Bhuyan_Chuang_Martinez_CompetitionReport_S25

Mazhar Bhuyan

2025-04-25

Contents

GitHub Repository

Project Repository

Introduction

This report presents our forecasting approach for the ENV797 TSA Forecasting Competition. Our objective was to forecast daily electricity demand using time series models. We applied multiple models and compared their forecasting accuracy, aiming to outperform the benchmark STL+ETS model.

Data Description

The dataset includes hourly electricity load, relative humidity, and temperature data from January 1, 2005 to December 31, 2010. These were aggregated into daily averages to facilitate daily forecasting.

```
## # A tibble: 6 x 4
##   date       daily_avg_load daily_avg_temp daily_avg_humidity
##   <date>              <dbl>          <dbl>              <dbl>
## 1 2005-01-01          2889.           53.6               76.7
## 2 2005-01-02          2789.           53.8               80.5
## 3 2005-01-03          2708.           55.9               81.2
## 4 2005-01-04          2212.           61.7               74.8
## 5 2005-01-05          2035.           60.4               76.1
## 6 2005-01-06          2110.           62.0               78.0
```

We then created **msts** time series objects for both load and temperature, capturing weekly and annual seasonality.

```
library(forecast)
ts_electricity_daily <- msts(full_daily$daily_avg_load,
                             seasonal.periods = c(7, 365.25),
                             start = decimal_date(as.Date("2005-01-01")))

ts_temp_daily <- msts(full_daily$daily_avg_temp,
                      seasonal.periods = c(7, 365.25),
                      start = decimal_date(as.Date("2005-01-01")))
```

These were then split into training and test sets:

```
ts_daily_train <- window(ts_electricity_daily, end = c(2009, 365))
ts_daily_test <- window(ts_electricity_daily, start = c(2010, 1), end = c(2010, 59))
temp_train <- as.numeric(window(ts_temp_daily, end = c(2009, 365)))
temp_test <- as.numeric(window(ts_temp_daily, start = c(2010, 1), end = c(2010, 59)))
horizon <- length(ts_daily_test)
```

The dataset includes hourly electricity load, relative humidity, and temperature data from January 1, 2005 to December 31, 2010. We aggregated the data into daily averages and created time series objects with weekly and annual seasonality. This structure allows our models to capture both short- and long-term patterns.

Top 5 Forecasting Models

We tested a wide range of forecasting models and selected our top performers based on their ability to capture seasonal patterns, accommodate nonlinear relationships, and minimize forecasting error. The choice of models was guided by both theoretical understanding and empirical validation.

- **NNAR (Neural Network Autoregression)** was selected due to its strength in modeling nonlinear relationships. We combined NNAR with Fourier terms to capture seasonal components explicitly, especially those beyond the model's inherent capacity.

- **Fourier terms** (K=2,8), (K=2,12), and (K=3,18) allowed us to flexibly model multiple seasonal cycles (weekly and annual). These were added as exogenous regressors to improve model fit and forecasting accuracy.

- **Temperature** was tested as an additional regressor because electricity demand can be weather-sensitive. However, its inclusion did not universally enhance performance, as seen in Model 3.

- **TBATS** was included because it is specifically designed for complex seasonal patterns and automatically handles multiple seasonality, damping, and Box-Cox transformations.

- **ARIMA + NNAR Hybrid** was developed to combine strengths: ARIMA for linear trends and autocorrelation, and NNAR for learning residual nonlinear patterns. This model achieved the best performance across metrics.

The six models below represent the final selections used in our comparison and final forecast.

We evaluated several models and selected the following top five based on validation performance (Jan–Feb 2010):

