Forecasting Five Decades of Singapore's Energy Generation with Various Time Series Methods

Bryan Chen, ChengCheng Song, John Tan, Ken Nguyen

2025-12-01

Contents

Introduction 2

Exploratory Data Analysis 3

- Time Series and ACF/PACF Plots . . . 3
- Time Series Decomposition . . . 4

Model Fitting & Selection 5

- ARIMA . . . 5
- ETS . . . 7
- Holt-Winters . . . 9
- NNAR . . . 11
- Prophet . . . 13
- Forecast Combination . . . 15

Discussing Model Diagnostics 18

Conclusions and Future Work 18

References 19

R Source Code 20

Time Series and ACF/PACF Plots 20

Time Series Decomposition 20

ARIMA 21

ETS 22

Holt-Winters 23

NNAR 24

Prophet 25

Forecast Combination 26

Introduction

This study analyzes monthly electricity generation (measured in Gigawatt Hours) in Singapore from January 1975 to July 2025, using official data published by SINGSTAT and sourced from the Energy Market Authority. The data set captures five decades of evolution in Singapore's power sector, reflecting changes driven by rapid industrialization, population growth, economic restructuring, and energy policy reforms. The data are updated monthly and follow different industrial classification standards over time; it is according to the Singapore Standard Industrial Classification (SSIC) 2015 for the years 2020 and before, and SSIC 2020 for 2021 onward, to ensure continuity while aligning with updated economic structures. Given the long time span and the presence of trend, seasonality, and potential structural changes, this data set is well suited for evaluating and comparing multiple time series forecasting models for short- and medium-term electricity generation planning.

Exploratory Data Analysis

Time Series and ACF/PACF Plots

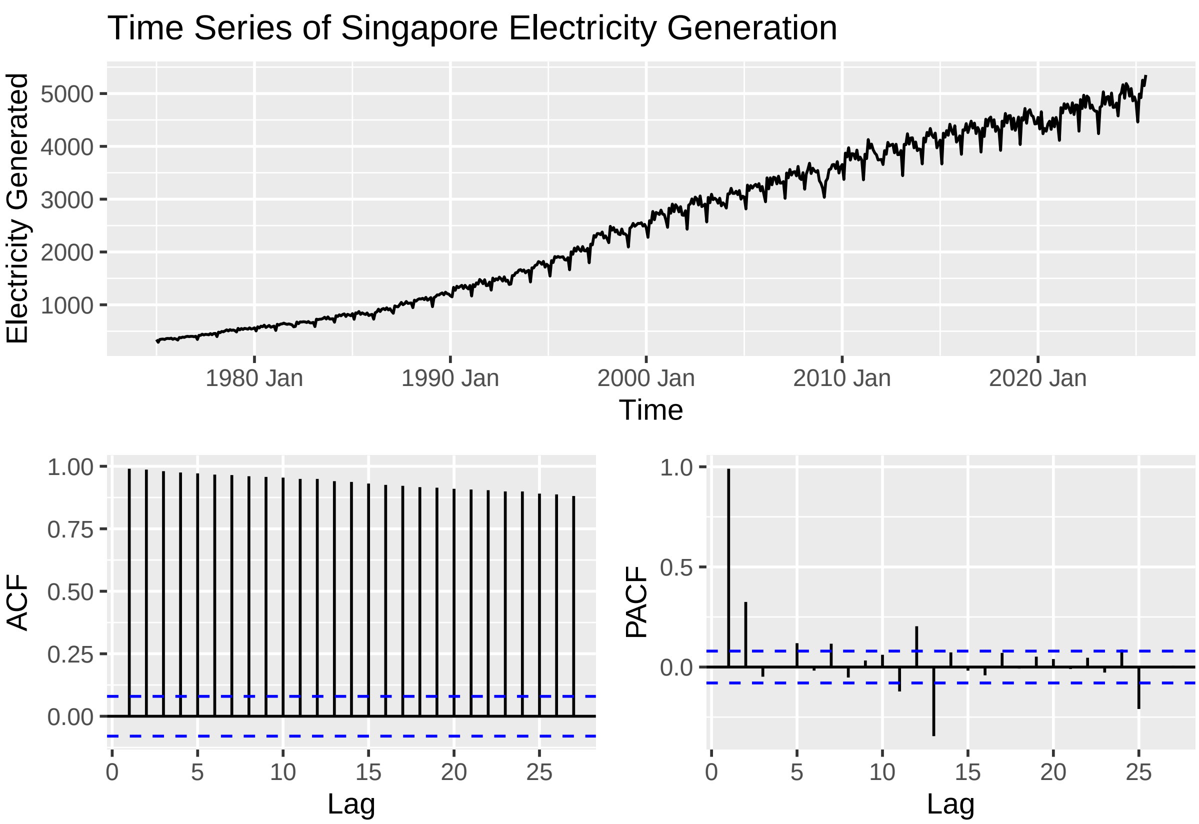

Figure 1: Time Series, ACF and PACF for SG Electricity Generation

The time series of the monthly electricity generation series (*Figure 1*) reveals a steady upward trend, indicating a consistent increase in electricity production over the past five decades. The ACF decays very slowly, suggesting strong persistence and non-stationarity, which is characteristic of a series with a deterministic or stochastic trend. The PACF exhibits significant spikes at lags 1, 2, 12, 13, 25, and so on, indicating the presence of both short-term autocorrelations (lags 1 and 2) and seasonal patterns with a period of 12 months, as well as potential higher-order seasonal interactions. Overall, these patterns suggest that any effective forecasting model should account for trend, seasonality, and autocorrelation structure, potentially requiring differencing and seasonal components to achieve stationarity and accurate predictions.

