

Споредба на цените на HURX во период 2018-2024, нивно предвидување и паметно менаџирање на батериски систем

Изработил: Столе Јовановски – 221112

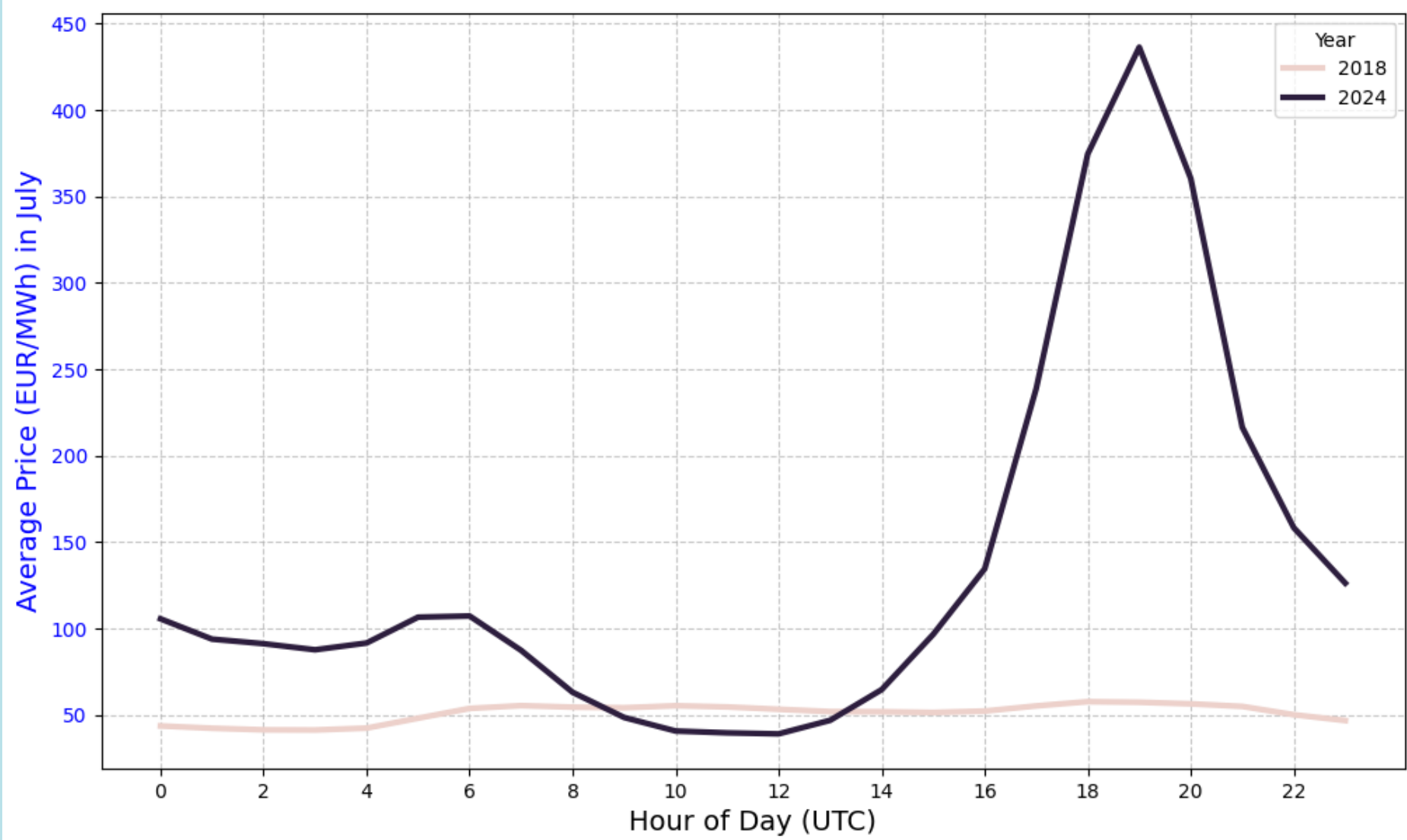
Ментор: проф. д-р Александра Дединец

Зошто е важно паметното управување со енергетските системи?