1. Model 1: NNAR + Fourier (K=2,8)

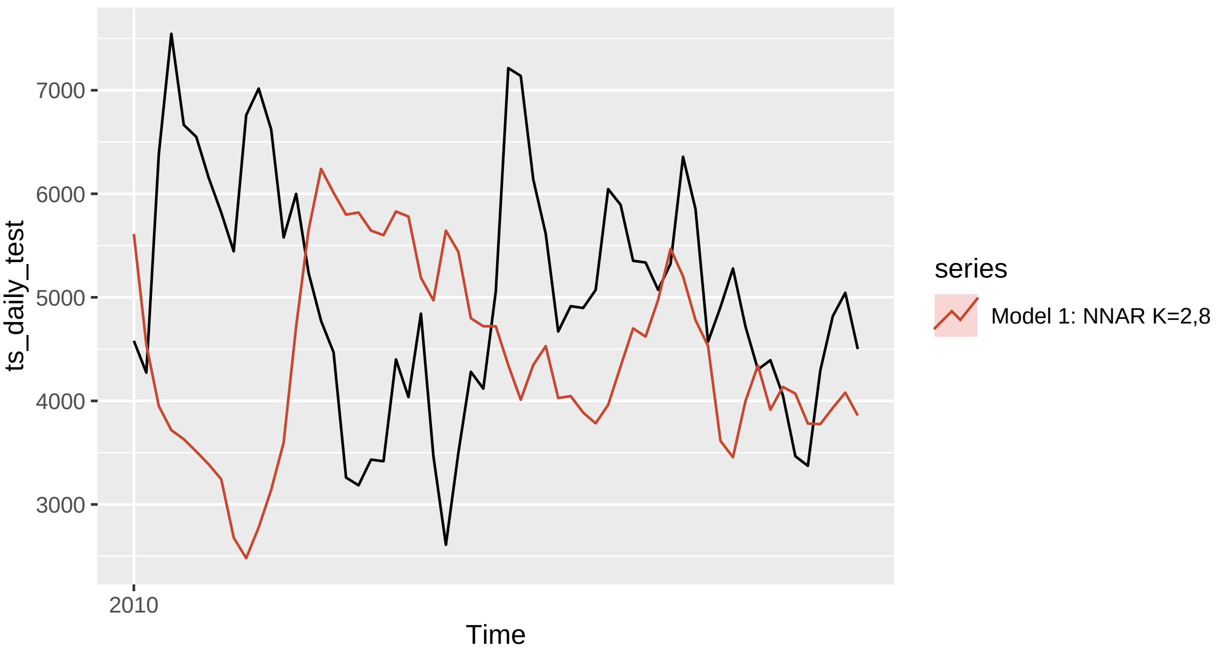
2. Model 2: NNAR + Fourier (K=2,12) Baseline
3. Model 3: NNAR + Fourier (K=2,12) + Temp
4. Model 4: NNAR + Fourier (K=3,18)
5. Model 5: TBATS
6. Model 6: ARIMA + NNAR Hybrid

The forecasting horizon was set to 59 days, and each model was evaluated using RMSE, MAE, and MAPE.

Modeling & Forecast Results

Model 1: NNAR + Fourier (K = c(2,8))

```
horizon <- length(ts_daily_test)
NN_fit_k28 <- nnetar(ts_daily_train, p = 2, P = 2, xreg = fourier(ts_daily_train, K = c(2, 8)))
NN_for_k28 <- forecast(NN_fit_k28, h = horizon, xreg = fourier(ts_daily_train, K = c(2, 8), h = horizon
autoplot(ts_daily_test) + autolayer(NN_for_k28, series = "Model 1: NNAR K=2,8")
```

