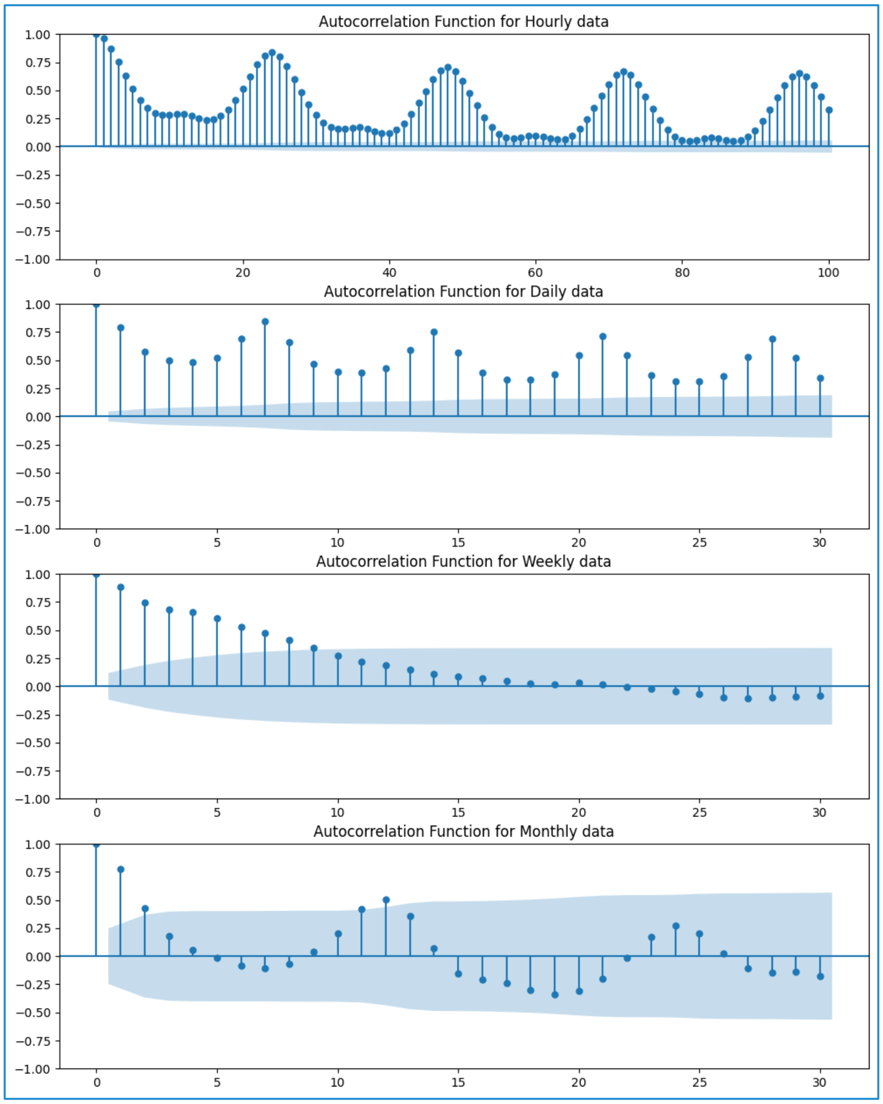


Figure 8: Hourly, daily, weekly and monthly autocorrelation graphs

This can be further verified by Figure 9 which displays seasonal decomposition (Hyndman & Athanasopoulos, 2021) of hourly Consumption into trend, seasonality and residual components over a sample period from 2019-01-01 to 2019-02-01, using an *additive* model with a *period* value of 24 from the *statsmodels* package. This decomposition model assumes that a time series can be broken down and reconstructed from the sum of the 3 components above (Parmezan et al., 2019). As such, this helps to ensure the residual component is stationary thanks to the remove of trend and seasonality. As shown in Figure 9, daily seasonality is clearly visible.

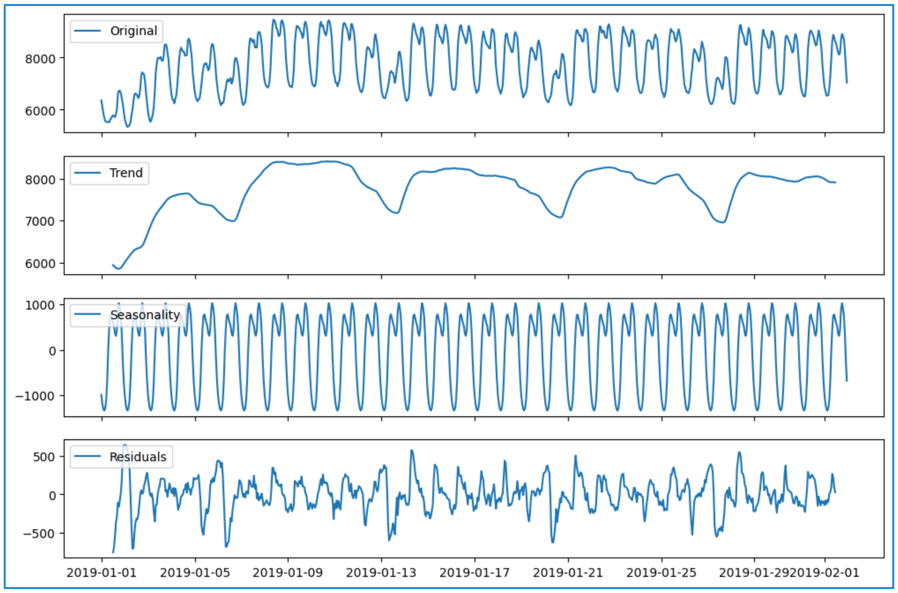


Figure 9: Additive seasonal decomposition of Consumption

## 3.2. FEATURE ENGINEERING

It has been mentioned that for time series forecasting, the inclusion of exogenous variables is crucial to account for external influences on the target variable prediction (Wang et al., 2024). To prepare the dataset for training, several additional exogenous features were thus added. First, due to the presence of periodic patterns across different time granularities found previously, certain time-related features such as *month*, *week*, *day_of_week* and *hour* were extracted as ordinal values from the original DateTime column using the *Feature-engine* package (Galli, 2021). However, the cyclical nature of these features needs to be captured more accurately through sine and cosine transformations which represent the correct difference between the last and first ordinal values (Amat Rodrigo & Escobar Ortiz, 2023). Figure 10 highlights sample plots of the new *hour_sin* and *hour_cos* features after Sine and Cosine transformations on *hour*.