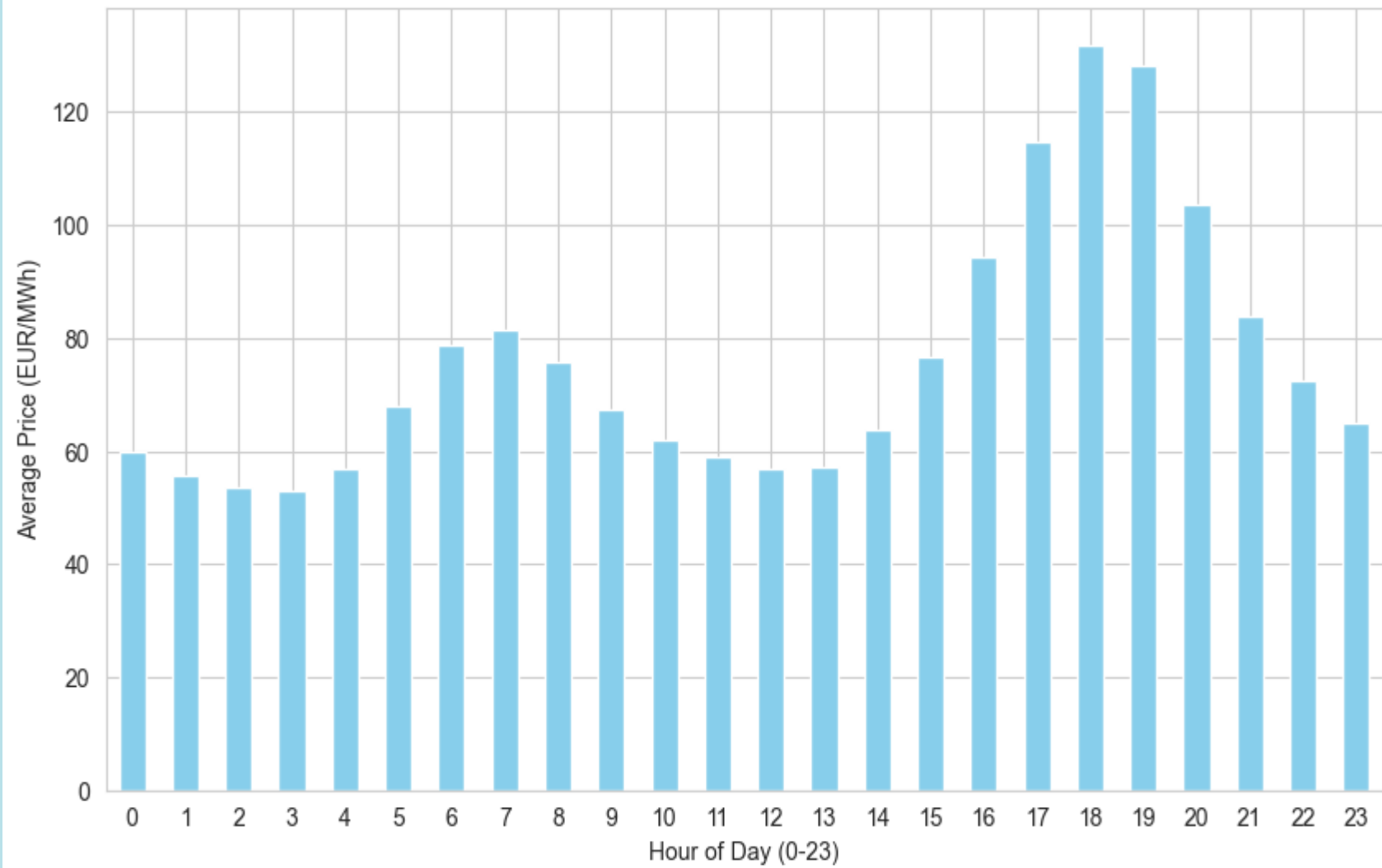
1. Новите технологии обезбедуваат нови извори на електрична енергија
2. Сепак овие нови обновливи извори на енергија (ОИЕ) знаат да бидат нестабилни, непредвидливи во нивното производство.
3. Идејата за овој проект е како да се намали неизвесноста при производство, купување или користење на ел. енергија од вакви извори.
4. Да се знае кои се очекувањата и како да се прилагодиме на новите времиња.