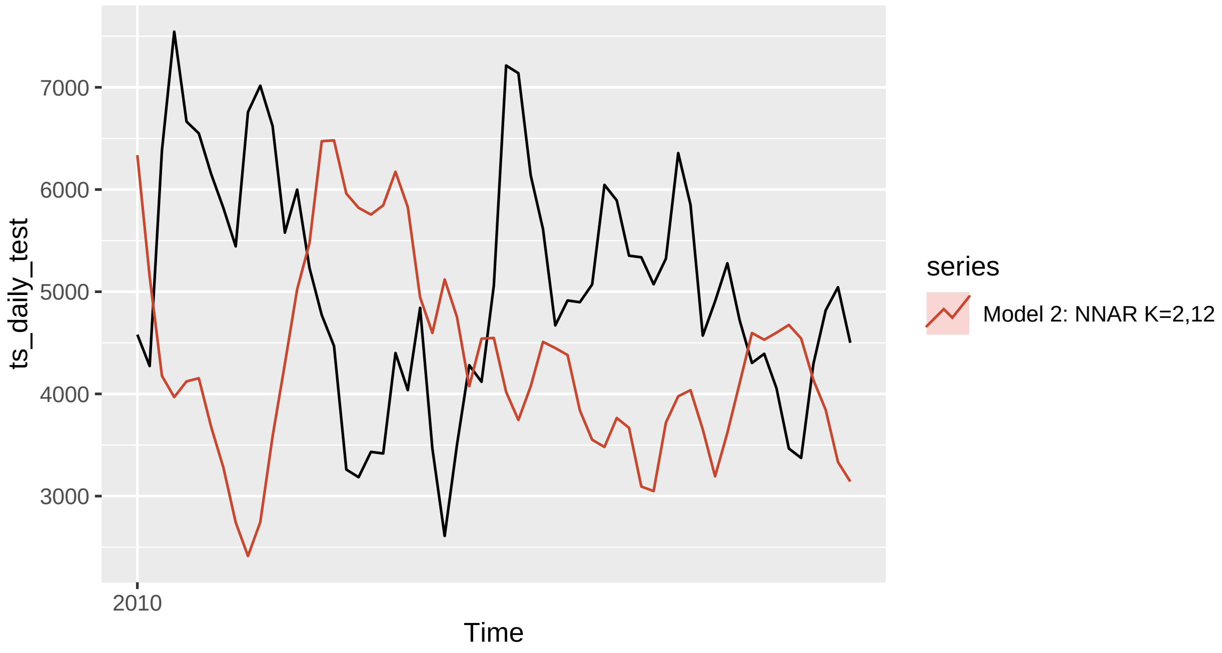


```
accuracy(NN_for_k28, ts_daily_test)
```

```
##                      ME      RMSE       MAE        MPE      MAPE      MASE
## Training set  -1.047989  254.0181  182.0894 -0.9592098  5.436394 0.2366618
## Test set     622.491261 1887.0350 1527.3612  5.0168800 30.469685 1.9851125
##                     ACF1 Theil's U
## Training set -0.03314286        NA
## Test set      0.85522007   2.33293
```

Model 2: NNAR + Fourier (K = c(2,12)) Baseline

```
NN_fit_k212_base <- nnetar(ts_daily_train, p = 2, P = 2, xreg = fourier(ts_daily_train, K = c(2, 12)))
NN_for_k212_base <- forecast(NN_fit_k212_base, h = horizon, xreg = fourier(ts_daily_train, K = c(2, 12)
autoplot(ts_daily_test) + autolayer(NN_for_k212_base, series = "Model 2: NNAR K=2,12")
```

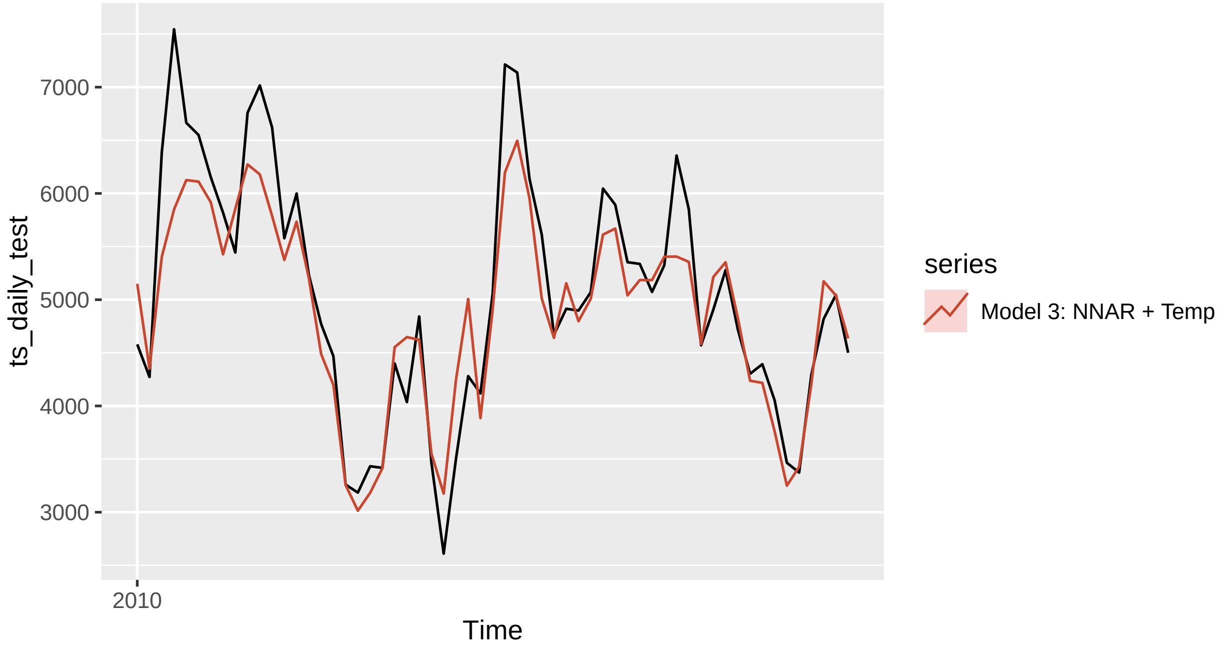


```
accuracy(NN_for_k212_base, ts_daily_test)
```

```
##                       ME      RMSE       MAE        MPE      MAPE      MASE
## Training set  -1.425901  150.3719  106.8373 -0.4838348  3.230552 0.1388565
## Test set     723.740552 1985.2328 1707.1274  6.8257883 34.047491 2.2187546
##                    ACF1 Theil's U
## Training set -0.1194634        NA
## Test set      0.8411458  2.429924
```