Time Series Decomposition

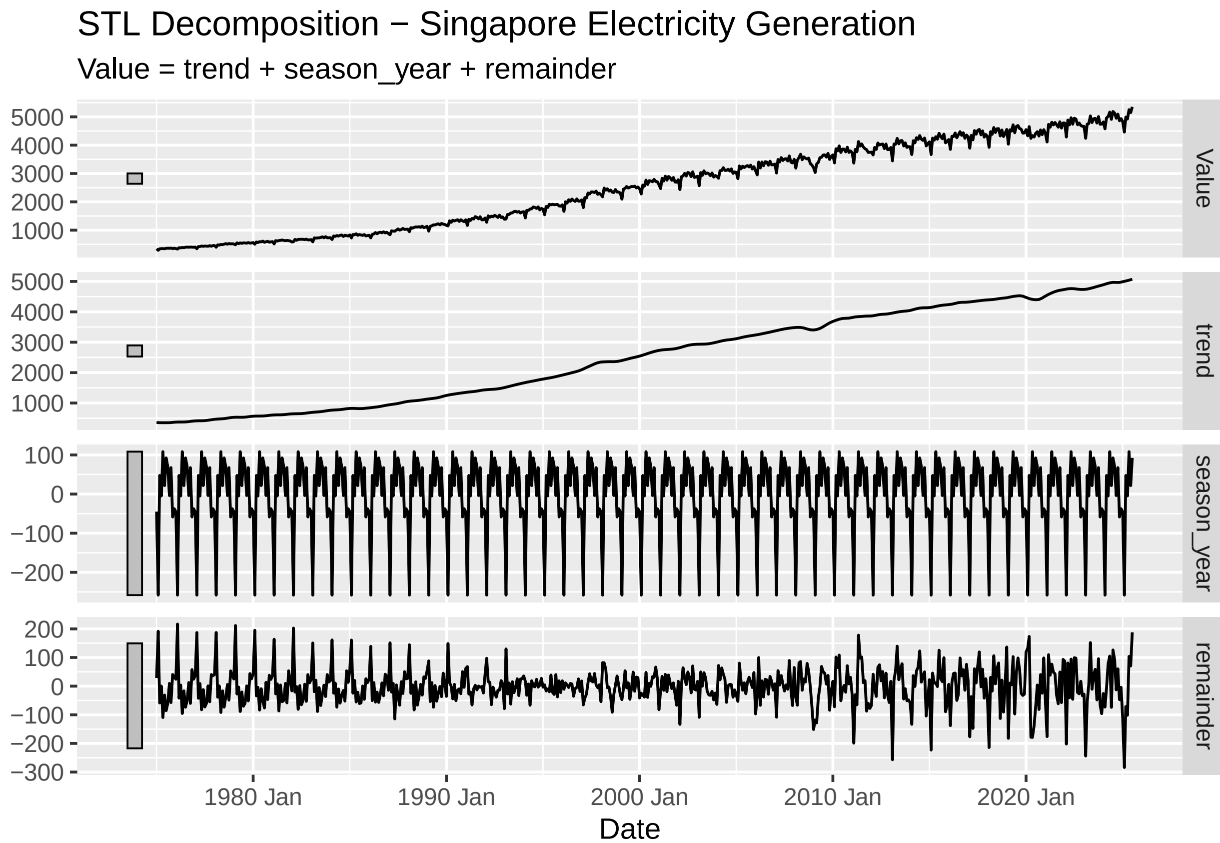

Figure 2: STL Decomposition of Electricity Generation Time Series

Observing the STL decomposition in *Figure 2*, we see a long-run upward trend in Singapore's electricity generation, as well as a clear and stable yearly seasonal pattern. Certain months repeatedly show higher electricity generation, while others consistently show lower levels. The amplitude of the seasonality appears roughly constant, suggesting that seasonal effects have been persistent over time. The remainder component shows an initial decrease in amplitude until around 2000 where the amplitude begins increasing. This suggests there is some cycles present in addition to the white noise. Overall, the STL decomposition confirms that the data contains substantial trend and seasonality, with some evolving cyclical behavior in the remainder component.

Model Fitting & Selection

ARIMA

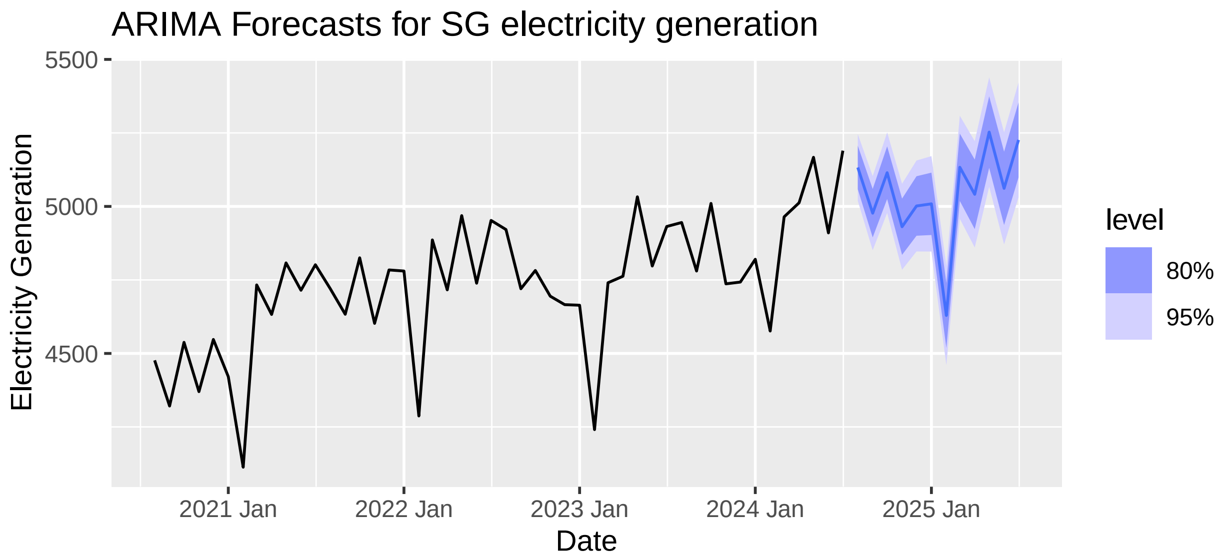

Figure 3: Forecasting 12-steps ahead with ARIMA model

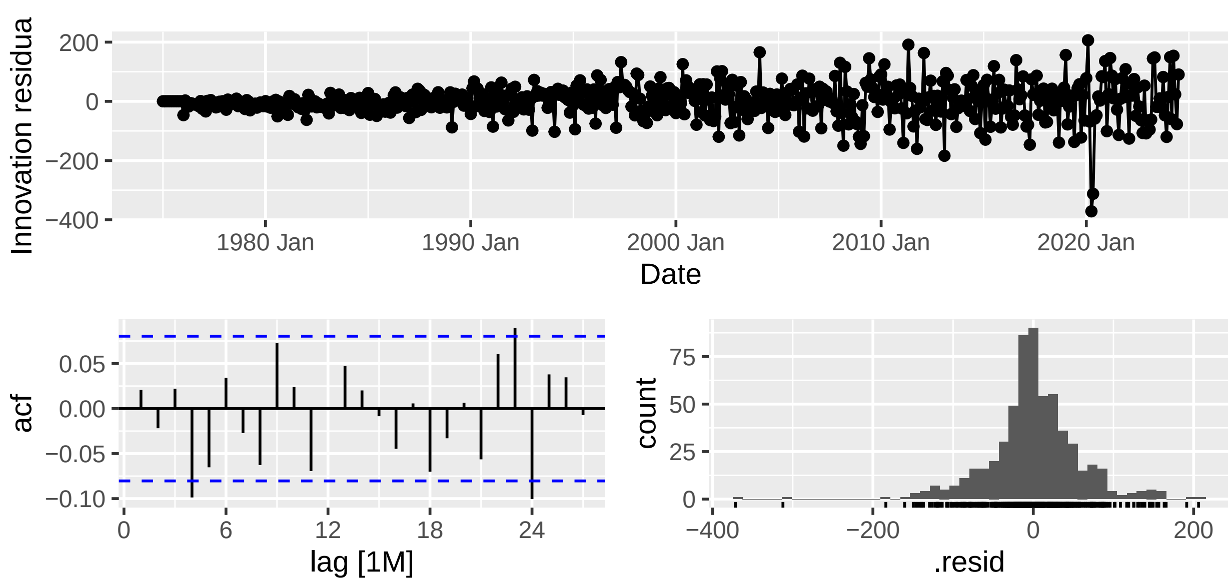

Figure 4: Residual Diagnostics for ARIMA model

Table 1: ARIMA(1,0,1)(0,1,2) with drift – Coefficient Estimates

Term	Estimate	Std. Error	p-value
ar1	0.9709	0.0120	0.0000
ma1	-0.4857	0.0439	0.0000
sma1	-0.8748	0.0491	0.0000
sma2	0.1309	0.0455	0.0042
constant	2.7354	0.3127	0.0000

Table 2: Ljung-Box Test for ARIMA Residuals

Test	X-squared	df	p-value
Ljung-Box	41.352	24	0.0152

The `ARIMA` function produces a ARIMA(1,0,1)(0,1,2) model with drift. The model applies a seasonal differencing as well as a S-MA(2) to deal with the annual seasonality in the time series. It also uses an ARMA(1,1) to model the cycles present. Observing the residuals of the model in *Figure 3*, there appears to be heteroskedasticity present, with the residuals fanning out in a cone-shape over time. Observing the ACF of the residuals, it appears that most of the dynamics in the time series has been modeled, with a few significant spikes remaining at lags 4, 23, and 24. In addition, there appears to be a slight sinusoidal wave pattern in the ACF. This indicates some cycles or seasonality remaining that could potentially be better modeled.