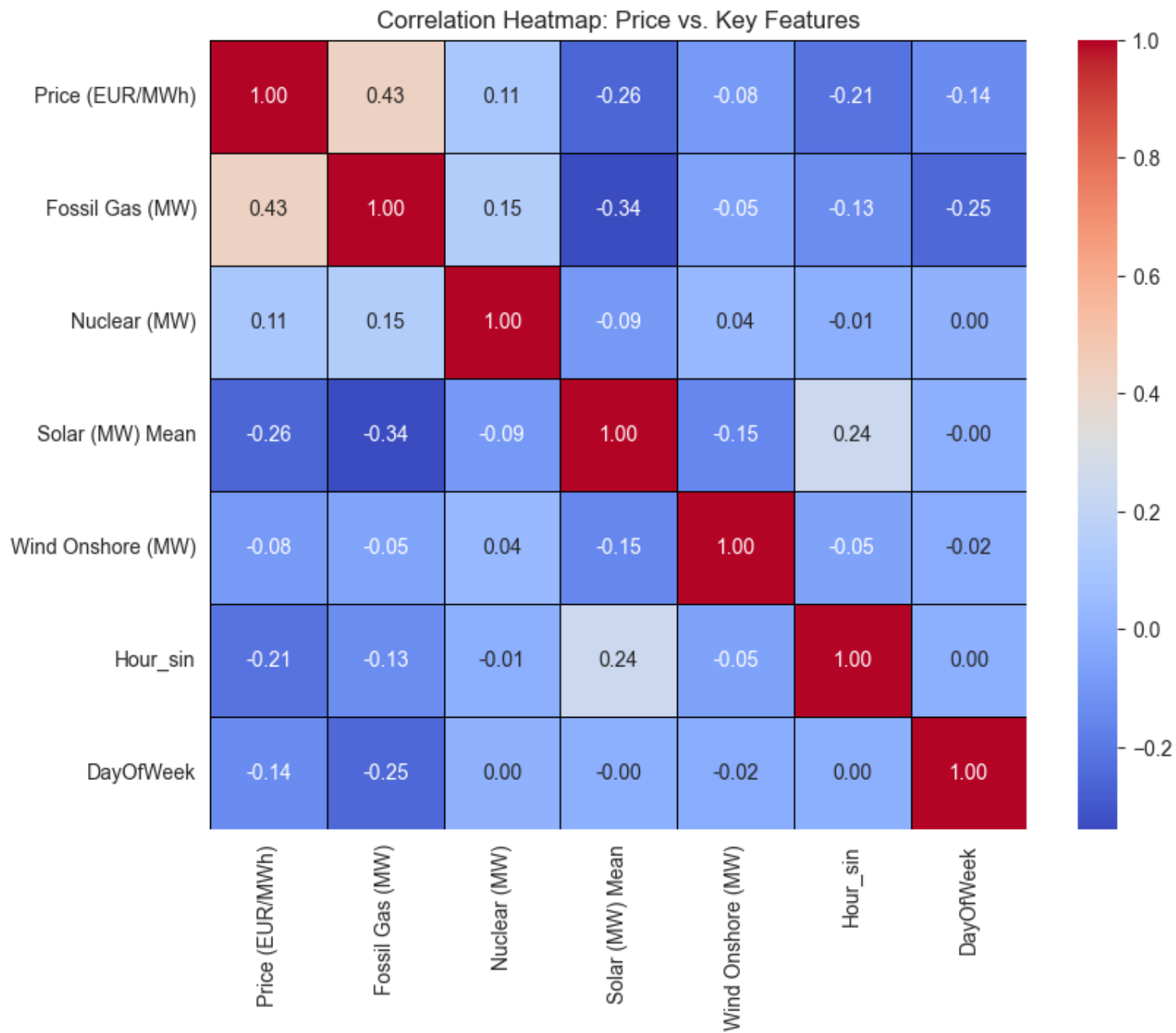
Производство на ел. енергија со помош на соларни панели:

Година	Број на часови	Просечно производство	Максимално производство
2018	0	NaN	NaN
2024	8760	649.01	3330.58

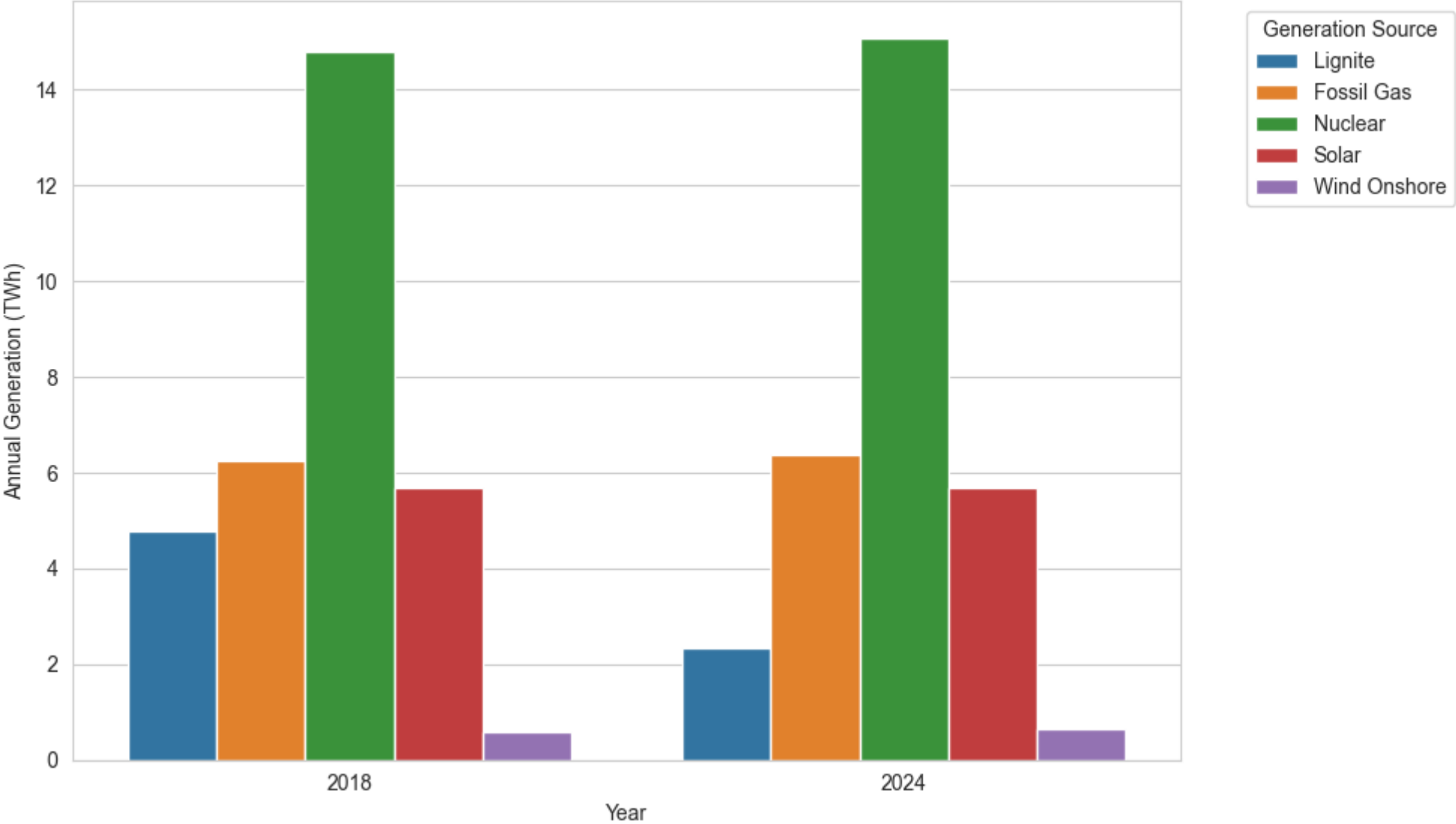


Average Hourly Price Profile (2018 & 2024)

