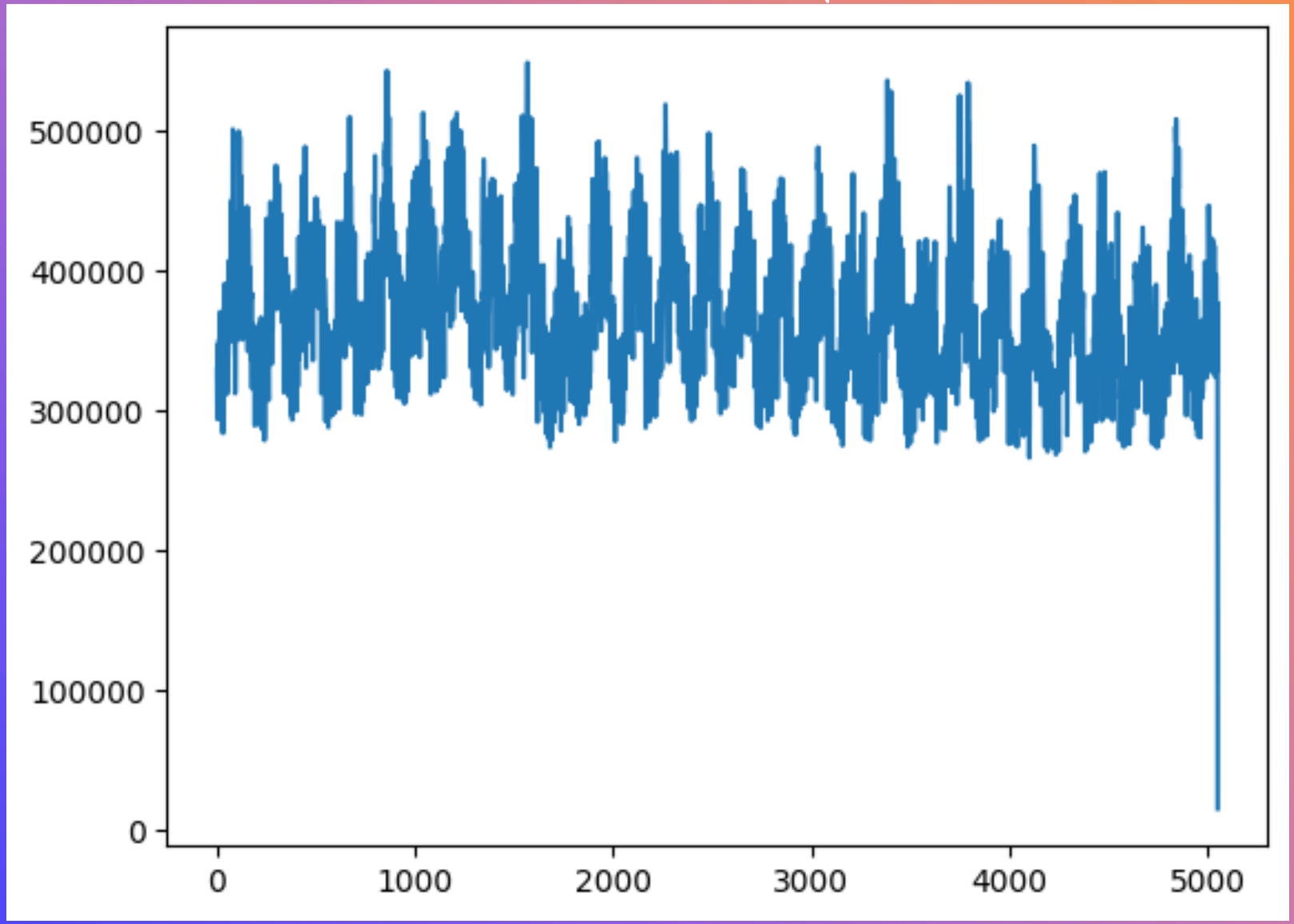
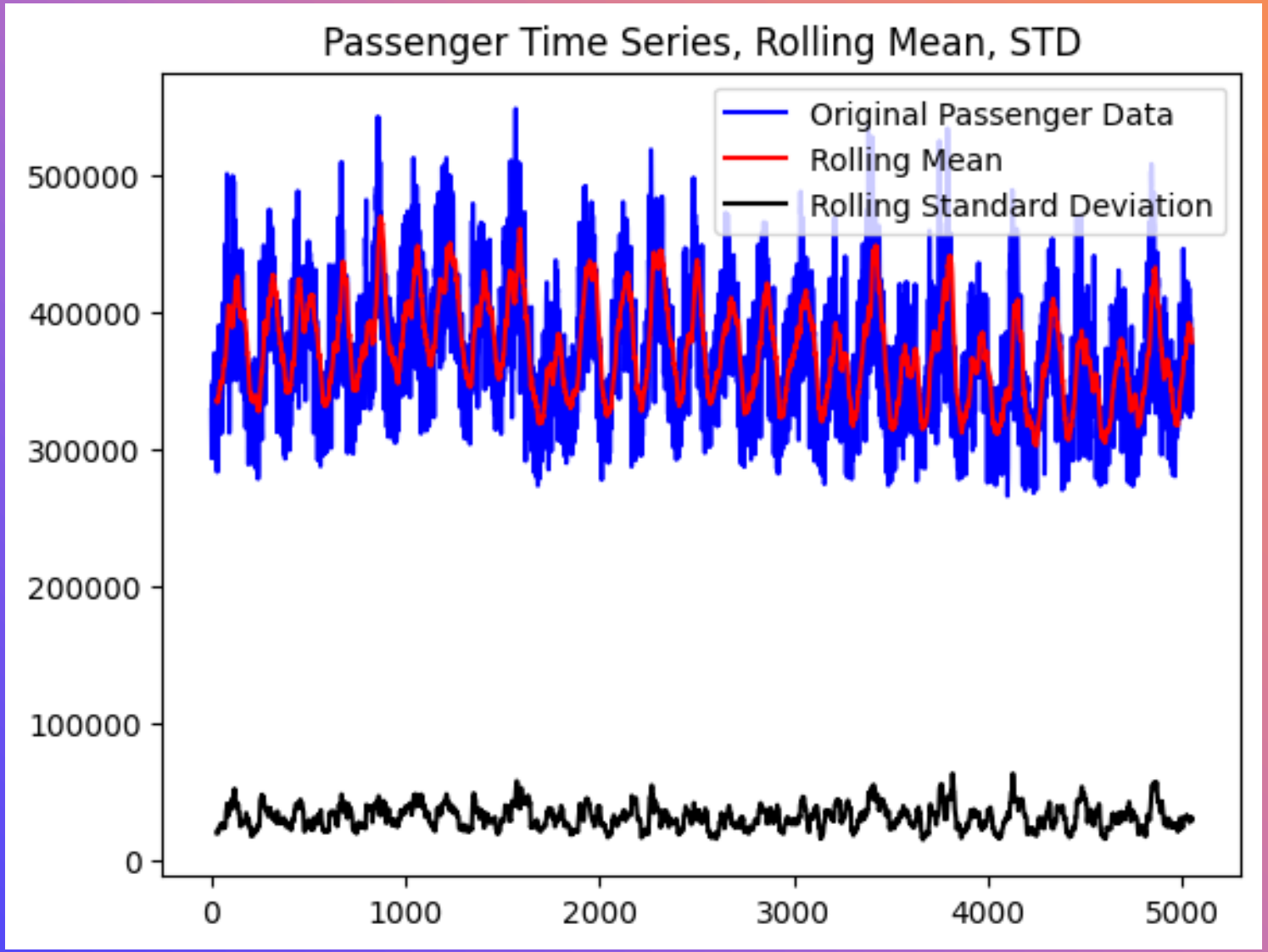
ELECTRIC POWER
MICHAELL ABELARD HENDRA 71487