Applying the Ljung-Box test produces a p-value of 0.01523 (seen in *Table 2*), which is significant at the 5% level ($p < 0.05$). Thus, we reject the null of no serial correlation, implying that the residuals do not look like white noise and there are dynamics remaining that could be modeled. Together, the visual pattern and statistical test indicate that the ARIMA model does not fully capture the underlying dynamics, and forecasts may therefore be systematically biased or inefficient.

ETS

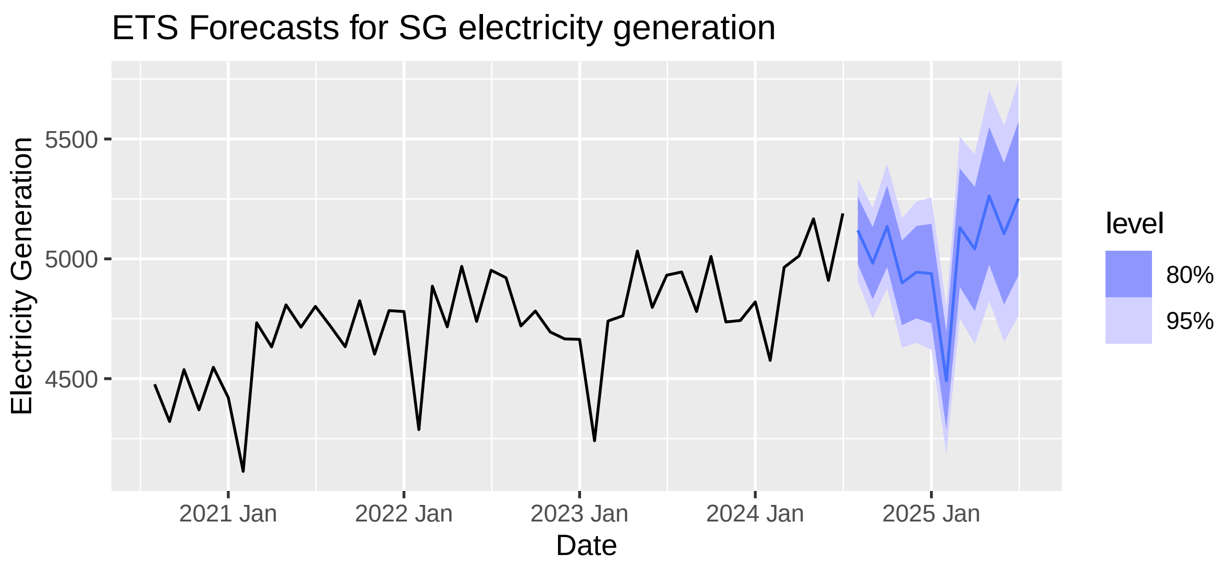

Figure 5: Forecasting 12-steps ahead with ETS model

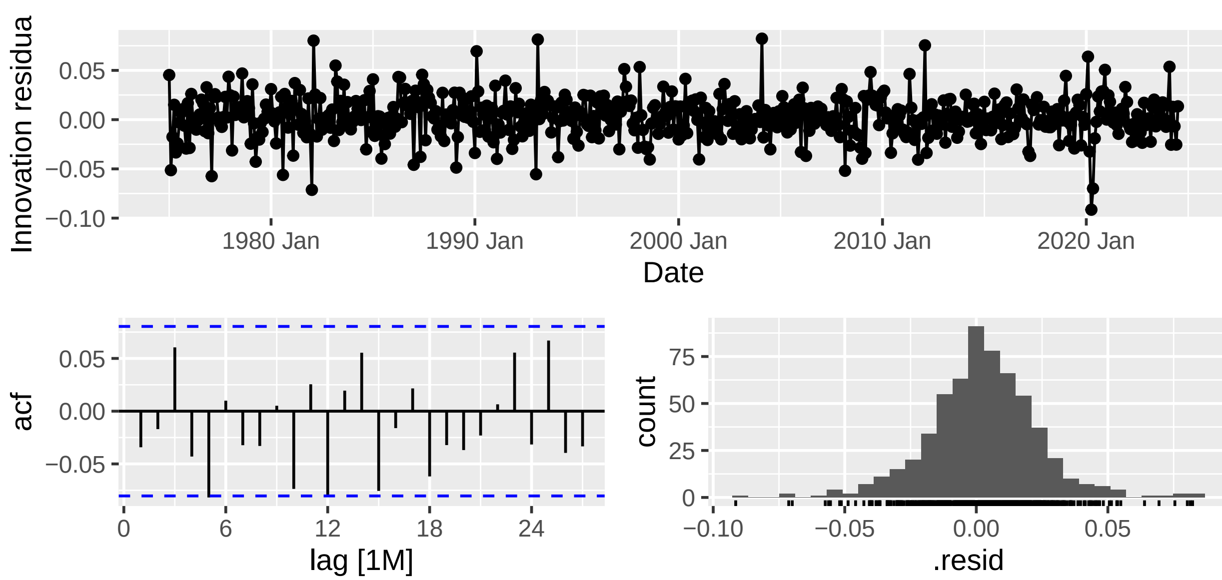

Figure 6: Residual Diagnostics for ETS model

Table 3: ETS(M,Ad,M) – Smoothing Parameters

Parameter	Estimate
alpha	0.434764
beta	0.029441
gamma	0.026436
phi	0.980000

Table 4: Ljung-Box Test for ETS Residuals

Test	X-squared	df	p-value
Ljung-Box	29.956	24	0.1862

The selected ETS(M, Ad, M) model indicates that electricity generation is best explained by multiplicative errors, an additive damped trend, and multiplicative seasonality, implying that both variability and seasonal effects scale with the level of the series. The smoothing parameters suggest a moderate response to new information in the level (alpha = 0.43), a very stable long-run growth rate (beta = 0.03), and slowly evolving seasonality (gamma = 0.03). The initial level (`l[0]` = 327) and positive initial trend (`b[0]` = 4.91) confirm an upward trajectory, while the seasonal indices show systematic monthly variation around the mean.

The residuals diagnostics in *Figure 6* show no significant autocorrelation spikes, no remaining seasonality, and no systematic patterns in the residuals, indicating that the ETS model has successfully captured the underlying trend and seasonal structure of the data. Since the Ljung-Box test in *Table 4* is not statistically significant ($p > 0.05$), we fail to reject the null hypothesis of no autocorrelation, confirming that the residuals behave as white noise and the ETS model is adequate.