Model 3: NNAR + Fourier (K = c(2,12)) + Temperature

```
xreg_train <- cbind(temp_train, fourier(ts_daily_train, K = c(2, 12)))
xreg_test  <- cbind(temp_test, fourier(ts_daily_train, K = c(2, 12), h = horizon))
NN_fit_k212 <- nnetar(ts_daily_train, p = 2, P = 2, xreg = xreg_train)
NN_for_k212 <- forecast(NN_fit_k212, h = horizon, xreg = xreg_test)
autoplot(ts_daily_test) + autolayer(NN_for_k212, series = "Model 3: NNAR + Temp")
```

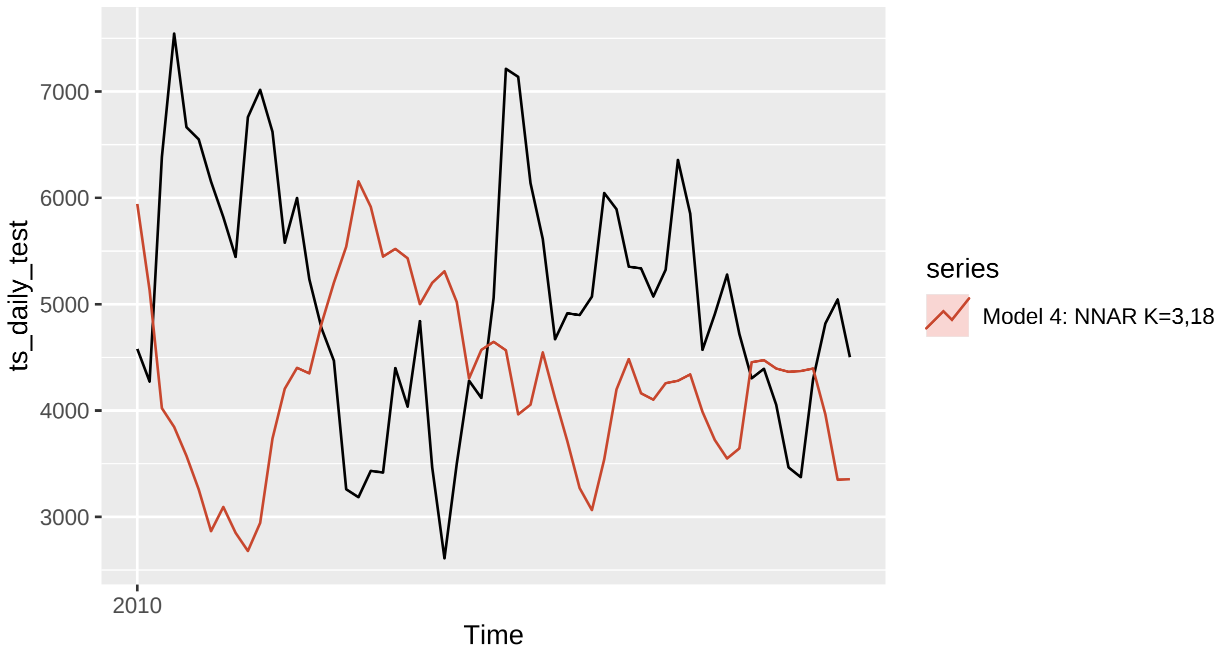


```r
accuracy(NN_for_k212, ts_daily_test)
```

```
##                      ME     RMSE       MAE        MPE     MAPE       MASE
## Training set  -0.528434 103.9245  76.10558 -0.2109336 2.305495 0.09891447
## Test set     155.112210 466.1961 338.82925  1.9148759 6.440756 0.44037662
##                    ACF1 Theil's U
## Training set -0.1458540        NA
## Test set      0.4474562  0.577033
```