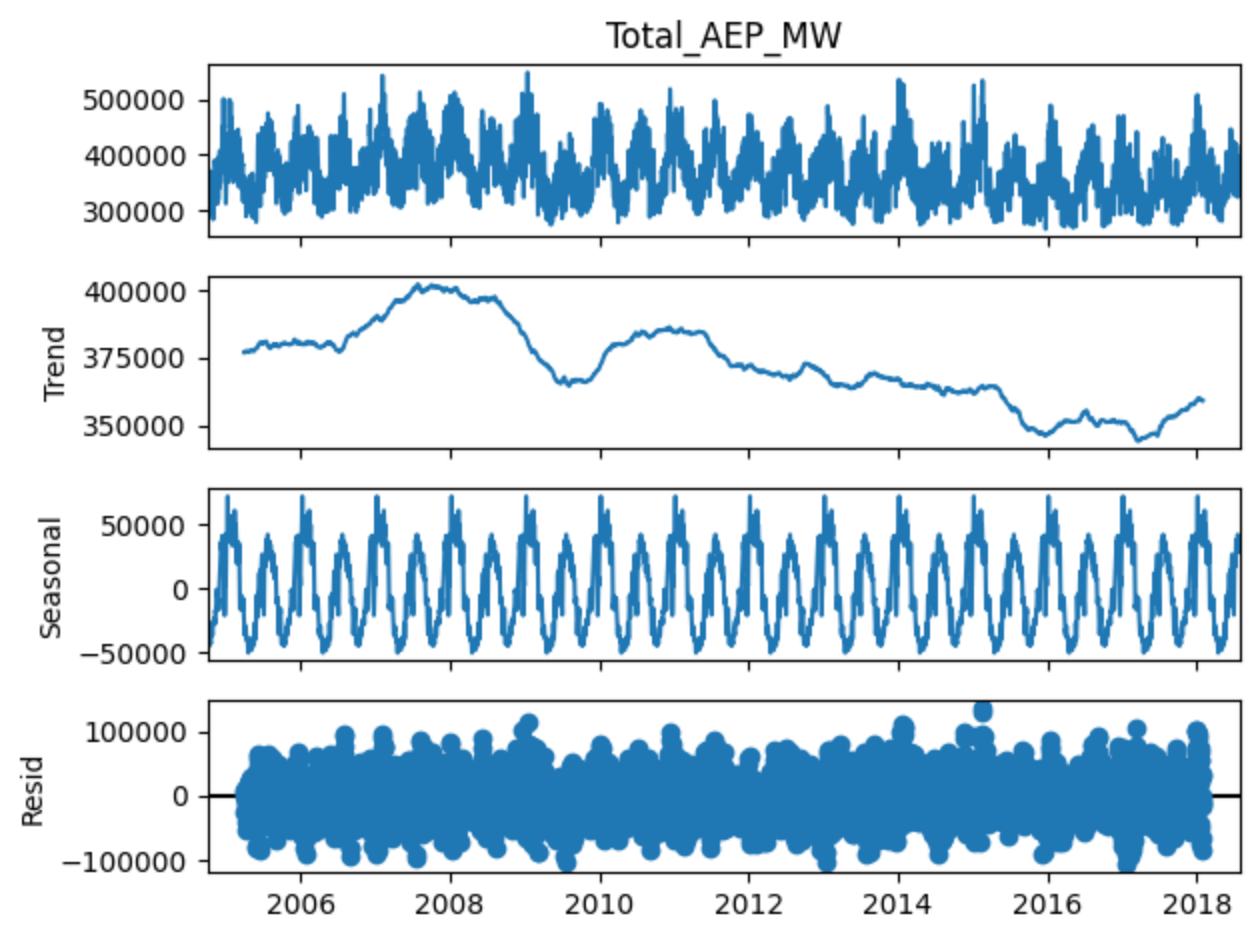
DATA


500000
400000
300000
200000
100000
0
0
1000
2000
3000
4000
5000