Holt-Winters

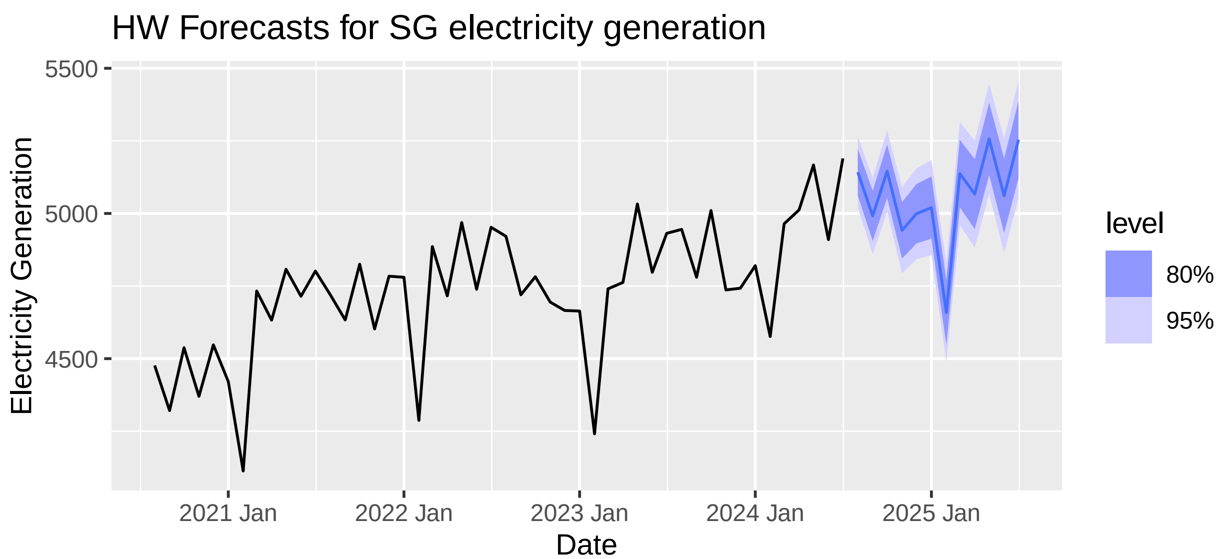

Figure 7: Forecasting 12-steps ahead with Holt-Winters model

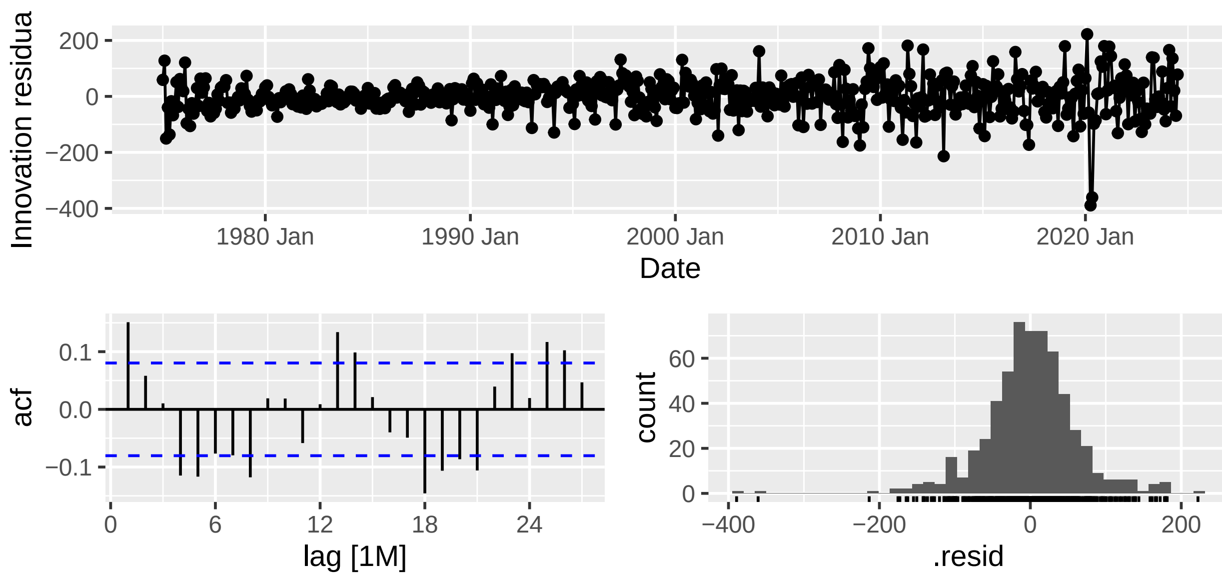

Figure 8: Residual Diagnostics for Holt-Winters model

Table 5: ETS(M,Ad,M) – Smoothing Parameters

Parameter	Estimate
alpha	0.395719
beta	0.000100
gamma	0.225460

Table 6: Ljung-Box Test for ETS Residuals

Test	X-squared	df	p-value
Ljung-Box	108.55	24	0

The fitted Holt–Winters model is an additive error, additive trend, and additive seasonality model, indicating that both the trend and seasonal effects change in absolute (not proportional) terms over time. The smoothing parameter alpha = 0.396 implies moderate responsiveness to new observations, while the extremely small beta = 0 suggests the trend is essentially stable. The seasonal smoothing gamma = 0.225 indicates gradual updating of seasonal patterns. The estimated innovation variance ($\sigma^2 = 3914.016$) is relatively large, reflecting high volatility in the series.

The residual diagnostics in *Figure 8* reveal a clear sinusoidal wave pattern, indicating unexplained cyclical structure remains in the data, which violates the assumption of independent residuals. This is formally confirmed by the Ljung-Box test in *Table 6* ($p < 0.001$), which shows strong residual autocorrelation. Together, the visual pattern and statistical test indicate that the Holt–Winters model does not fully capture the underlying dynamics, and forecasts may therefore be systematically biased or inefficient.