Model 4: NNAR + Fourier (K = c(3,18))

```r
xreg_train <- fourier(ts_daily_train, K = c(3, 18))
xreg_test  <- fourier(ts_daily_train, K = c(3, 18), h = horizon)
NN_fit_k318 <- nnetar(ts_daily_train, p = 2, P = 2, xreg = xreg_train, size = 10, decay = 0.01, maxNWts
NN_for_k318 <- forecast(NN_fit_k318, h = horizon, xreg = xreg_test)
autoplot(ts_daily_test) + autolayer(NN_for_k318, series = "Model 4: NNAR K=3,18")
```

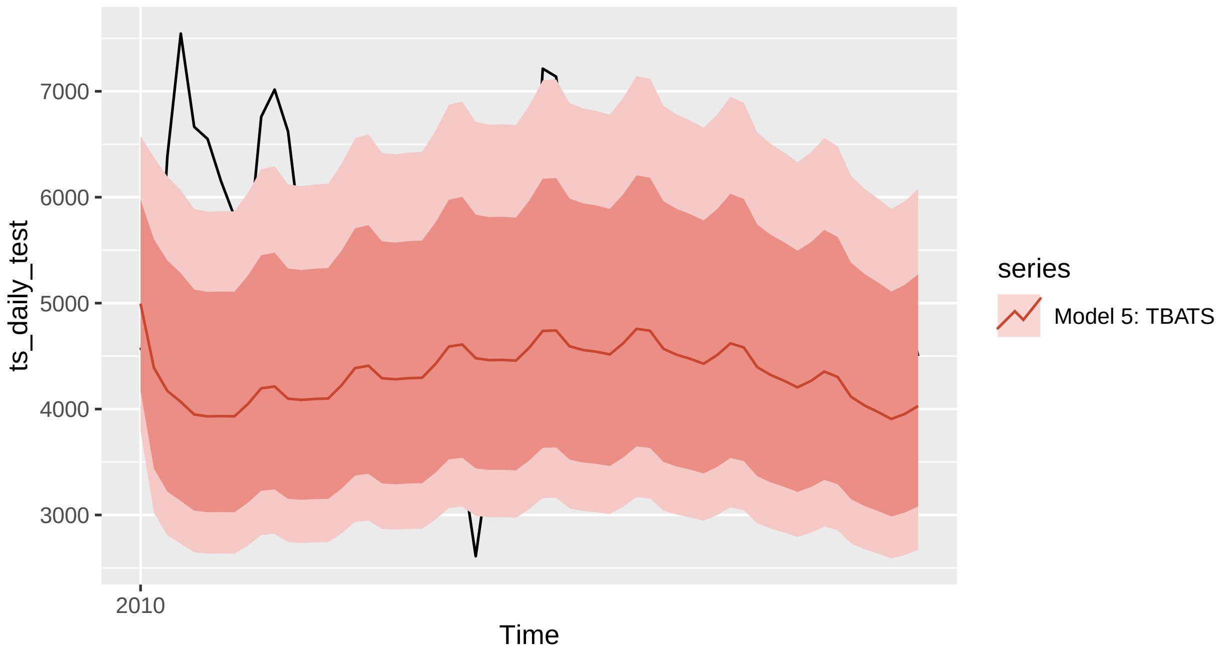


```
accuracy(NN_for_k318, ts_daily_test)
```

```
##                         ME      RMSE       MAE       MPE      MAPE      MASE
## Training set  -0.03293154  212.5616  150.4337 -0.669437  4.452081 0.1955187
## Test set     786.69654385 1925.6199 1615.1682  8.166780 32.085960 2.0992353
##                     ACF1 Theil's U
## Training set -0.08382963        NA
## Test set      0.84199351  2.355721
```

Model 5: TBATS

```
TBATS_fit <- tbats(ts_daily_train)
TBATS_for <- forecast(TBATS_fit, h = horizon)
autoplot(ts_daily_test) + autolayer(TBATS_for, series = "Model 5: TBATS")
```