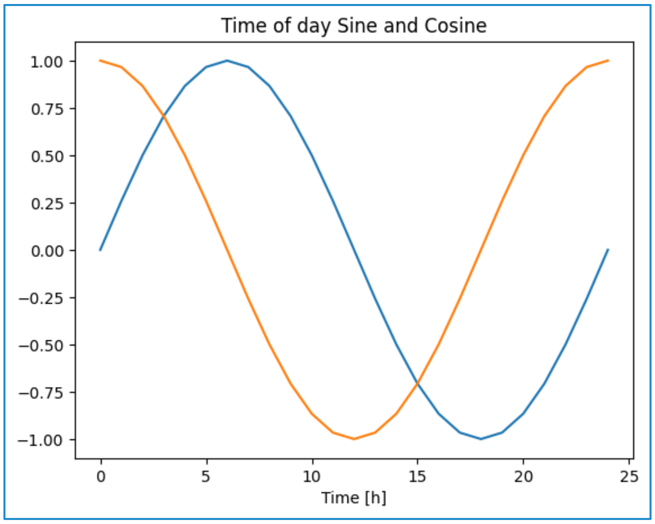


Figure 10: Sine and cosine transformation of *hour* column

Furthermore, section 3.1 showed that Consumption is clearly seasonal and thus non-stationary. For many traditional forecasting models such as ARIMA, this would violate their assumptions and affect their prediction quality. For our tree-based regression and neural network models, we thus included 3 additional columns named *Trend*, *Seasonal* and *Residual* from the additive decomposition of Consumption with a period of 24 mentioned in section 3.1. This proved to help enrich these models further.

## 3.3 PREPROCESSING

After feature engineering, the dataset consisted of 20 features. We then performed standardization on all the features as a preprocessing step before modelling due to the large difference in their value ranges (Jaadi, 2023). Figure 11 showcases the data distribution of these columns after standardization. It was noted that certain features such as Nuclear, Wind, Solar and Seasonal had slightly skewed distributions but no further preprocessing step was performed as the skewness did not seem overly severe.

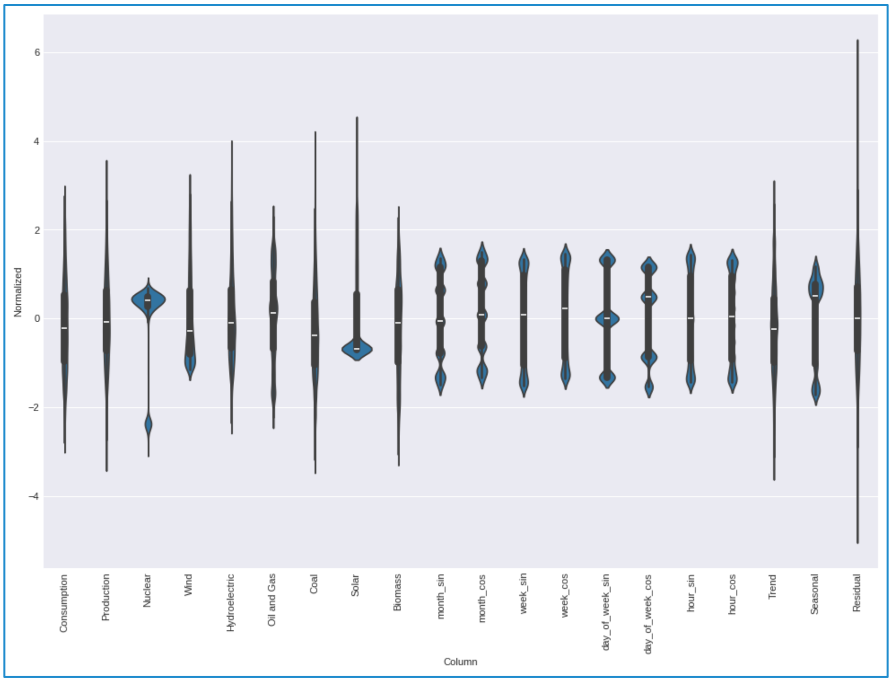


Figure 11: Data distribution of all engineered features

## 4. METHODS

To evaluate different architectures over varying demand forecasting requirements, we mainly categorized these requirements into single-step and multiple-step prediction scenarios. 70% of the data was used for training, 20% for validation and 10% for testing as shown in Figure 12 below.

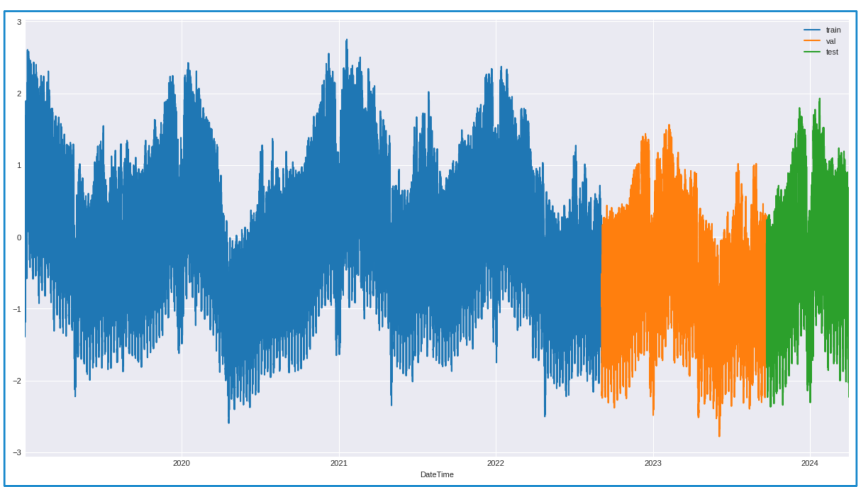


Figure 12: Train, validation and test data for Consumption

## 4.1 SINGLE-STEP PREDICTION

For short-term demand forecasting, we first focused on making univariate Consumption forecast over the next hour alone. Knowing that Consumption has a strong daily seasonality, a common data window (Time series forecasting, n.d.) was used across all models whereby 24 hours of historical time steps became input and the next hour's Consumption was the label. This 24-hour input window enabled these models to observe, capture and learn important temporal patterns over a period most relevant to the next prediction.