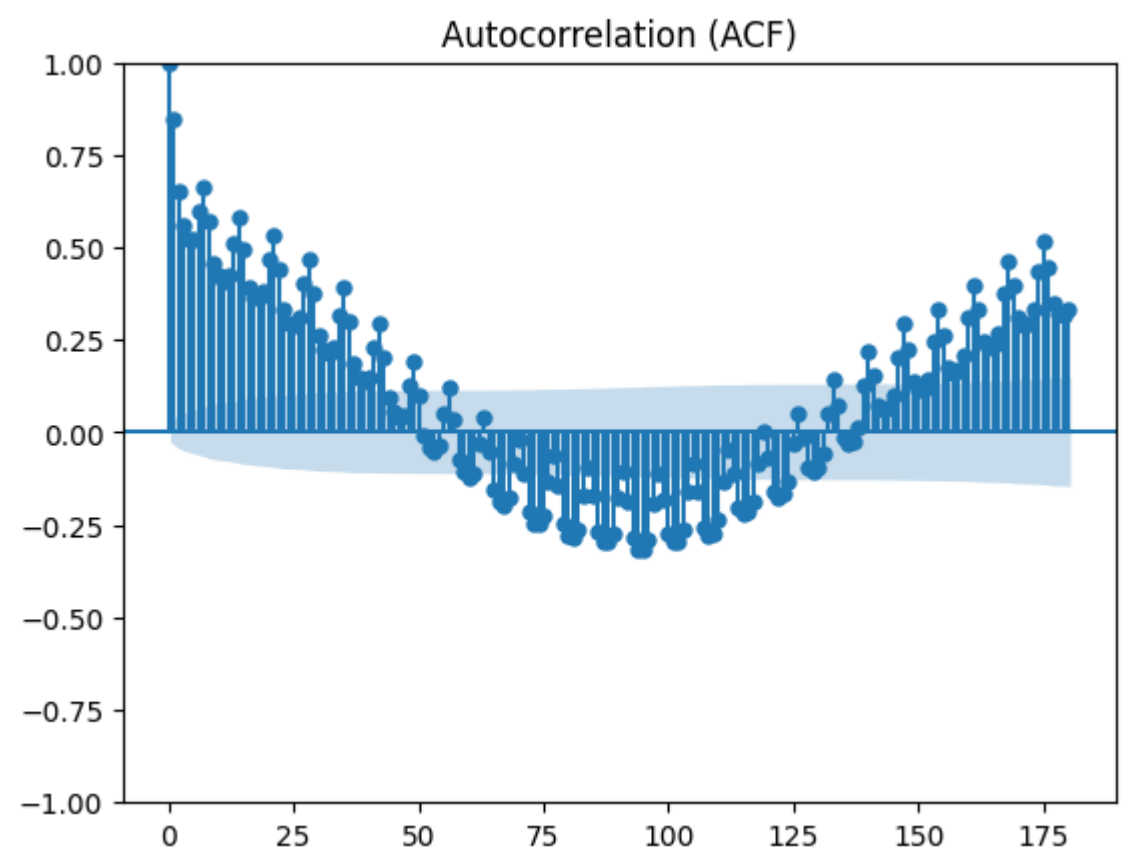
ROLLING MEAN (30)

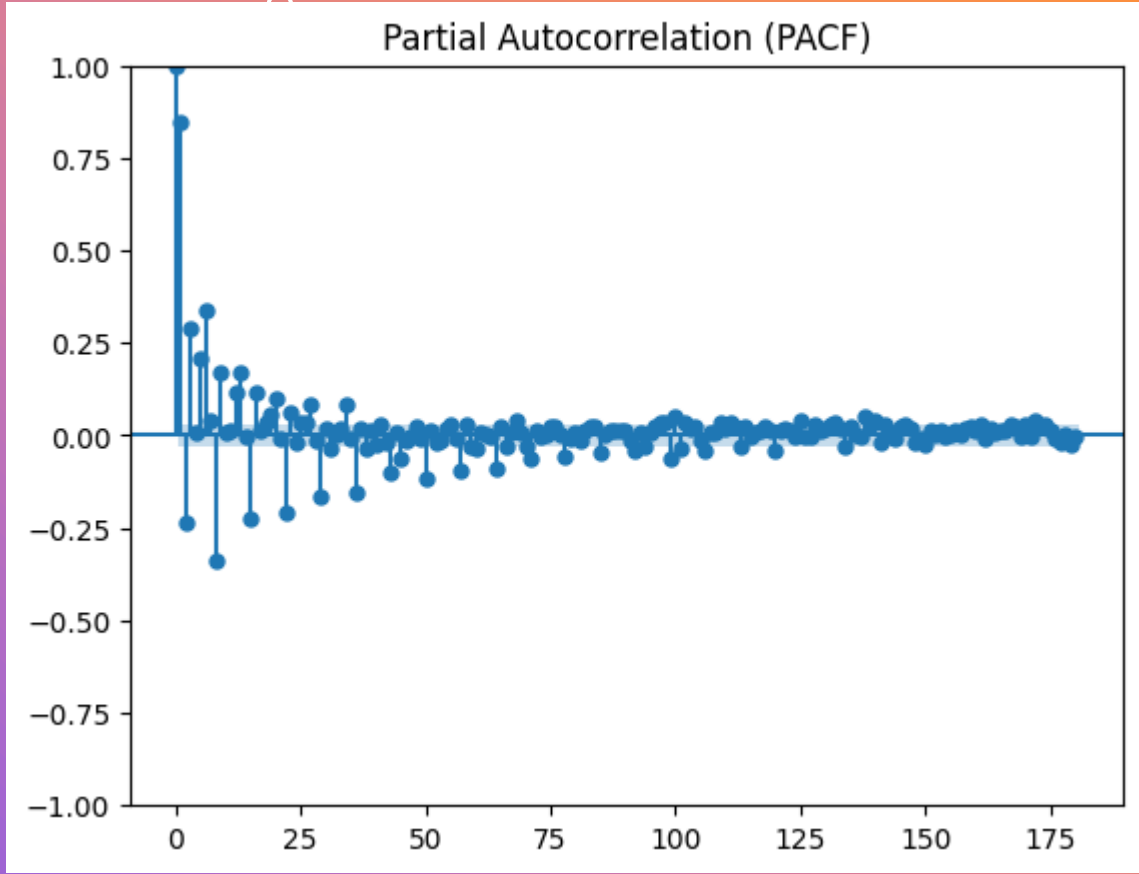


DESCRIBE LAG(365)

