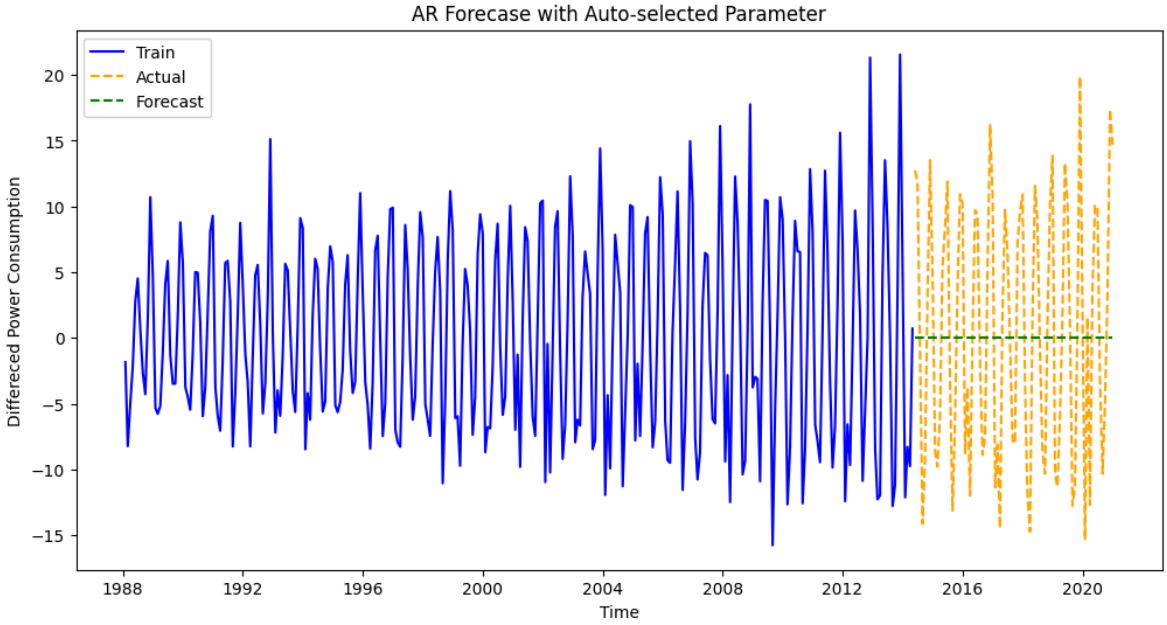
In [6]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.ar_model import AutoReg, ar_select_order
from statsmodels.stats.diagnostic import acorr_ljungbox
```

In [7]:
```
df = pd.read_csv(r"C:\Users\HP\OneDrive\Desktop\NLP\Data\ML471_S2_Datafile_Conce

df['Datetime'] = pd.to_datetime(df['Datetime'])
df.set_index('Datetime', inplace=True)

df = df.asfreq('h')

series = df['Power_Consumption_diff'].dropna()

train_size = int(len(series) * 0.8)
train, test = series[:train_size], series[train_size:]

best_aic = np.inf
best_lag = None

for lag in range(1, 15):
    try:
        model = AutoReg(train, lags=lag, freq='MS').fit()
        if model.aic < best_aic:
            best_aic = model.aic
            best_lag = lag
    except Exception as e:
        continue
print("Best AR lag:", best_lag)
print("Best AIC:", best_aic)

final_model = AutoReg(train, lags=best_lag).fit()

print("AR MODEL SUMMARY")
print(final_model.summary())

ljung_box = acorr_ljungbox(final_model.resid, lags=best_lag, return_df=True)
print("LJUNG-BOX TEST")
print(ljung_box)

forecast = final_model.predict(start=test.index[0], end=test.index[-1])
```

```
Best AR lag: None
Best AIC: inf
AR MODEL SUMMARY
                            AutoReg Model Results
==============================================================================
=
Dep. Variable:     Power_Consumption_diff   No. Observations:                  31
6
Model:                          AutoReg(0)   Log Likelihood               -1074.21
0
Method:                   Conditional MLE   S.D. of innovations              7.24
6
Date:                    Wed, 28 Jan 2026   AIC                           2152.42
1
Time:                            14:54:43   BIC                           2159.93
2
Sample:                          02-01-1988   HQIC                          2155.42
1
                               - 05-01-2014
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0556      0.408      0.136      0.892      -0.743       0.855
==============================================================================
LJUNG-BOX TEST
       lb_stat      lb_pvalue
1    44.056674   3.190043e-11
2   103.740249   2.972290e-23
3   300.163329   9.171067e-65
4   351.003089   1.065114e-74
5   399.785281   3.300223e-84
6   575.070226  5.552131e-121
7   620.394768  9.863019e-130
8   673.504169  3.614783e-140
9   864.252440  3.108019e-180
10  914.091664  5.901388e-190
c:\Users\HP\AppData\Local\Programs\Python\Python312\Lib\site-packages\statsmodels
\tsa\base\tsa_model.py:473: ValueWarning: No frequency information was provided,
so inferred frequency MS will be used.
  self._init_dates(dates, freq)
c:\Users\HP\AppData\Local\Programs\Python\Python312\Lib\site-packages\statsmodels
\tsa\deterministic.py:308: UserWarning: Only PeriodIndexes, DatetimeIndexes with
a frequency set, RangesIndexes, and Index with a unit increment support extendin
g. The index is set will contain the position relative to the data length.
  fcast_index = self._extend_index(index, steps, forecast_index)
```

In [8]:

```
plt.figure(figsize=(12,6))
plt.plot(train.index, train, label='Train', color='blue')
plt.plot(test.index, test, label='Actual', color='orange', linestyle='--')
plt.plot(forecast.index, forecast, label='Forecast', color='green', linestyle="-

plt.title("AR Forecase with Auto-selected Parameter")
plt.xlabel("Time")
plt.ylabel("Differeced Power Consumption")
plt.legend()
plt.show()
```


AR Forecase with Auto-selected Parameter
Train
Actual
Forecast
Differeced Power Consumption
20
15
10
5
0
−5
−10
−15
1988
1992
1996
2000
2004
2008
2012
2016
2020
Time