The regression model was constructed using Skforecast library (Amat Rodrigo & Escobar Ortiz, 2023) with LightGBM, a gradient-boosted and tree-based algorithm with many optimizations to speed up training and reduce computational overhead (Mandot, 2017), as the main regressor. Hyperparameter tuning was performed to decide the best combination of lags, number of estimators, max depth, learning rate and regularization terms. Bayesian Search was used to optimize tuning by gradually reducing the search space (Amat Rodrigo & Escobar Ortiz, 2023). Figure 13 showcases the outcome of our hyperparameter search whereby 24-hour lags were shown to give the lowest mean absolute error (MAE) on validation data.

| | lags | params | mean_absolute_error | n_estimators | max_depth | learning_rate | reg_alpha | reg_lambda |
|---|---|---|---|---|---|---|---|---|
| 0 | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14... | {'n_estimators': 400, 'max_depth': 4, 'learnin... | 0.094181 | 400.0 | 4.0 | 0.040886 | 0.2 | 0.1 |
| 1 | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12] | {'n_estimators': 400, 'max_depth': 4, 'learnin... | 0.105914 | 400.0 | 4.0 | 0.048325 | 0.1 | 0.1 |
| 2 | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12] | {'n_estimators': 500, 'max_depth': 3, 'learnin... | 0.118984 | 500.0 | 3.0 | 0.030417 | 0.2 | 0.2 |

Figure 13: Hyperparameter tuning results

Once the best hyperparameters have been found, backtesting (Lewinson, 2023) was used to evaluate the model performance on test data using MAE as the metric. We then obtained the model's predictions and plotted predicted Consumption (normalized) versus actual over a sample test period of 7 days for evaluation. Figure 14 represents the plot and shows that predicted Consumption closely aligns with actual values.

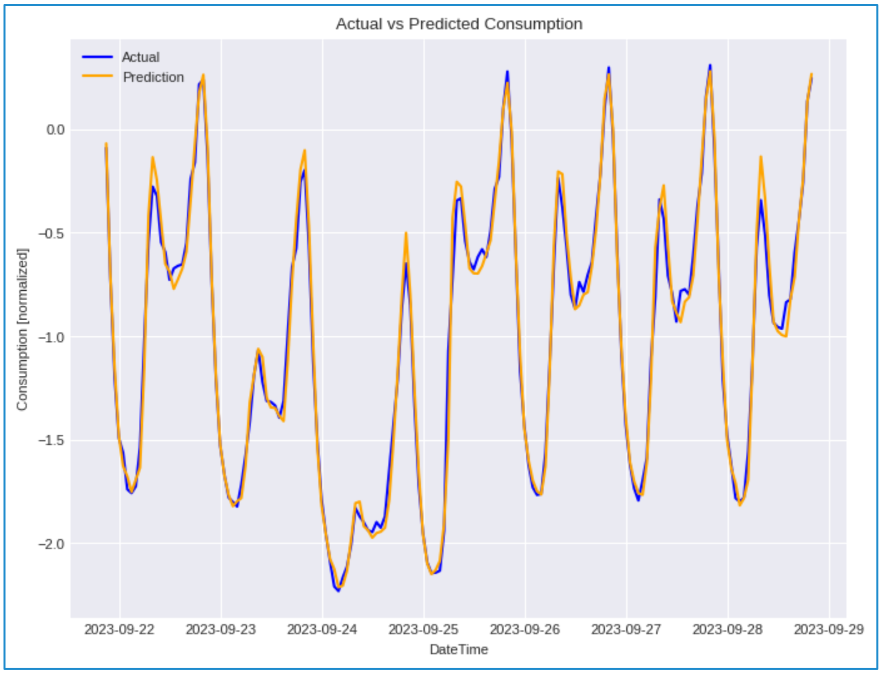


Figure 14: Predicted versus actual Consumption [normalized] for regression model (1 step)

From the trained tree-based model, we retrieved feature importance rankings (Scikit-learn, n.d.) to analyze the best predictors for Consumption. Figure 15 shows that *Trend, lag_1, Residual, Seasonal* and *hour_sin* are the top 5 most important features. This signifies that combining traditional statistical time series analysis techniques for feature engineering helped with improving the performance of our regression and ANN-based models.

| | feature | importance |
|---|---|---|
| 28 | Trend | 1082 |
| 0 | lag_1 | 968 |
| 30 | Residual | 728 |
| 29 | Seasonal | 659 |
| 26 | hour_sin | 470 |
| 18 | Solar | 279 |
| 1 | lag_2 | 275 |
| 27 | hour_cos | 206 |
| 23 | week_cos | 122 |
| 10 | lag_11 | 109 |

Figure 15: Feature importance rankings from tree-based regression model

Afterwards, 3 different single-step, ANN-based architectures were built for a similar purpose. The first model used purely Dense hidden layers with ReLU activation function. The second model used a combination of a 1-dimensional causal Convolutional layer with ReLU activation function followed by a sequence-to-vector LSTM layer. This architecture was chosen so that the Convolutional layer could extract different temporal patterns in each window through its filters while the LSTM layer processes and retains only relevant information to produce a single output on the final time step (Geron, 2023). The final model replaced the causal Convolutional layer with multiple dilated Convolutional layers similar to the WaveNet architecture (Oord et al., 2016) to potentially enhance its learning ability over a longer window while keeping the same LSTM layer. All 3 models used a single-unit Dense output layer to enable their single-step prediction.