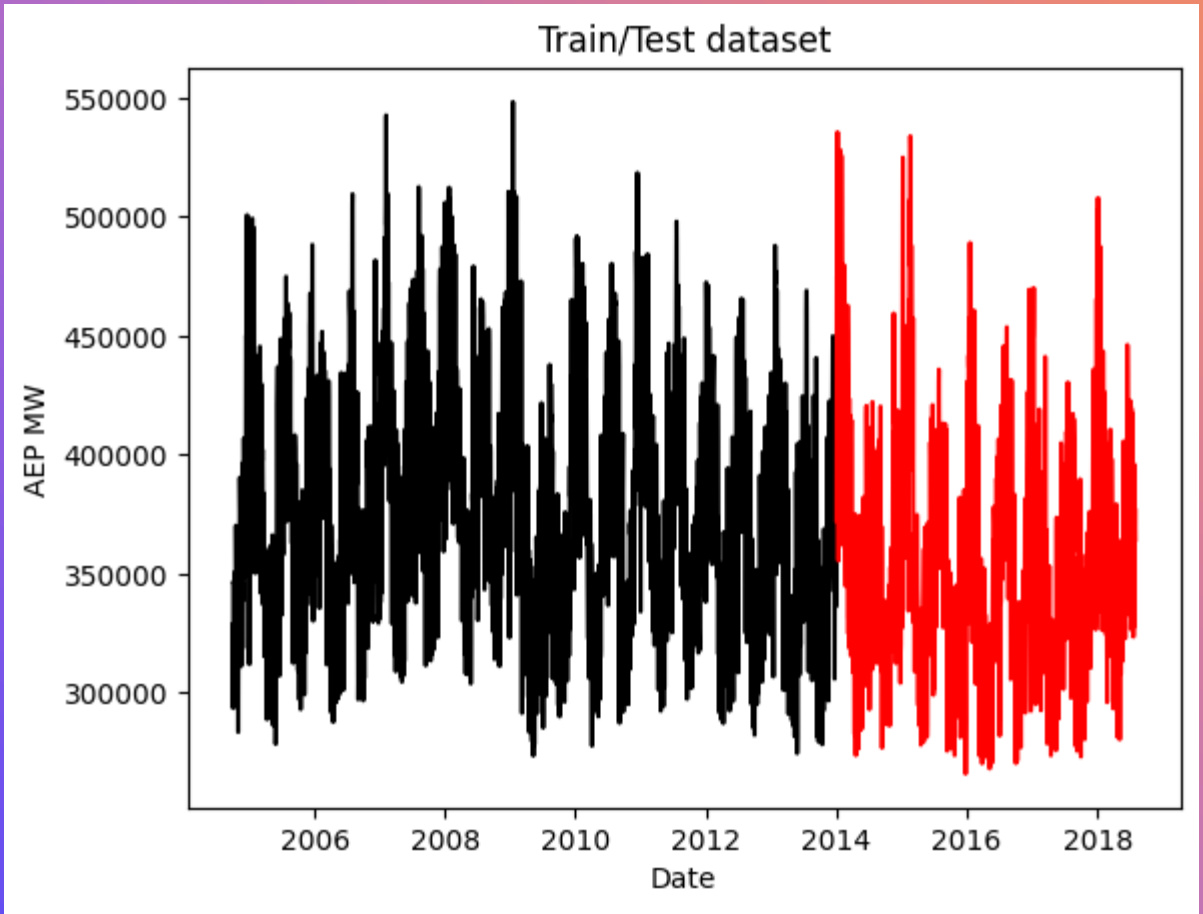


ACF & PACF


Autocorrelation (ACF)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
25
50
75
100
125
150
175


Partial Autocorrelation (PACF)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
25
50
75
100
125
150
175

DATA


Train/Test dataset
550000
500000
450000
400000
350000
300000
AEP MW
2006
2008
2010
2012
2014
2016
2018
Date

ARIMA(2,1,2)

SARIMAX Results

Dep. Variable:	train	No. Observations:	3379
Model:	ARIMA(2, 1, 2)	Log Likelihood	-38721.823
Date:	Thu, 19 Jun 2025	AIC	77453.645
Time:	14:05:31	BIC	77484.270
Sample:	10-01-2004 - 12-31-2013	HQIC	77464.595
Covariance Type:	opg		