NNAR

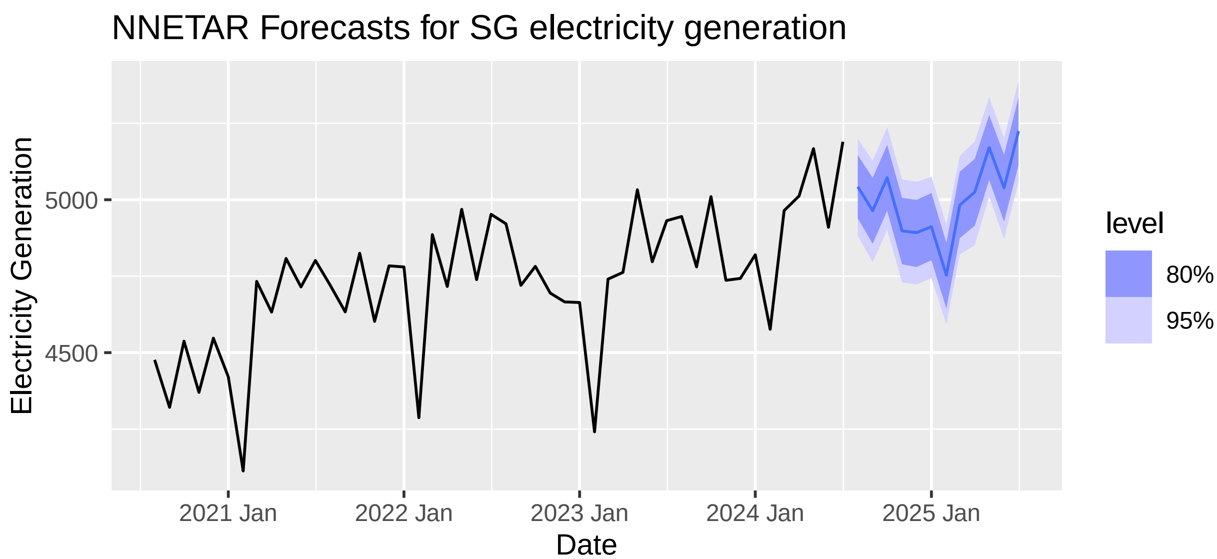

Figure 9: Forecasting 12-steps ahead with NNAR model

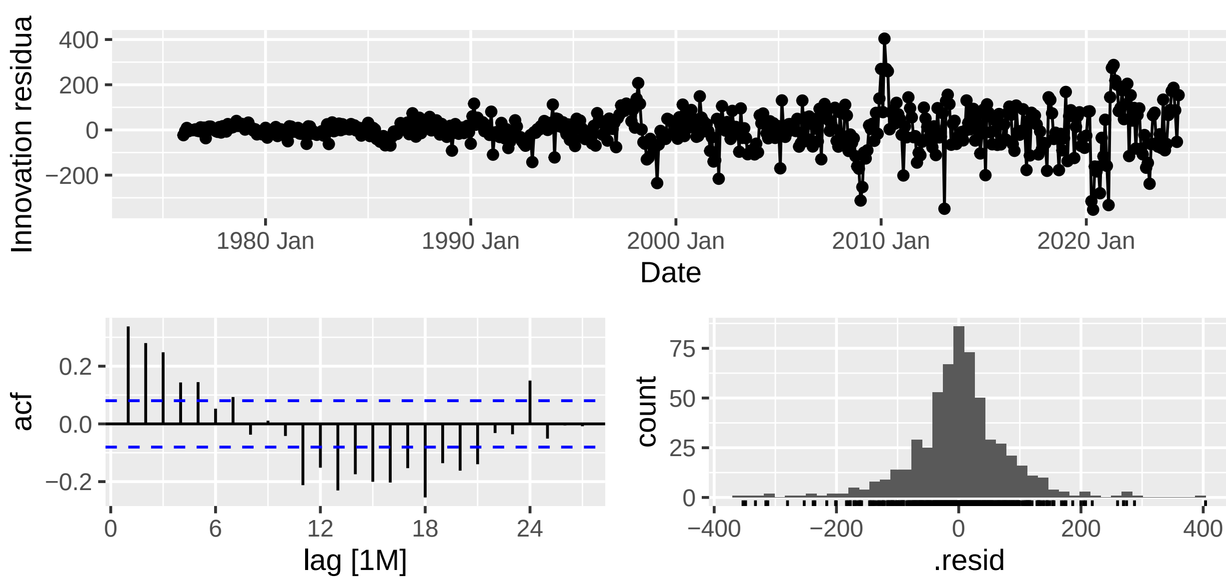

Figure 10: Residual Diagnostics for NNAR model

Table 7: NNAR(2,1,2) Model Summary

Description	Value
Model	NNAR(2,1,2)
Networks averaged	20
Network structure	3-2-1
Number of weights	11
Output activation	Linear
Variance estimate	6751

Table 8: Ljung-Box Test for NNAR Residuals

Test	X-squared	df	p-value
Ljung-Box	427.25	24	$< 2.2\text{e-}16$

The fitted model is a Neural Network Autoregression (NNAR) with two lags in the input layer, one difference, and two neurons in the hidden layer, applied to monthly data with a 12-month seasonal period. It is an average of 20 networks, each a small 3-2-1 network with 11 weights, using linear output units, allowing for continuous forecasting.

The diagnostic plot in *Figure 10* exhibits significant heteroscedasticity with a clear sinusoidal wave in the residuals, indicating that the NNAR model might not fully capture the underlying dynamics. This is confirmed by the Ljung-Box test in *Table 8* ($p < 2.2\text{e-}16$), which indicates strong residual autocorrelation. Nevertheless, it is important to note that NNAR models are nonlinear and can capture complex relationships that classical models cannot, and as a result, its residuals may show autocorrelation, heteroscedasticity, or non-normality even if the model forecasts well. NNAR is often optimized for point forecasts, not for producing residuals that behave like white noise.