All neural network models used Huber loss with Adam optimizer and MAE as their metric. They were trained on maximum 50 epochs but with early stopping callback checking for the lowest validation loss and a patience value of 5. Figure 16 shows training, validation and test losses for the purely Dense model while Figure 17 displays its single-step predictions versus actual labels on 3 sample data windows.

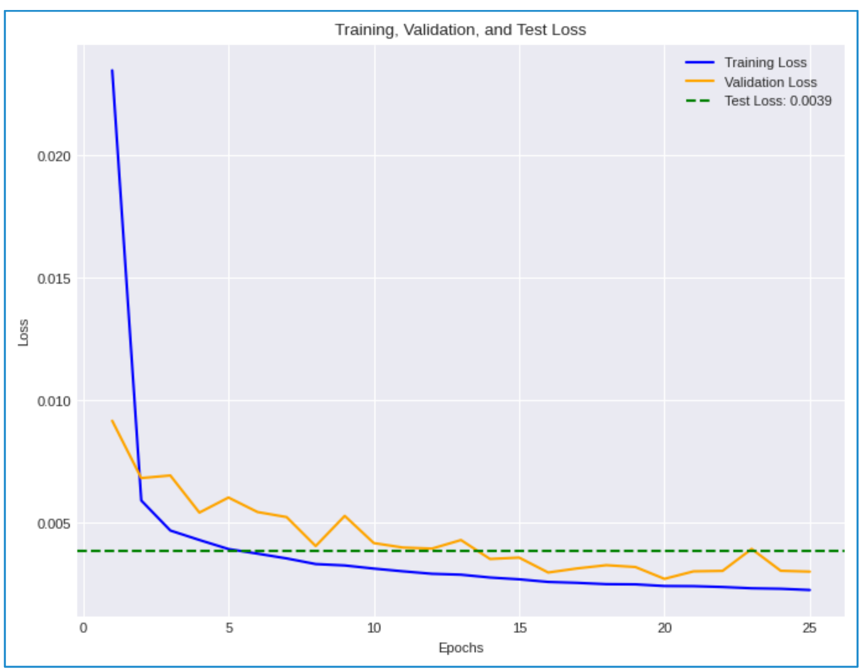


Figure 16: Training, validation and test loss for purely Dense model

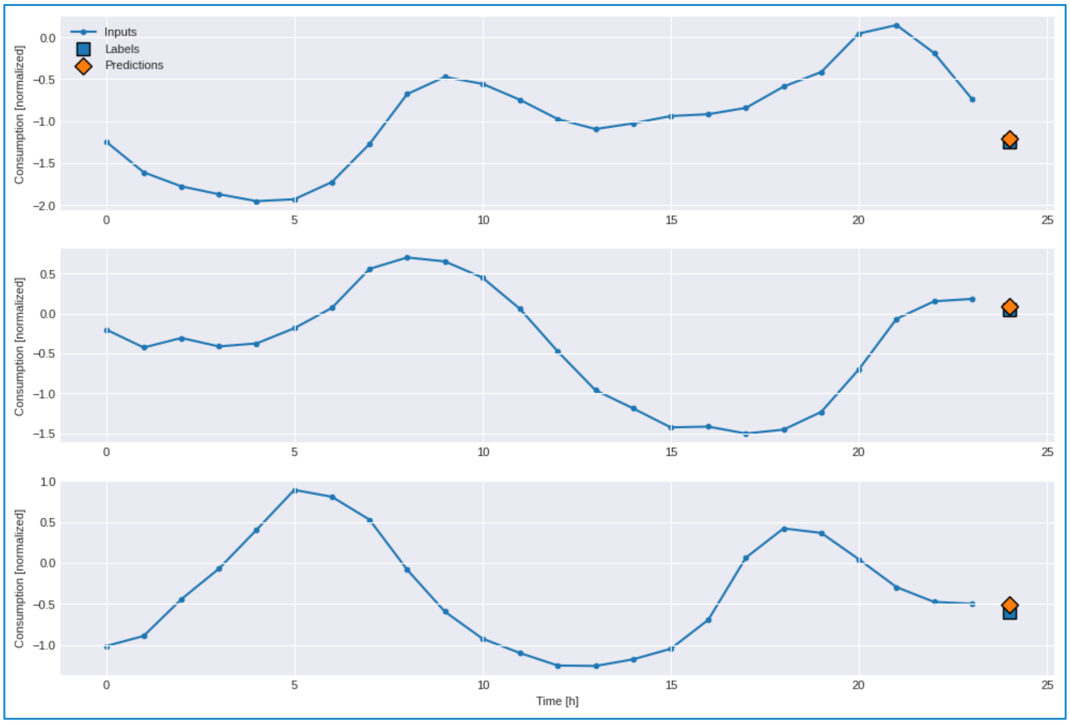


Figure 17: Purely Dense model's predictions versus actual labels on 3 sample data windows

## 4.1 MULTIPLE-STEP PREDICTION

For longer-term demand forecasting, different tree-based regression and neural network models were built with certain extensions or modifications to previous architectures to examine their adaptability. Most notably, while still using 24 previous hourly data as input, each model would need to generate multivariate predictions on multiple target variables and for the next 24 hours in the future. These requirements are much more complex than purely univariate predictions which traditional statistical methods are often applied to (Hewamalage, Bergmeir, & Bandara, 2021). Yet, they provide greater insight for stakeholders by allowing them to plan further ahead.

First, the tree-based regression model using Skforecast was modified to make direct multiple-step forecasting. This was achieved by performing all the hyperparameter tuning and backtesting similarly as before except for the *steps* parameter being set as 24. However, despite this minimal change, Skforecast's underlying mechanism is to train a different model for each time step within the next 24-hour horizon (Amat Rodrigo & Escobar Ortiz, 2023). As a result, the computation cost increased significantly. Fortunately, performance did not suffer nearly as much due to the relatively short prediction horizon. Figure 18 plots actual versus predicted Consumption for this model over a sample test period of 7 days.