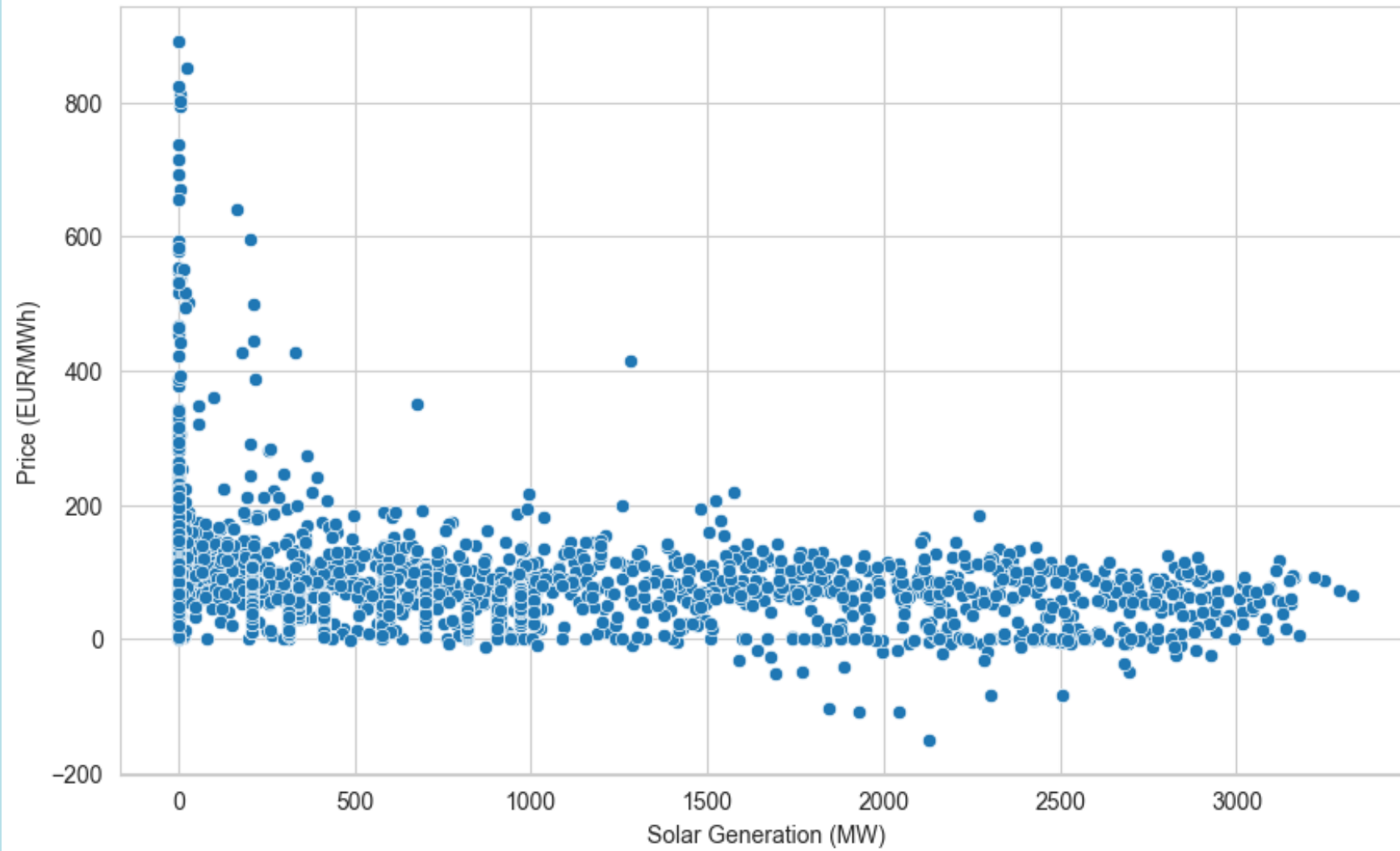




Annual Electricity Generation Mix Comparison (TWh)



Price vs. Solar Generation (Sample)



READY

READY

Приказ на MLPRegressor невронска мрежа

Зошто MLPRegressor ?

1. Моделирање на нелинеарни врски
2. Способност за обработка на многу карактеристики (High Dimensionality)
3. Флексибилност на архитектурата

```
X = df_train[features].values
y = df_train[target].values

# Резервна опција за чистење на NaN (за робустност)
X[np.isnan(X)] = np.nanmean(X)
y[np.isnan(y)] = np.nanmean(y)

scaler_X = MinMaxScaler()
scaler_y = MinMaxScaler()