Prophet

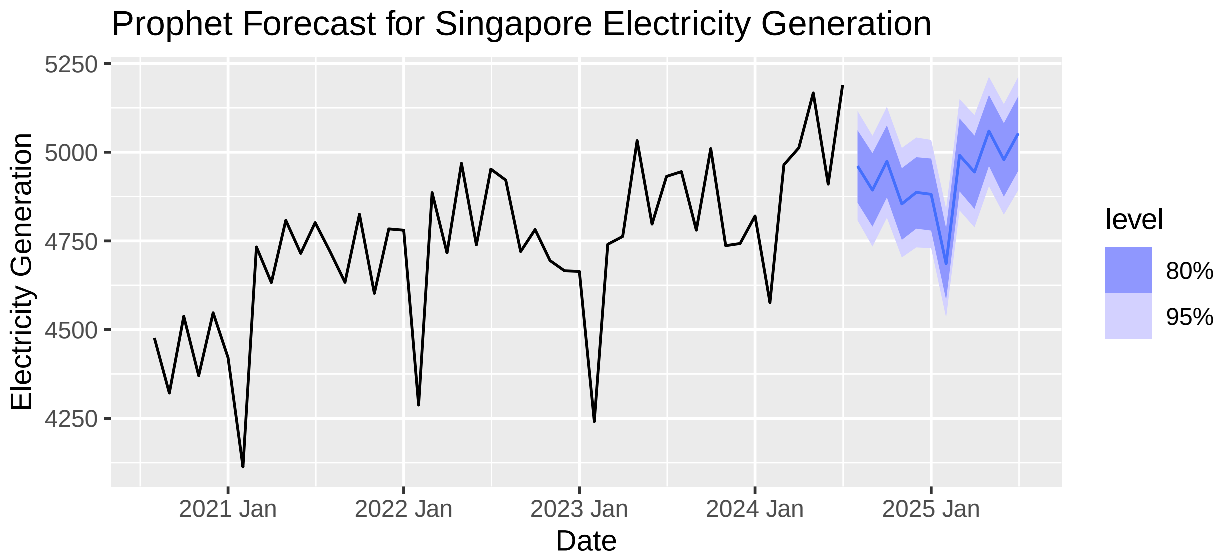

Figure 11: Forecasting 12-steps ahead with Prophet model

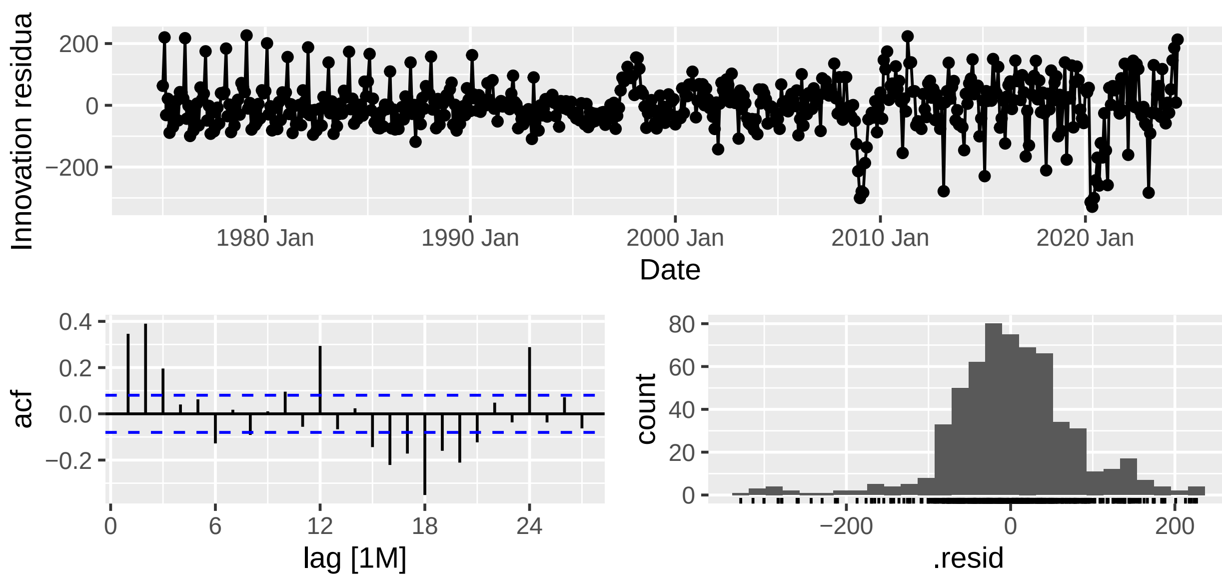

Figure 12: Residual Diagnostics for Prophet model

Table 9: Ljung-Box Test for Prophet Residuals

Test	X-squared	df	p-value
Ljung-Box	510.98	24	< 2.2e-16

Looking at the residual plot of the prophet model in *Figure 12*, it does not seem to be scattered randomly around zero. The ACF plot of the residuals also show several significant spikes in a sinusoidal wave pattern, indicating possible unexplained cyclical dynamics that remain in the data. This is formally confirmed by the Ljung-Box Test in *Table 9* ($p < 2.2e-16$), which shows strong residual autocorrelation. Together, the visual pattern and statistical test indicate that the Prophet model does not fully capture the underlying dynamics and forecasts may therefore be systematically biased or inefficient.

However, similar to the NNAR model, it is worth noting that these models are typically mathematical/machine learning focused, and may not follow the conventional statistical assumptions, despite having superior performance over statistical focused methods. Thus, it may do poorly in the classical tests that other models like ARIMA and exponential smoothing are typically tested against despite stronger performance.