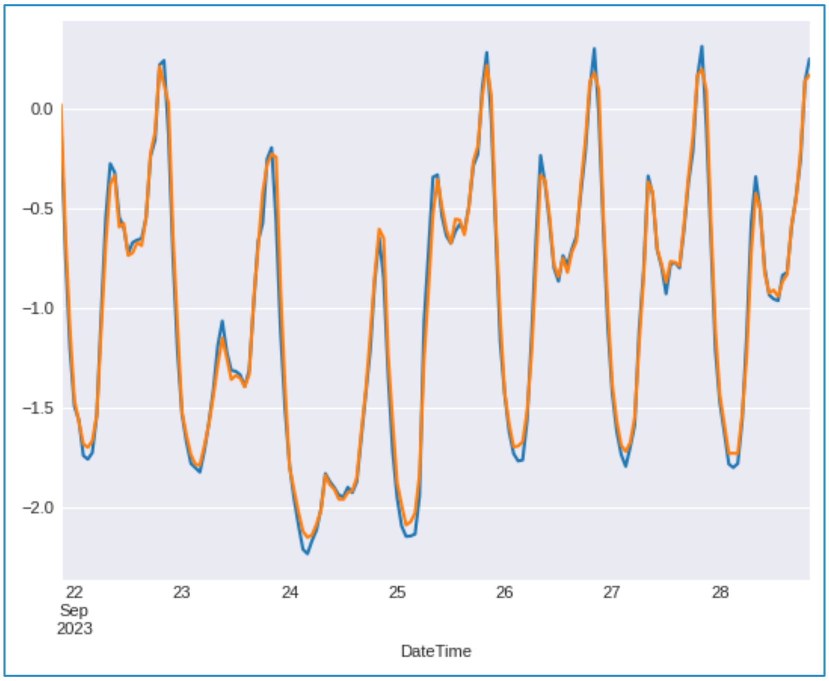


Figure 18: Predicted versus actual Consumption [normalized] for regression model (24 steps)

Meanwhile, neural network architectures can accommodate direct multiple-step prediction more naturally by modifying the number of units in its output Dense layer. Based on this, 3 different multiple-step, multiple-target and ANN-based architectures were built. These models are flexible enough to make longer-term predictions on different targets such as Production or even *Residual*. The first model utilized a combination of causal Convolutional and Dense layers while the second combined causal Convolutional with LSTM layers. Meanwhile, the final model made use of Autoregressive LSTM to produce a single-step prediction each time and feed this back as input for the next predictions repeatedly. The benefit of this Autoregressive model is that it can produce a projection sequence of arbitrary length (Time series forecasting, n.d.).

Figures 19 and 20 display the Autoregressive LSTM model's performance and predictions for the next 24 Consumption hours over 3 random data windows. We observed a light divergence between training and validation loss at the final epochs which signified this model was slightly overfitting.