X_scaled = scaler_X.fit_transform(X)
y_scaled = scaler_y.fit_transform(y.reshape(-1, 1))

# Поделба на податоци
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y_scaled, test_size=0.2, random_state=42)

# 4.3. Дефинирање и тренирање на моделот (MLP Tuning)
# Зголемување на бројот на итерации и невроните за да се искористат новите латентни карактеристики
mlp_model = MLPRegressor(
    hidden_layer_sizes=(150, 100, 50), # Поголема мрежа
    activation='relu',
    solver='adam',
    max_iter=1000, # Повеќе итерации
    random_state=42,
    verbose=False,
    early_stopping=True,
    n_iter_no_change=30 # Подолга толеранција за подобро тренирање
)

print("Тренирање на MLP модел...")
mlp_model.fit(X_train, y_train.ravel())
print("MLP моделот е трениран.")

# 4.4. Евалуација на моделот
y_pred_scaled = mlp_model.predict(X_test)

# Враќање на предвидувањата во оригинална скала
y_pred = scaler_y.inverse_transform(y_pred_scaled.reshape(-1, 1))
y_test_original = scaler_y.inverse_transform(y_test)

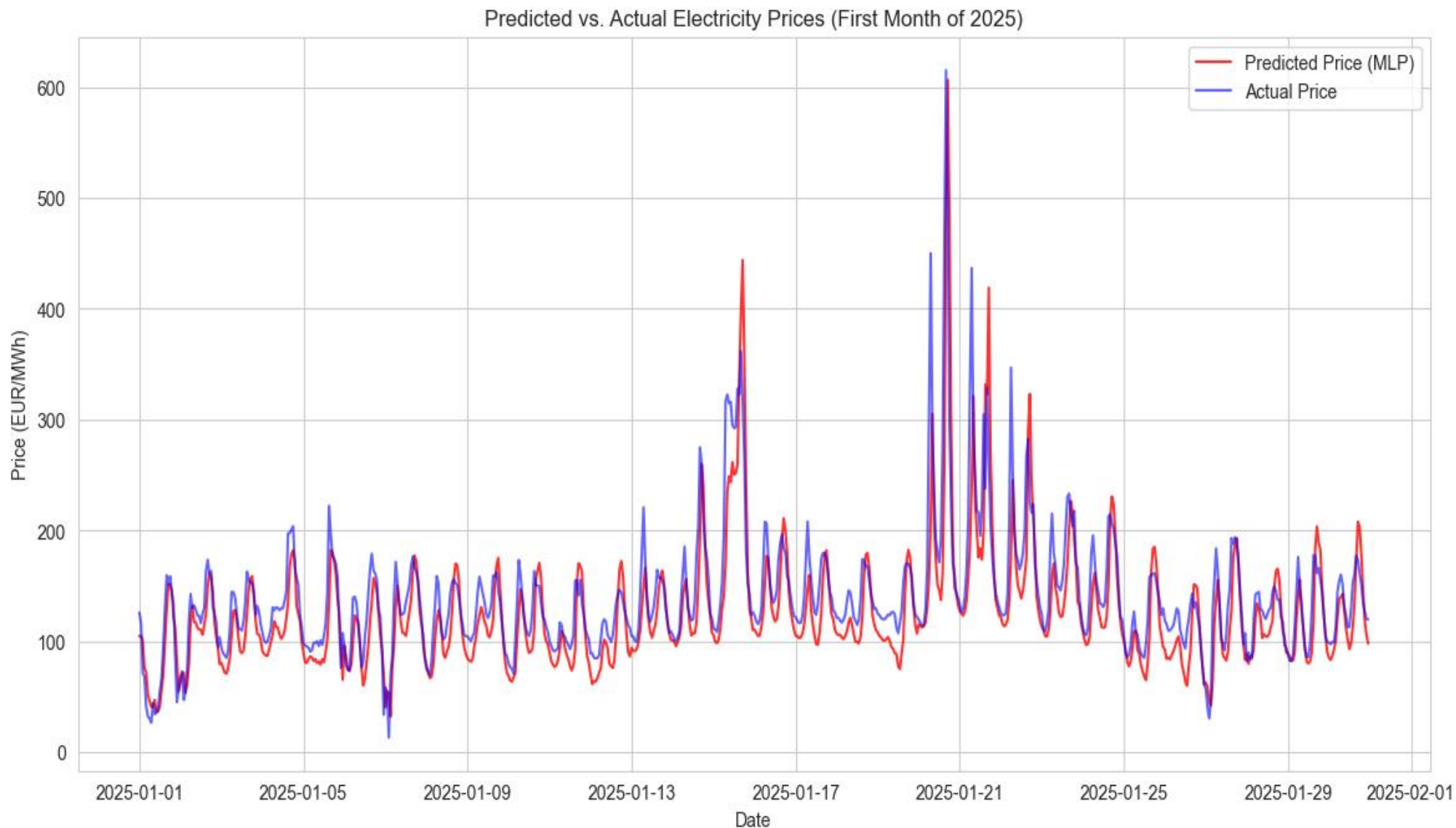
# Евалуациски метрики
rmse = np.sqrt(mean_squared_error(y_test_original, y_pred))
mae = mean_absolute_error(y_test_original, y_pred)
r2 = r2_score(y_test_original, y_pred)
```