Forecast Combination

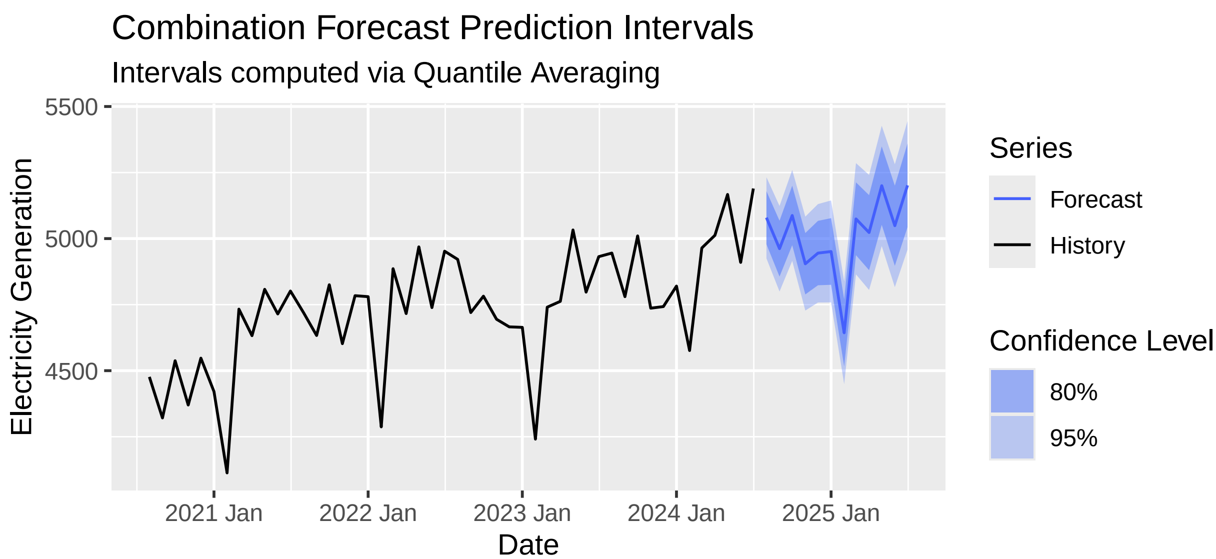

Figure 13: Forecasting 12-steps ahead with Combination model

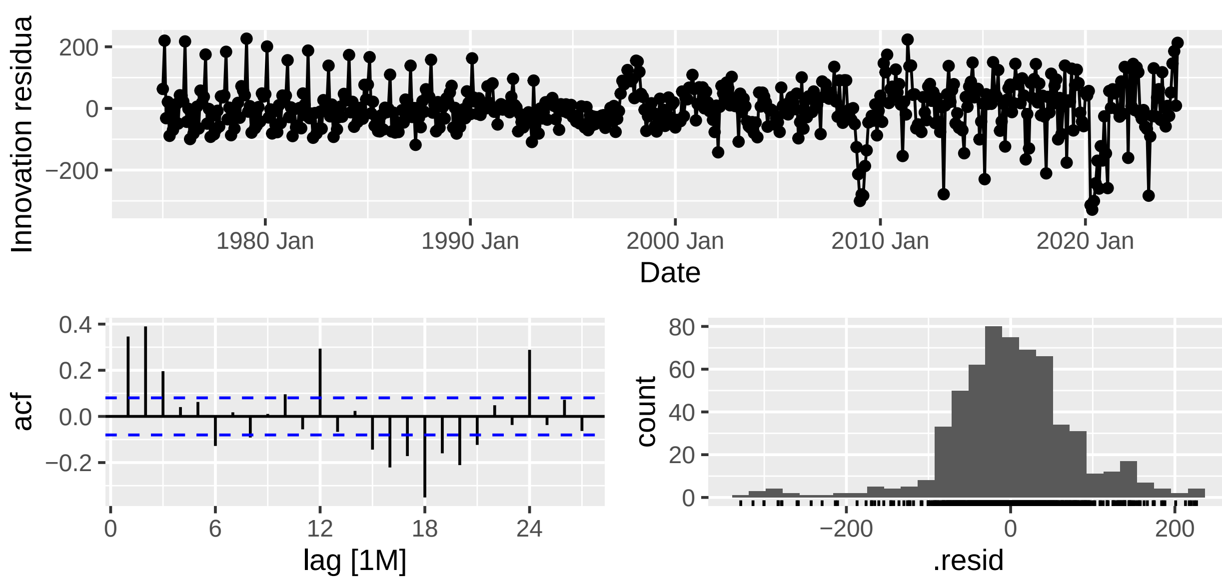

Figure 14: Residual Diagnostics for Combination model

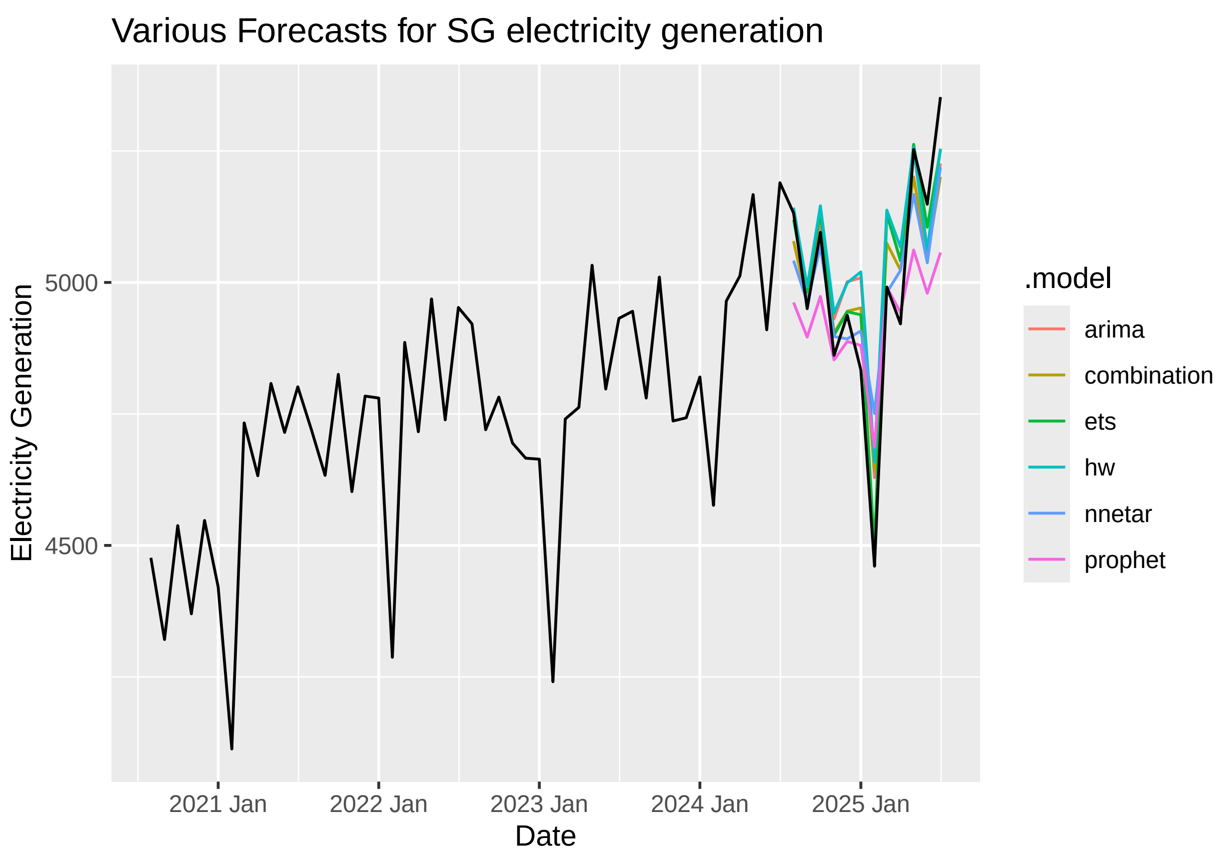

Figure 15: Projected forecasts of all the models

Table 10: Ljung-Box Test for Combination Residuals

Test	X-squared	df	p-value
Ljung-Box	1074.6	24	$< 2.2e\text{-}16$

The forecast combination was constructed with the average of predictions from ARIMA, ETS, Holt-Winters, NNAR, and Prophet models, along with their 80% and 95% prediction intervals. As shown in *Figure 14*, the combined approach produces a smoother series that lies between volatile individual models forecasts. While the combined model improves point accuracy upon previous forecasts (RMSE = 93.67), the residuals diagnostics indicate strong AR tendencies, as reflected by the sinusoidal pattern in the ACF and significant lags. In addition, the low Ljung-box p-value in *Table 10* also suggest serial correlation remaining in the residuals. This suggests that the combination behaves as a balanced compromise but does not fully resolve the cyclical structure in the data.

Discussing Model Diagnostics

Table 11: Model Performance Diagnostics

Model	AIC	AICc	BIC	RMSE	MAPE
ARIMA	6403.44	6403.59	6429.65	102.956	1.7
ETS	**8231.266**	**8232.453**	**8310.26**	**71.959**	**1.134**
Holt-Winters	8741.005	8742.066	8815.61	111.376	1.896
NNAR	-	-	-	111.433	1.715
Prophet	-	-	-	147.118	2.269
Combination	-	-	-	93.669	1.541

Unfortunately, because the NNAR and Prophet models do not use typical statistical methods, we are unable to generate AIC, AICc, and BIC values for these, and thus have to compare the model performance using the RMSE and MAPE values. As indicated in *Table 11*, the ETS model provides the strongest statistical accuracy, achieving both the lowest RMSE (71.96) and MAPE (1.134). ARIMA performs moderately well but retains residual autocorrelation, while Holt-Winters and NNAR both show higher error levels and persistent patterns in their residuals. Prophet has the weakest performance, with the highest RMSE and serial correlation present.

While the combination model improves RMSE relative to most of the models, its high Ljung-Box statistic indicates that it fails to capture all the underlying dynamics. However, we do see overall strong performance, producing the second lowest RMSE and MAPE score. This highlights the robust performance of a combination model which averages the respective biases that each model imposes on the time series forecast.

Conclusions and Future Work

Over the fifty years of electric power production in Singapore, all models clearly establish the strong positive trend and the steady repetition of the annual cycle. With respect to the techniques compared in this report, the ETS model produces the best forecasts since it more accurately detects the processes in the series. Additionally, the residuals - the remainders of the processes to be estimated - fulfill the characteristics of the white noise. Despite the capabilities of the machine-learning techniques NNAR and Prophet to simulate nonlinear processes to a degree of accuracy, the unexplained cycles in both techniques remain significant. These machine-learning methods may be more suitable to time series that exhibit more complex seasonality and higher frequency, allowing it to parse out information from extremely noisy data.

Future improvements may implement additional predictors derived from external factors, for example, industrial production, population trends, or temperature patterns. Other techniques of combining the two models may also prove to be more efficient in terms of reducing the autocorrelation of the residuals to a greater extent compared to the simple averaging approach. Using the forecasts as coefficients within a combination model, or giving different weights to each model forecast can also potentially produce stronger results. Given the length of the sample covering a total of 50 years, structural breaks or global effects may be identified to further refine the accuracy of the forecast in circumstances of policy or technological shifts.

References

Singapore Department of Statistics. (2024). Electricity Generation, Monthly (2025) [Dataset]. data.gov.sg. Retrieved December 5, 2025 from https://data.gov.sg/datasets/d_ae4afbaf5bc96bde19d8ce85810ab9f4/view

R Source Code

Time Series and ACF/PACF Plots

```
p <- autoplot(sg_elec_ts) +

  labs(title = "Time Series of Singapore Electricity Generation",

       x = "Time",

       y = "Electricity Generated")


p1 <- ggAcf(sg_elec_ts$Value) + ggtitle(NULL)

p2 <- ggPacf(sg_elec_ts$Value) + ggtitle(NULL)


combined_acf_pacf <- (p1 + p2) +

  plot_annotation(title = "ACF and PACF for Electricity Generation")


p / combined_acf_pacf
```

Time Series Decomposition

```
sg_electricity_decomp <- sg_elec_ts |>

  model(STL(Value ~ season(window = "periodic"))) |>

  components() |>

  autoplot() +

  labs(title = "STL Decomposition - Singapore Electricity Generation")


sg_electricity_decomp
```