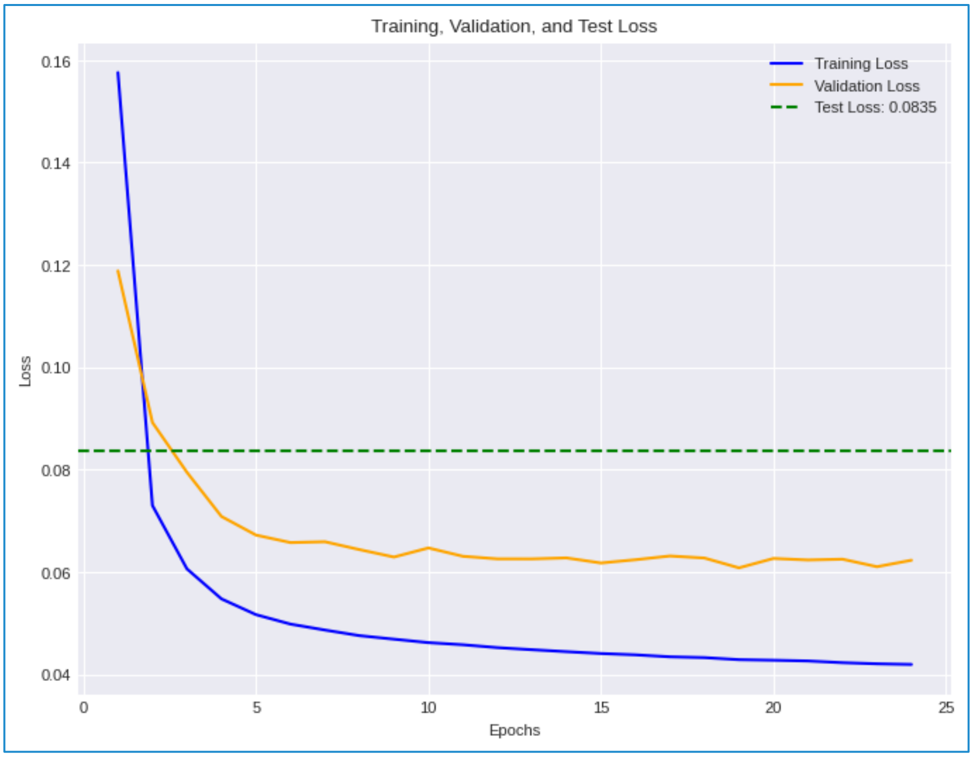


Figure 19: Training, validation and test loss for Autoregressive LSTM model

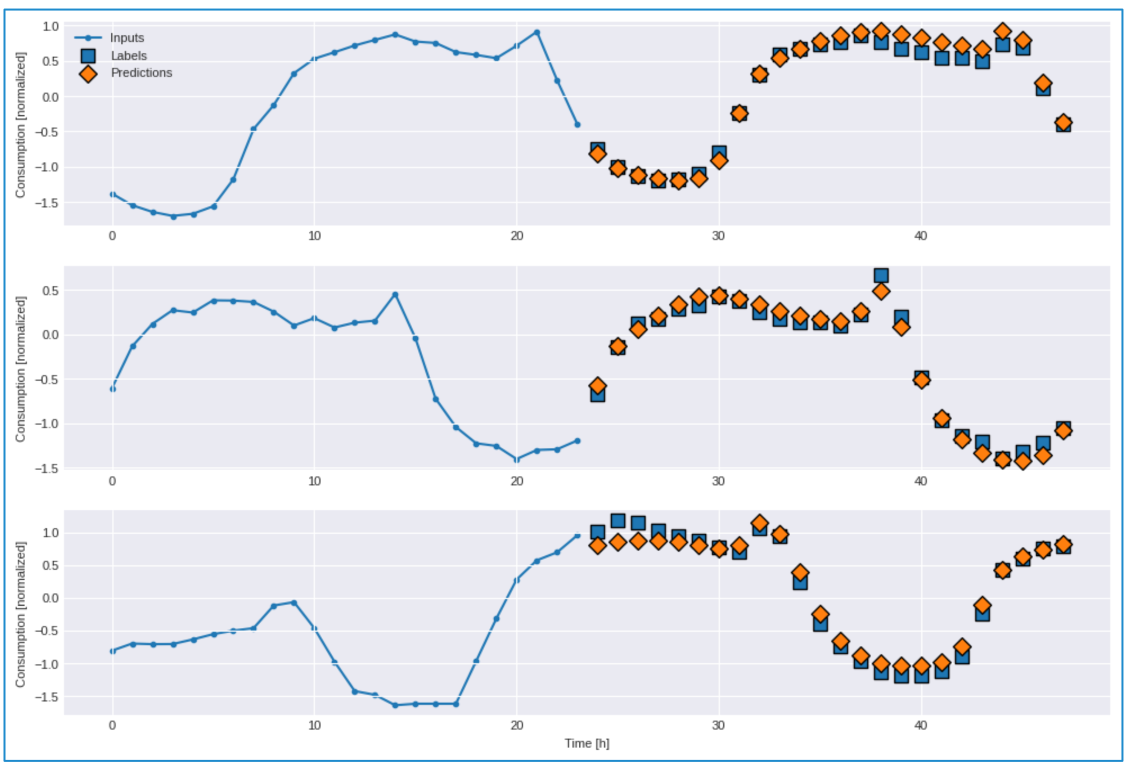


Figure 20: Autoregressive LSTM model's predictions versus actual labels on 3 sample windows

# 5. RESULTS

## 5.1 SINGLE-STEP PREDICTION

Figure 21 highlights validation and test evaluation results among single-step prediction models. The LightGBM regression model performed best with the lowest test MAE. This could be explained as this model underwent a more extensive hyperparameter tuning process while other neural network models did not due to time constraints. Despite that, the differences between them were not significant, highlighting neural networks' ability to work on time series prediction tasks quite effectively. Among ANN-based architectures, the purely Dense model had the lowest validation and test MAE, followed by causal Convolutional with LSTM and finally dilated Convolutional with LSTM. The reason the Dense model outperformed other variants could be attributed to the dataset being small and feature engineering steps being able to capture major temporal information. Meanwhile, LSTM usually excels when temporal relationships are more complex, nonlinear and over a longer term. On the other hand, there could be various potential reasons to explain the dilated Convolutional with LSTM's poorer performance such as not having a sufficiently large enough training window or dataset to learn meaningful long-term temporal patterns.

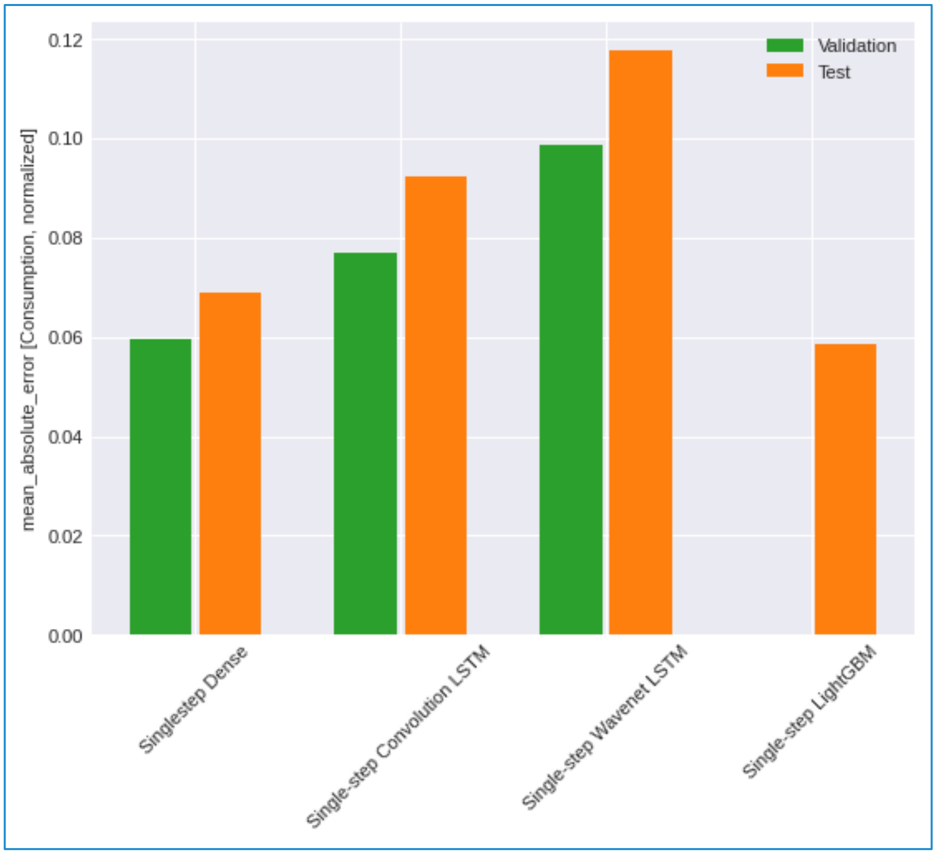


Figure 21: Validation and test MAE across 4 models for single-step prediction

## 5.2 MULTIPLE-STEP PREDICTION

Figure 22 shows our evaluation results across multiple-step prediction models. As explained above, the LightGBM model performed much better due to its underlying mechanism of using 24 separate single-step regressors. This would not be sustainable for prediction over a much longer horizon. Meanwhile, the Convolutional with Dense model performed slightly better than Convolutional with LSTM and Autoregressive LSTM. Among the 2 LSTM-based models, Autoregressive had a slightly lower validation MAE but did not generalize as well on the test set. Still, its flexibility in enabling projections of varying length into the future could be especially helpful for certain stakeholders. Overall, all 4 models had worst performance than their single-step counterparts due to a greater unpredictability involved in long-term forecasting.