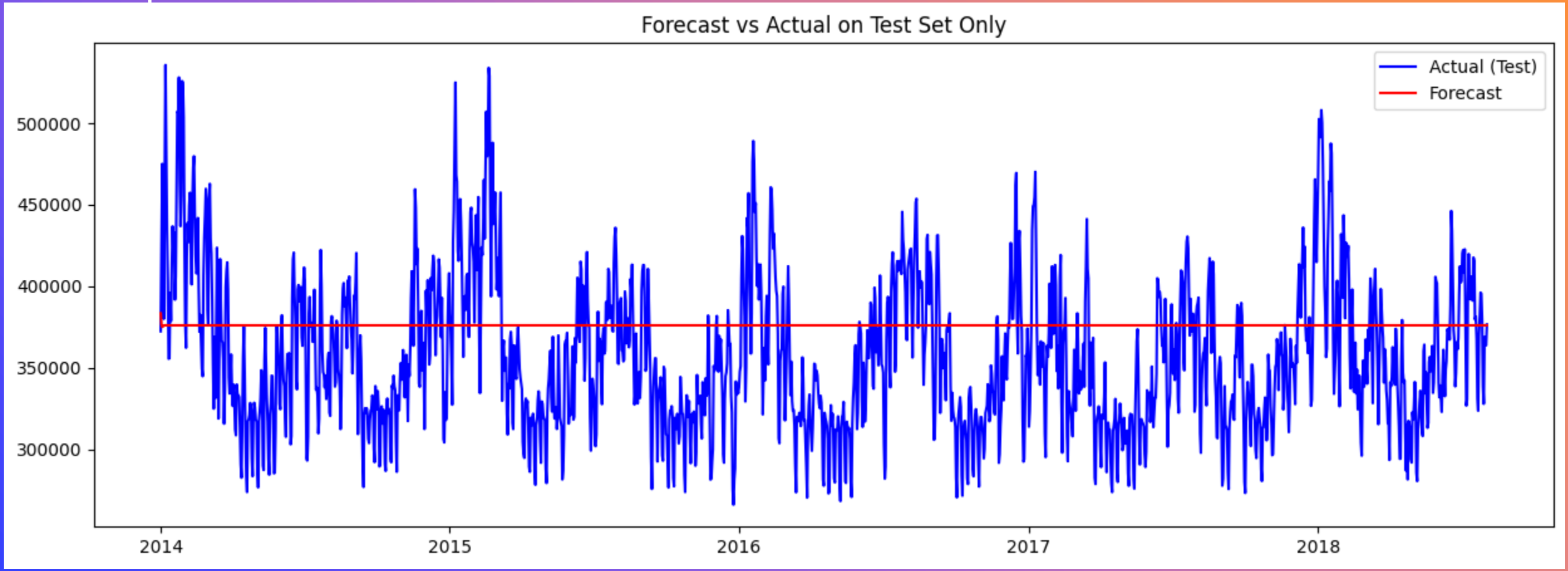
	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.5132	0.030	17.245	0.000	0.455	0.572
ar.L2	-0.1629	0.026	-6.243	0.000	-0.214	-0.112
ma.L1	-0.4580	0.028	-16.570	0.000	-0.512	-0.404
ma.L2	-0.3472	0.028	-12.571	0.000	-0.401	-0.293
sigma2	4.79e+08	1.13e-11	4.24e+19	0.000	4.79e+08	4.79e+08

Ljung-Box (L1) (Q):	0.29	Jarque-Bera (JB):	66.26
Prob(Q):	0.59	Prob(JB):	0.00
Heteroskedasticity (H):	0.81	Skew:	0.25
Prob(H) (two-sided):	0.00	Kurtosis:	3.47

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 2.88e+35. Standard errors may be unstable.

RMSE: 50106.41
Relative: 14.13%

ARIMA(2,1,2)

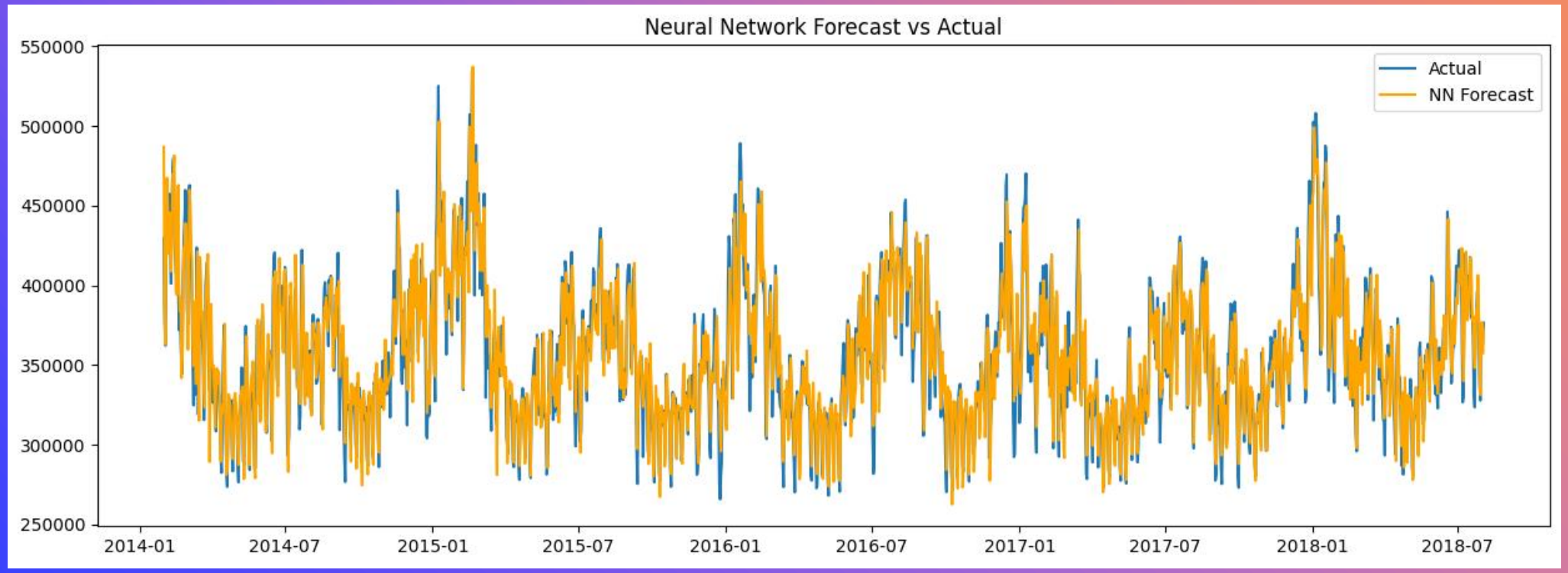


RMSE: 50106.41
Relative: 14.13%

NEURAL NETWORK

```
model = Sequential()
model.add(Dense(64, activation='relu', input_shape=(lookback,)))
model.add(Dense(32, activation='relu'))
model.add(Dense(1))  # output 1 hari ke depan

model.compile(optimizer='adam', loss='mse')
model.fit(X_train, y_train, epochs=20, batch_size=32, verbose=1)
```