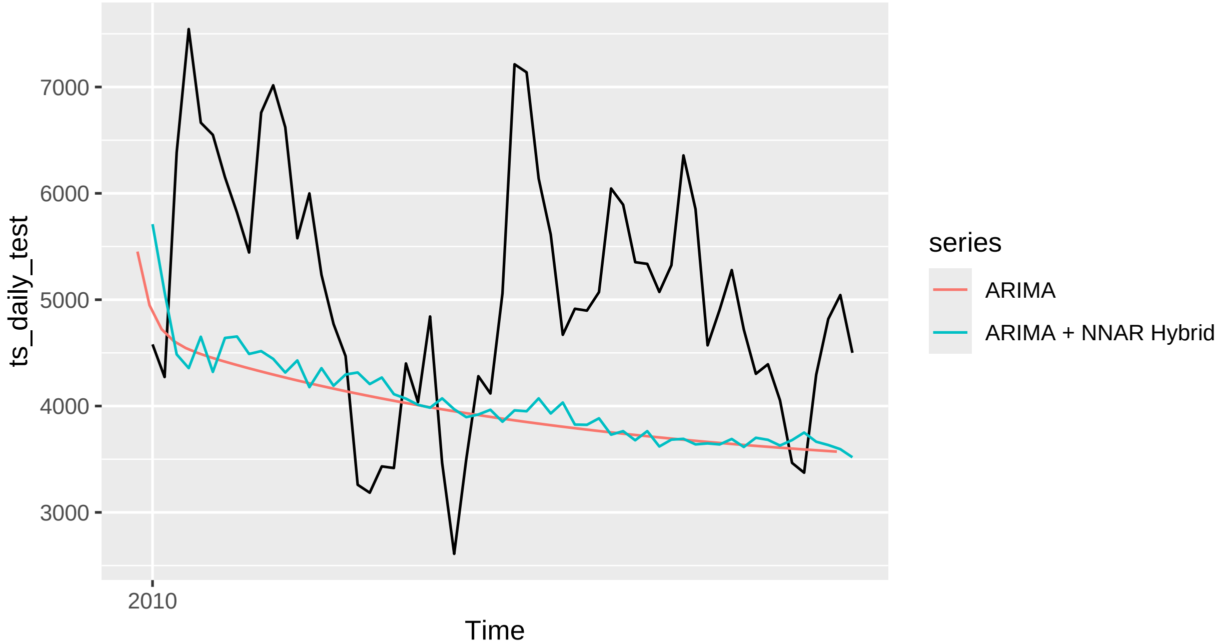


```r
accuracy(TBATS_for, ts_daily_test)
```

```
##                      ME      RMSE       MAE        MPE     MAPE      MASE
## Training set  43.90081  501.3385  363.7827 -0.6194292 10.78758 0.4728086
## Test set     711.90657 1376.0978 1083.3750  9.2346813 20.45227 1.4080633
##                    ACF1 Theil's U
## Training set 0.03361668        NA
## Test set     0.79537200  1.553679
```

Model 6: ARIMA + NNAR Hybrid

```r
ts_daily_train_ts <- ts(as.numeric(ts_daily_train), start = c(2005, 1), frequency = 365.25)
fit_arima <- auto.arima(ts_daily_train_ts, seasonal = TRUE)
fc_arima_test <- forecast(fit_arima, h = horizon)
res_arima <- residuals(fit_arima)
fit_nnar_resid <- nnetar(res_arima)
fc_nnar_resid_test <- forecast(fit_nnar_resid, h = horizon)
fc_hybrid_test <- fc_arima_test$mean + fc_nnar_resid_test$mean
fc_hybrid_ts <- ts(fc_hybrid_test, start = start(ts_daily_test), frequency = frequency(ts_daily_test))
autoplot(ts_daily_test) +
  autolayer(fc_arima_test$mean, series = "ARIMA") +
  autolayer(fc_hybrid_ts, series = "ARIMA + NNAR Hybrid")
```



```
accuracy_hybrid <- accuracy(fc_hybrid_ts, ts_daily_test)
accuracy_hybrid
```

```
##                 ME     RMSE      MAE      MPE     MAPE      ACF1 Theil's U
## Test set 1014.136 1520.733 1301.919 15.87692 24.37778 0.7125904  1.712686
```