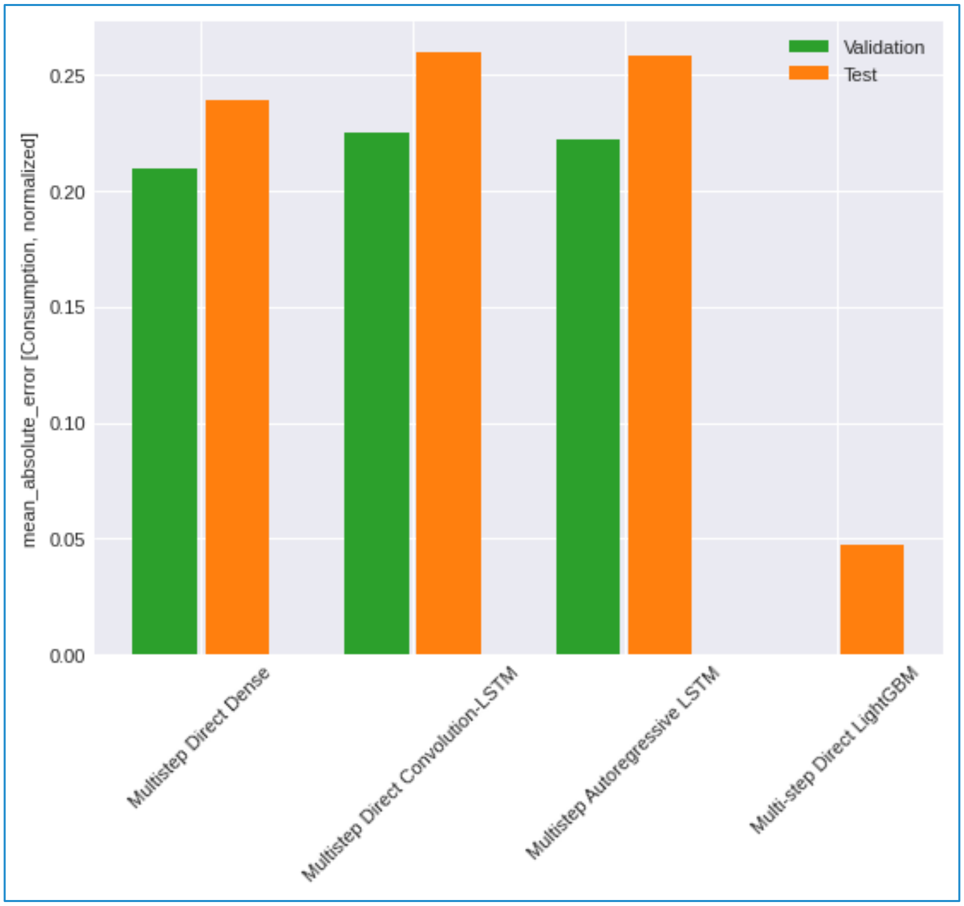


Figure 22: Validation and test MAE across 4 models for multiple-step, multiple-target prediction

## 6. CONCLUSION

In this study, we focused on the flexibility of neural network architectures in working with time series data such as in demand forecasting tasks. Through an application on the electricity consumption forecasting dataset, we highlighted their ability to handle an extensive range of prediction requirements that are useful for stakeholders in the energy sector. The study's key findings included several components. First, we surveyed existing literature on common time series problems and how various traditional statistical, machine learning and neural network models have been applied to solving them. Then, we performed exploratory data analysis and utilized traditional time series analysis techniques through feature engineering to cover certain limitations of ANN-based architectures. Then, by formulating demand forecasting as a supervised learning problem, we successfully applied LightGBM along with multiple ANN variants such as purely Dense, Convolutional and LSTM for short-term, longer-term, univariate and multivariate predictions with reasonable performance results. Lastly, based on those results, we highlighted the adaptability of ANN architectures through comparisons with powerful tree-based regression models such as LightGBM.

Having said that, the study faced certain limitations. Given the time and resource constraints, we were unable to do very comprehensive hyperparameter tuning for different models and time windows, thus potentially limiting their predictive ability and performance results especially for ANN variants. Furthermore, certain architectures such as dilated Convolutional or LSTM may perform better on longer and more complex sequences or training time, hence the nature of our dataset could be a limiting factor to truly evaluating their expressive power. Lastly, more hybrid models could be examined to truly showcase and justify the potential of ANNs. In future studies, we hope to continue resolving these issues.