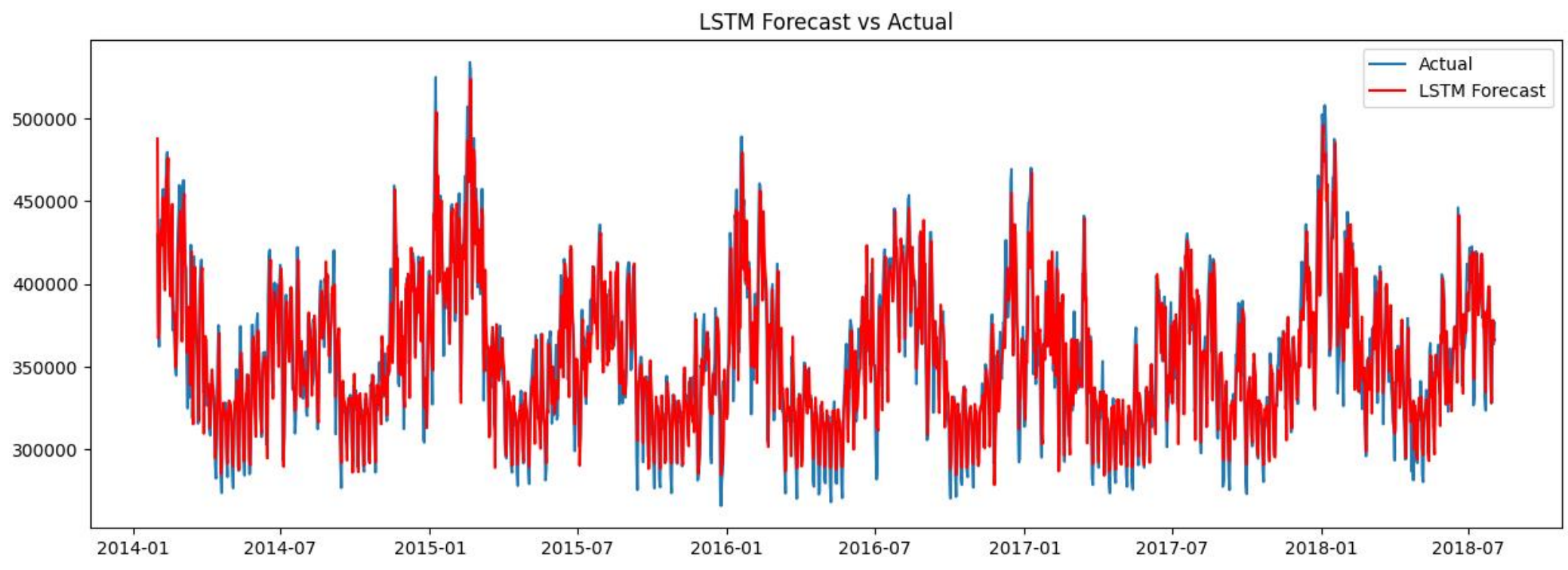
RMSE: 18503.98
Relative: 5.20%



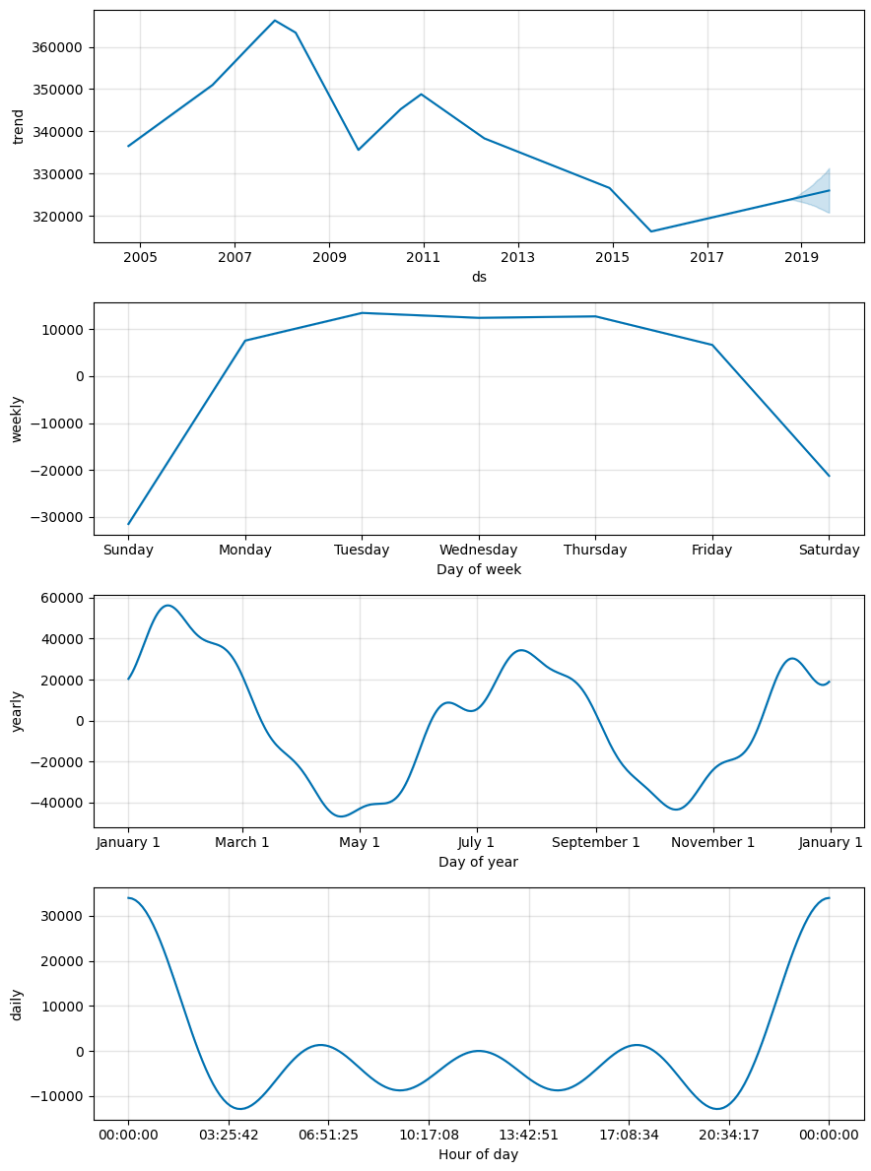
LSTM

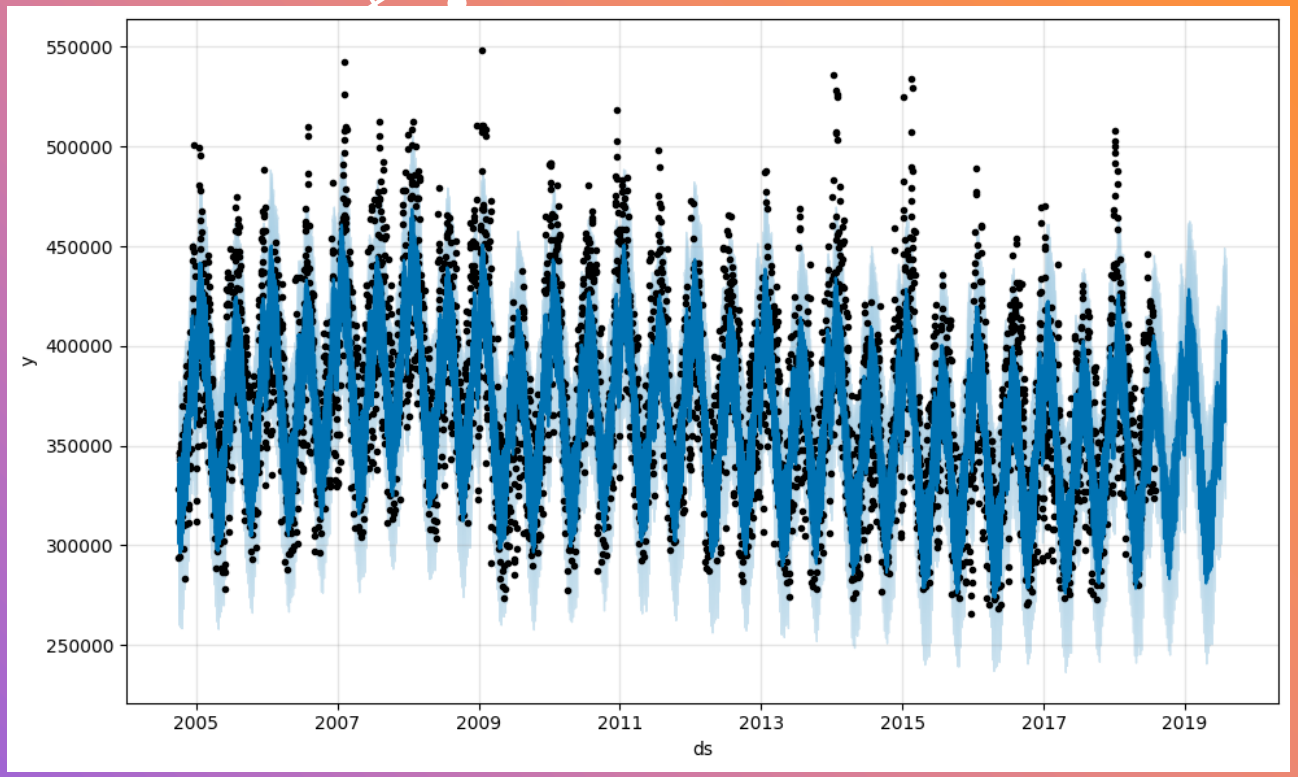
```python
model = Sequential()
model.add(LSTM(64, activation='relu', return_sequences=True, input_shape=(lookback, 1)))
model.add(LSTM(32, activation='relu', return_sequences=True))  # penting!
model.add(LSTM(16, activation='relu'))  # terakhir, return_sequences=False (default)
model.add(Dense(1))


optimizer = Adam(learning_rate=0.001)  # contoh: learning rate lebih kecil
model.compile(optimizer=optimizer, loss='mse')
```