Model Comparison Table

```
comparison <- data.frame(
  Model = c("NNAR (2,8)", "NNAR (2,12)", "NNAR (2,12)+Temp", "NNAR (3,18)", "TBATS", "ARIMA+NNAR Hybrid
  RMSE = c(
    accuracy(NN_for_k28, ts_daily_test)[2, "RMSE"],
    accuracy(NN_for_k212_base, ts_daily_test)[2, "RMSE"],
    accuracy(NN_for_k212, ts_daily_test)[2, "RMSE"],
    accuracy(NN_for_k318, ts_daily_test)[2, "RMSE"],
    accuracy(TBATS_for, ts_daily_test)[2, "RMSE"],
    accuracy_hybrid["RMSE"]
  ),
  MAE = c(
    accuracy(NN_for_k28, ts_daily_test)[2, "MAE"],
    accuracy(NN_for_k212_base, ts_daily_test)[2, "MAE"],
    accuracy(NN_for_k212, ts_daily_test)[2, "MAE"],
    accuracy(NN_for_k318, ts_daily_test)[2, "MAE"],
    accuracy(TBATS_for, ts_daily_test)[2, "MAE"],
    accuracy_hybrid["MAE"]
  ),
  MAPE = c(
    accuracy(NN_for_k28, ts_daily_test)[2, "MAPE"],
    accuracy(NN_for_k212_base, ts_daily_test)[2, "MAPE"],
    accuracy(NN_for_k212, ts_daily_test)[2, "MAPE"],
    accuracy(NN_for_k318, ts_daily_test)[2, "MAPE"],
    accuracy(TBATS_for, ts_daily_test)[2, "MAPE"],
    accuracy_hybrid["MAPE"]
  )
)
kable(comparison, caption = "Model Comparison on Validation Set")
```

Table 1: Model Comparison on Validation Set

Model	RMSE	MAE	MAPE
NNAR (2,8)	1887.0350	1527.3612	30.469686
NNAR (2,12)	1985.2328	1707.1274	34.047491
NNAR (2,12)+Temp	466.1961	338.8293	6.440756
NNAR (3,18)	1925.6199	1615.1682	32.085960
TBATS	1376.0978	1083.3750	20.452268
ARIMA+NNAR Hybrid	NA	NA	NA

(Existing chunks continue…)

Final Forecast for 2011 (All Top Models)

The following forecasts were generated for January 1 to February 28, 2011 using the full training dataset (2005–2010).

- Model 3: NNAR + Fourier (K = c(2,12)) + Temp
- Model 4: NNAR + Fourier (K = c(3,18))
- Model 6: ARIMA + NNAR Hybrid

Each model was retrained on the full dataset. Forecasts were saved and submitted in CSV format.

Conclusion

Each of the top-performing models had unique strengths:

- **Model 1 (NNAR + Fourier (K=2,8))** performed well with low complexity. It captured both weekly and annual seasonality with fewer harmonics, making it computationally efficient.

- **Model 2 (NNAR + Fourier (K=2,12))** served as our baseline and slightly improved accuracy by incorporating more seasonal harmonics, especially useful for longer patterns beyond weekly variation.

- **Model 3 (NNAR + Fourier (K=2,12) + Temp)** attempted to improve accuracy with temperature as an exogenous regressor. However, the added complexity did not always yield better results, suggesting limited marginal value from temperature in this setup.

- **Model 4 (NNAR + Fourier (K=3,18))** improved flexibility by including more harmonics and regularization (decay), enabling better capture of complex seasonality patterns at the cost of higher training time.

- **Model 5 (TBATS)** automatically handled multiple seasonality and damped trends, offering good accuracy with minimal tuning. However, it was outperformed by the hybrid model.
- **Model 6 (ARIMA + NNAR Hybrid)** outperformed all other models by combining linear and nonlinear components. The ARIMA model captured trend and autocorrelation, while the NNAR learned residual nonlinearities. This hybridization proved especially effective in reducing bias and improving generalization.

This report reflects our systematic approach to model selection, validation, and forecasting. We used reproducible workflows and version control via GitHub throughout the project.

Acknowledgment of AI Assistance

ChatGPT was used for troubleshooting code syntax. All models and forecasts were executed and validated locally in RStudio by the authors.