## REFERENCES


1. Abbasimehr, H., Shabani, M., & Yousefi, M. (2020). An optimized model using LSTM network for demand forecasting. *Computers & Industrial Engineering, 143,* 106435. https://doi.org/10.1016/j.cie.2020.106435
2. Altexsoft. (2022, March 21). *Time Series Analysis and Forecasting: Examples, Approaches, and Tools*. Altexsoft. https://www.altexsoft.com/blog/time-series-analysis-and-forecasting-novel-business-perspectives
3. Amat Rodrigo, J., & Escobar Ortiz, J. (2024). *skforecast* (Version 0.14.0) [Computer software]. BSD-3-Clause. https://doi.org/10.5281/zenodo.8382788
4. Bianchi, F. M. (2024). *Time Series Analysis with Python*. GitHub. https://github.com/FilippoMB/python-time-series-handbook
5. Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control*. New Jersey: John Wiley & Sons.
6. Brown, R. G. (1959). *Statistical forecasting for inventory control*. McGraw/Hill.
7. Brownlee, J. (2019, August 21). *4 Strategies for Multi-Step Time Series Forecasting*. Machine Learning Mastery. https://machinelearningmastery.com/multi-step-time-series-forecasting
8. Brownlee, J. (2020, August 15). *Time Series Forecasting as Supervised Learning*. Machine Learning Mastery. https://machinelearningmastery.com/time-series-forecasting-supervised-learning
9. Cash Flow Inventory. (2024, February 12). *Demand Forecasting – Importance, Methods & Best Practices*. Cash Flow Inventory. https://cashflowinventory.com/blog/demand-forecasting
10. Comanita, S. (2024, April 02). *Hourly Electricity Consumption and Production*. Kaggle. https://www.kaggle.com/datasets/stefancomanita/hourly-electricity-consumption-and-production/data
11. Dupont, F. (2023, October 18). *Understanding Stakeholders in the Energy and Public Utilities sector.* Borealis. https://www.boreal-is.com/blog/stakeholders-in-energy-and-utilities-sector
12. Elamin, N., & Fukushige, M. (2018). Quantile Regression Model for Peak Load Demand Forecasting with Approximation by Triangular Distribution to Avoid Blackouts.

*International Journal of Energy Economics and Policy*, *8*(5), 119–124. Retrieved from https://www.econjournals.com/index.php/ijeep/article/view/6742

13. Flores, J. H. F., Engel, P. M., & Pinto, R. C. (2012). Autocorrelation and partial autocorrelation functions to improve neural networks models on univariate time series forecasting. In *The 2012 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8). https://doi.org/10.1109/IJCNN.2012.6252470
14. Galli, S. (2021). Feature-engine: A Python package for feature engineering for machine learning. *Journal of Open Source Software, 6*(65), 3642. https://doi.org/10.21105/joss.03642
15. Geron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras & Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems* (3rd ed.). O'Reilly Media, Inc.
16. Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems, 28*(10), 2222–2232. https://doi.org/10.1109/TNNLS.2016.2582924
17. Holt, C. C. (1957). *Forecasting seasonals and trends by exponentially weighted averages* (ONR Memorandum No. 52). Carnegie Institute of Technology, Pittsburgh USA. Reprinted in the *International Journal of Forecasting*, 2004. https://doi.org/10.1016/j.ijforecast.2003.09.015
18. Hu, H., Gong, S., & Taheri, B. (2024). Energy demand forecasting using convolutional neural network and modified war strategy optimization algorithm. *Heliyon, 10*(6), e27353. https://doi.org/10.1016/j.heliyon.2024.e27353
19. Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: principles and practice*, 3rd edition, OTexts: Melbourne, Australia. OTexts.com/fpp3. Accessed on 15 November 2024.
20. Jaadi, Z. (2023, August 04). *When and Why to Standardize Your Data*. Builtin. https://builtin.com/data-science/when-and-why-standardize-your-data
21. Lewinson, E. (2023, November 08). *Putting Your Forecasting Model to the Test: A Guide to Backtesting*. Towards Data Science. https://towardsdatascience.com/putting-your-forecasting-model-to-the-test-a-guide-to-backtesting-24567d377fb5

22. Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLoS ONE, 13*(3), e0194889. https://doi.org/10.1371/journal.pone.0194889
23. Mandot, P. (2017, August 18). *What is LightGBM, How to implement it? How to fine tune the parameters?* Medium. https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc
24. Meira de Oliveira, E., & Cyrino Oliveira, F. L. (2018). Forecasting mid-long term electric energy consumption through bagging ARIMA and exponential smoothing methods. *Energy*, *144*, 776–788. https://doi.org/10.1016/j.energy.2017.12.049
25. Miraki, A., Parviainen, P., & Arghandeh, R. (2024). Electricity demand forecasting at distribution and household levels using explainable causal graph neural network. *Energy and AI, 16*, 100368. https://doi.org/10.1016/j.egyai.2024.100368
26. Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., & Kavukcuoglu, K. (2016). WaveNet: A Generative Model for Raw Audio. *arXiv*. https://arxiv.org/abs/1609.03499
27. Parmezan, A. R. S., Souza, V. M. A., & Batista, G. E. A. P. A. (2019). Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. *Information Sciences, 484*, 302–337. https://doi.org/10.1016/j.ins.2019.01.076
28. Petneházi, G. (2019). Recurrent Neural Networks for Time Series Forecasting. *arXiv*. https://arxiv.org/abs/1901.00069
29. Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting, 36*(3), 1181–1191. https://doi.org/10.1016/j.ijforecast.2019.07.001
30. Scikit-learn. (n.d.). *Feature importances with a foreset of trees*. Retrieved November 30, 2024, from https://scikit-learn.org/1.5/auto_examples/ensemble/plot_forest_importances.html
31. Seabold, Skipper, & Perktold, J. (2010). *Statsmodels: Econometric and statistical modeling with Python*. 9th Python in Science Conference.

32. Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting, 36*(1), 75–85. https://doi.org/10.1016/j.ijforecast.2019.03.017
33. *Time series forecasting*. (n.d.). TensorFlow. https://www.tensorflow.org/tutorials/structured_data/time_series
34. Wang, Y., Wu, H., Dong, J., Qin, G., Zhang, H., Liu, Y., Qiu, Y., Wang, J., & Long, M. (2024). TimeXer: Empowering Transformers for time series forecasting with exogenous variables. *arXiv*. https://arxiv.org/abs/2402.19072
35. Wen, R., Torkkola, K., Narayanaswamy, B., & Madeka, D. (2018). *A multi-horizon quantile recurrent forecaster*. *arXiv*. https://arxiv.org/abs/1711.11053
36. Williams, S., & Short, M. (2020). Electricity demand forecasting for decentralised energy management. *Energy and Built Environment, 1*(2), 178–186. https://doi.org/10.1016/j.enbenv.2020.01.001
37. Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management Science*, *6*(3), 324–342. https://pubsonline.informs.org/doi/10.1287/mnsc.6.3.324