ARIMA

```r
# Create train/test split (last year of data as test)
sg_elec_ts_train <- sg_elec_ts[1:595,]
sg_elec_ts_test <- sg_elec_ts[596:607,]

# Create a smaller window to see closer forecast
sg_elec_window <- filter(sg_elec_ts_train, Date > yearmonth("2020 Jul"))
sg_elec_window_full <- filter(sg_elec_ts, Date > yearmonth("2020 Jul"))
```

```r
# Fit ARIMA model
arima_fit <- sg_elec_ts_train |>
  model(arima = ARIMA(Value))

# Model summary
report(arima_fit)

# Model forecasts
fc_arima <- arima_fit |>
  forecast(new_data = sg_elec_ts_test)

autoplot(fc_arima, sg_elec_window) +
  labs(title = "ARIMA Forecasts for SG electricity generation",
       y = "Electricity Generation")

# Residual diagnostics
gg_tsresiduals(arima_fit)
```

```r
# Ljung-Box test
res_arima <- residuals(arima_fit)
Box.test(res_arima$.resid, lag = 24, type = "Ljung-Box")

# Check Forecast accuracy
accuracy(fc_arima$.mean, sg_elec_ts_test$Value)
```

ETS

```r
# Fit ETS model
fit_ets <- sg_elec_ts_train |>
  model(ETS(Value))

# Model forecasts
fc_ets <- fit_ets |>
  forecast(new_data = sg_elec_ts_test)

autoplot(fc_ets) + autolayer(sg_elec_window, Value) +
      labs(title = "ETS Forecasts for SG electricity generation",
       y = "Electricity Generation")

# Model summary
report(fit_ets)

# Residual diagnostics
gg_tsresiduals(fit_ets)
```

```r
# Ljung-Box test
res_ets <- residuals(fit_ets)
Box.test(res_ets$.resid, lag = 24, type = "Ljung-Box")

# Check accuracy
accuracy(fc_ets$.mean, sg_elec_ts_test$Value)
```

Holt-Winters

```r
# Fit Holt-Winters model
fit_hw <- sg_elec_ts_train |>
  model(HW = ETS(Value ~ error("A") + trend("A") + season("A")))

# Model forecasts
fc_hw <- fit_hw |> forecast(new_data = sg_elec_ts_test)

autoplot(fc_hw) + autolayer(sg_elec_window, Value) +
    labs(title = "HW Forecasts for SG electricity generation",
       y = "Electricity Generation")

# Model summary
report(fit_hw)

# Residual diagnostics
gg_tsresiduals(fit_hw)

# Ljung-Box test
```

```r
res_hw <- residuals(fit_hw)

Box.test(res_hw$.resid, lag = 24, type = "Ljung-Box")


# Check accuracy

accuracy(fc_hw$.mean, sg_elec_ts_test$Value)
```

NNAR

```r
# fit NNAR model

nnetar_fit <- sg_elec_ts_train |>

  model(NNETAR(Value))


# Model forecasts

fc_nnetar <- nnetar_fit |>

  forecast(new_data = sg_elec_ts_test)


autoplot(fc_nnetar, sg_elec_window) +

  labs(title = "NNETAR Forecasts for SG electricity generation",

       y = "Electricity Generation")


# Model summary

report(nnetar_fit)


# Residual diagnostics

gg_tsresiduals(nnetar_fit)


# Ljung-Box test
```

```r
res_nnetar <- residuals(nnetar_fit)
Box.test(res_nnetar$.resid, lag = 24, type = "Ljung-Box")


# Check Forecast accuracy
accuracy(fc_nnetar$.mean, sg_elec_ts_test$Value)
```

Prophet

```r
# fit prophet
prophet_fit <- sg_elec_ts_train |>
  model(
    prophet(Value ~ season(period = "year", order = 6)))


# forecast
fc_prophet <- prophet_fit |>
  forecast(new_data = sg_elec_ts_test)


fc_prophet |>
  autoplot(sg_elec_window) +
  labs(
    x = "Date",
    y = "Electricity Generation",
    title = "Prophet Forecast for Singapore Electricity Generation"
  )


# Prophet decomposition
prophet_decomp <- prophet_fit |>
```

```r
  components()

prophet_decomp |>
  autoplot() +
  labs(title = "Prophet Model Decomposition")

# residual diagnostics
gg_tsresiduals(prophet_fit)

# Ljung-Box Test
res_prophet <- residuals(prophet_fit)
Box.test(res_prophet$.resid, lag = 24, type = "Ljung-Box")

# Check accuracy
accuracy(fc_prophet$.mean, sg_elec_ts_test$Value)
```

Forecast Combination

```r
# fit combination
combo_fit <- sg_elec_ts_train |>
  model(
    arima = ARIMA(Value),
    ets   = ETS(Value),
    nnetar  = NNETAR(Value),
    prophet = prophet(Value ~ season(period = "year", order = 6)),
    hw    = ETS(Value ~ error("A") + trend("A") + season("A"))
  ) |>
```

```
  mutate(combination = (arima + ets + nnetar + prophet + hw) / 5)

# forecast
fc_combo <- combo_fit |>
  forecast(new_data = sg_elec_ts_test)

fc_combo |>
  autoplot(level = NULL) + autolayer(sg_elec_window_full, Value) +
  labs(title = "Various Forecasts for SG electricity generation",
       y = "Electricity Generation")

# Residual diagnostics
combo_fit |>
  select(combination) |>
  gg_tsresiduals()

# Ljung-Box test
res_combo <- residuals(combo_fit)
Box.test(res_combo$.resid, lag = 24, type = "Ljung-Box")

# Check Forecast accuracy
fc_combo_values <- fc_combo |> filter(.model == "combination")
accuracy(fc_combo_values$.mean, sg_elec_ts_test$Value)

# 2. Calculate intervals for the COMPONENTS only
# manually average the bounds of the 5 component models.
fc_intervals <- fc_combo |>
```

```r
  filter(.model != "combination") |>  # Drop the complex combination column
  hilo(level = c(80, 95)) |>          # Calculate bounds for ARIMA, Prophet, etc.
  unpack_hilo(c("80%", "95%")) |>     # Extract numeric columns
  index_by(Date) |>                   # Group by time step
  summarise(
    # Calculate the average forecast mean
    mean_forecast = mean(.mean),
    # Average the Lower and Upper bounds across all 5 models
    lo80 = mean(`80%_lower`),
    hi80 = mean(`80%_upper`),
    lo95 = mean(`95%_lower`),
    hi95 = mean(`95%_upper`)
  )

# 3. Plot
fc_intervals |>
  ggplot(aes(x = Date)) +
  # --- Add the Prediction Intervals (Ribbons) ---
  geom_ribbon(aes(ymin = lo95, ymax = hi95, fill = "95%"), alpha = 0.3) +
  geom_ribbon(aes(ymin = lo80, ymax = hi80, fill = "80%"), alpha = 0.5) +
  # --- Add the Forecast Line ---
  geom_line(aes(y = mean_forecast, color = "Forecast")) +
  # --- Add Historical Data ---
  geom_line(data = sg_elec_window, aes(y = Value, color = "History")) +
  # --- Styling ---
  scale_fill_manual(values = c("80%" = "#446ffc", "95%" = "#446ffc"),
                    name = "Confidence Level") +
```

```
scale_color_manual(values = c("Forecast" = "#445ffc", "History" = "black"),
                   name = "Series") +
labs(
  title = "Combination Forecast Prediction Intervals",
  subtitle = "Intervals computed via Quantile Averaging",
  y = "Electricity Generation"
)
```