Евалуација на предвидувањата за 2025 (за достапните актуелни цени)

RMSE: 29.92 EUR/MWh

MAE: 20.30 EUR/MWh

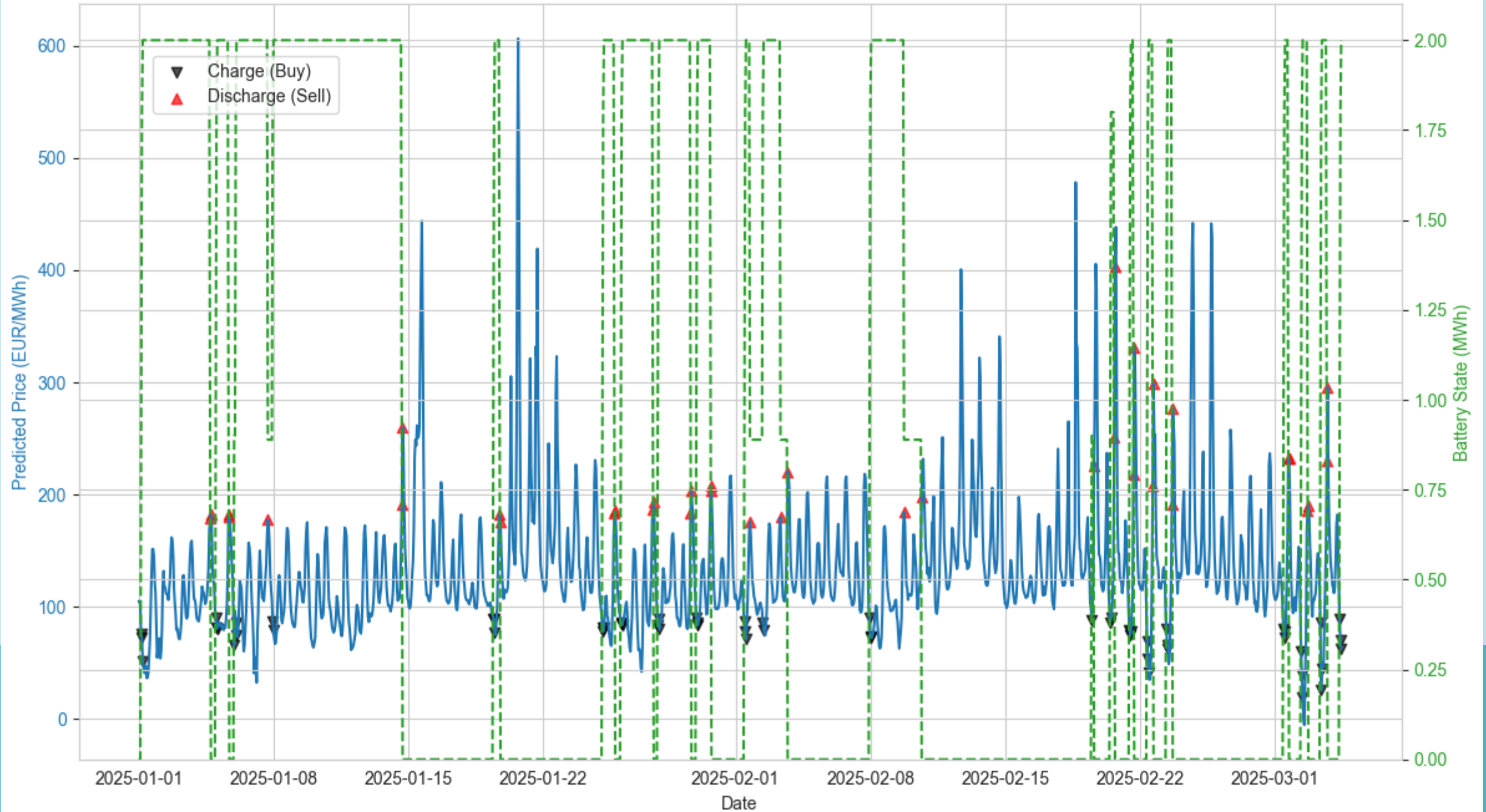
R2: 0.7603



- Параметри на батеријата: 2 MWh / 1 MW
- Ефикасност на полнење: 0.9
- Ефикасност на празнење: 0.9
- Ова ни значи дека при секој 1 мегават струја што ја полниме/празниме батеријата имаме загуба од 10%
- Затоа при правење на математиката треба да вклучиме дека треба да купиме малце над 2MW струја (околу 2.2MW) за да ја наполниме батеријата до максимум.
- Нашата батерија ќе ја полниме со ел. енергија само кога цената на струјата на берза ќе падне под 15 перцентил, а ќе ја празниме (трошиме) кога цената ќе го достигне 85 перцентил
- Вкупен приход од трговија (за симулирани 1500 часа, нешто над 2 месеци): 10,397.74 EUR
- Проекција на годишен приход: 60,722.77 EUR

Што доколку сакаме да имплементираме батериски систем за нашите потреби?

Battery Arbitrage Simulation: 2MWh / 1MW (2025 Forecast)



Благодарам на вниманието