RMSE: 22717.48
Relative: 5.28%



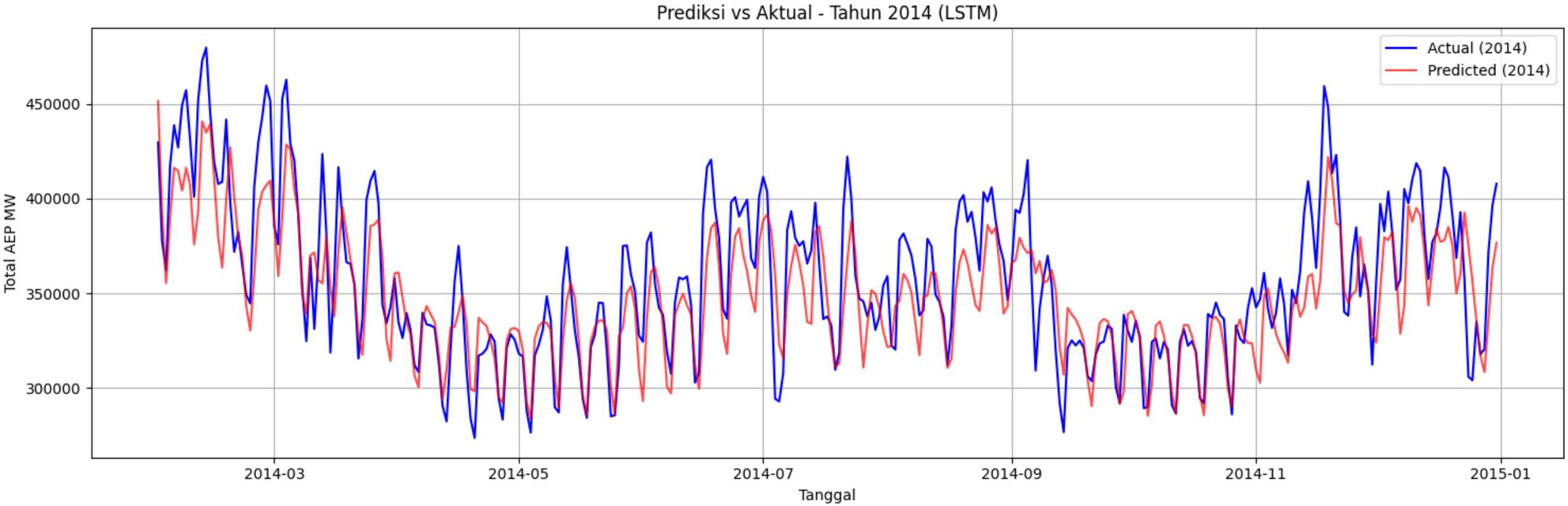
PROPHET





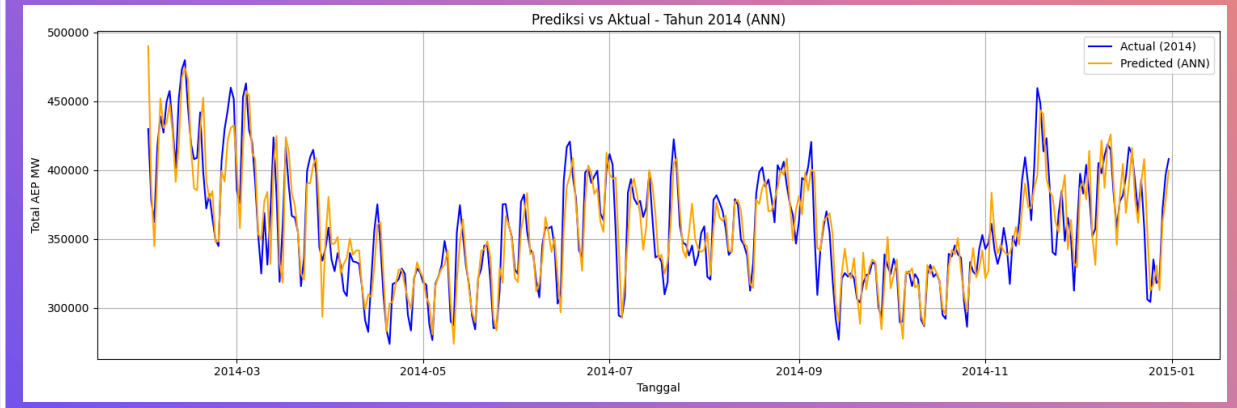
RMSE: 31884.45
Relative: 8.99%

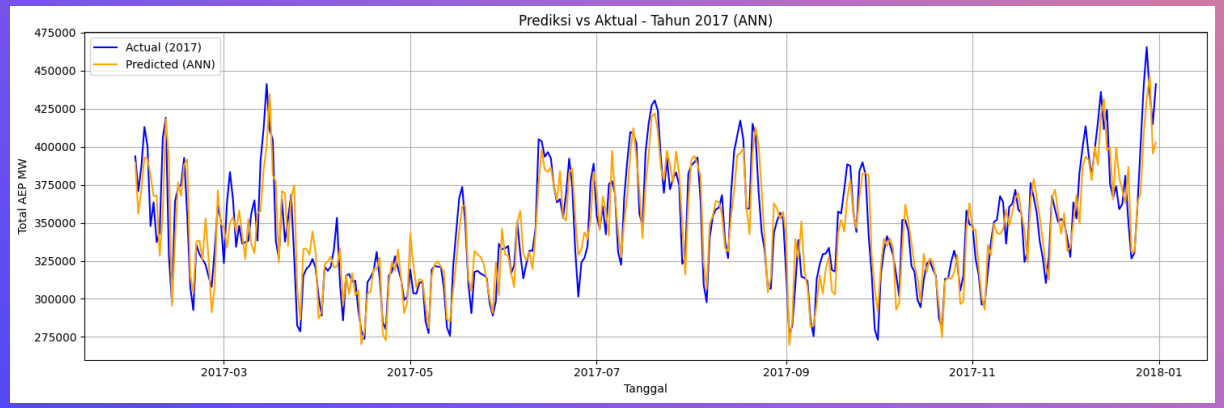
LSTM (2014-2015)

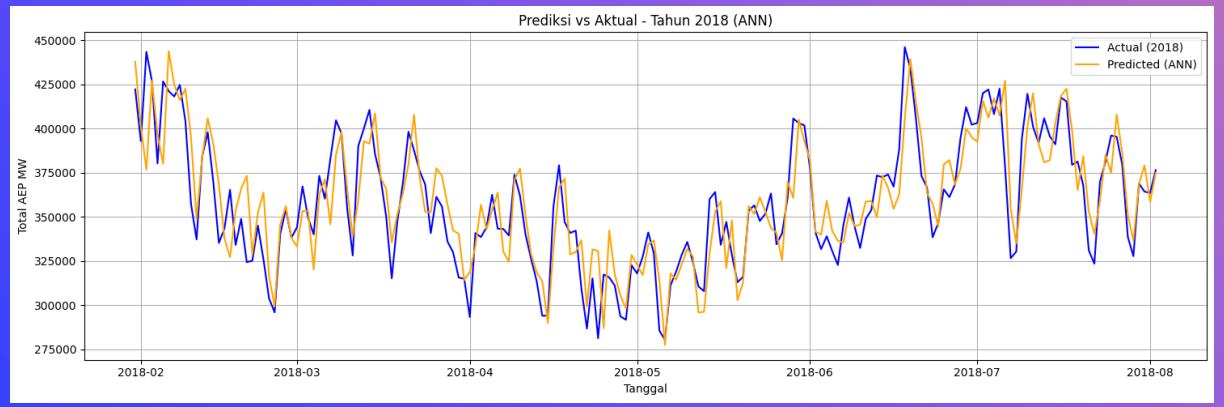


RMSE: 23807.96
Relative: 6.71%

ANN (2014-2016)







2014:
RMSE: 18186.03
Relative: 5.13%
2015:
RMSE: 17683.81
Relative: 4.99%
2016:
RMSE: 18357.98
Relative: 5.18%
2017:
RMSE: 16197.32
Relative: 4.69%
2018:
RMSE: 17923.32
Relative: 5.01%

MODEL METRICS

Model	RMSE	%
ARIMA(2,1,2)	50106.41	14.13%
LSTM	22717.48	5.28%
NN	18503.98	5.20%
Prophet	31884.45	8.99%
LSTM (Split)	23807.96	6.71%
ANN (Split)	17669.29	5%

THANK YOU