**INTRODUCTION :**

Briefly explain the importance of accurate electricity price forecasting.State the objective of your innovative approach.Problem Statement.Define the problem of predicting future electricity prices.Discuss the limitations of traditional forecasting methods.Innovation Proposal.Introduce the use of advanced time series forecasting techniques.Explain the choice of Prophet and deep learning models.Prophet Model.Provide an overview of the Prophet forecasting model.Explain why it's suitable for electricity price forecasting.Discuss its key features and advantages.Deep Learning Models.Explore the use of deep learning for time series forecasting.Mention popular deep learning models like LSTM, GRU, and CNN.Discuss how deep learning can capture complex patterns in electricity price data.Implementation.Describe the data preprocessing steps.Explain the model training process for both Prophet and deep learning models.Mention any specific libraries or tools used.Evaluation and Results.Present the evaluation metrics used to assess model performance (e.g., RMSE, MAE).Share the results of your experiments, including accuracy improvements.Comparison.Compare the performance of the Prophet model and deep learning models.Discuss trade-offs and situations where one might be preferred over the other.

**MODELS :**

**LSTM**

tf.keras.backend.clear\_session()

multivariate\_lstm = tf.keras.models.Sequential([

LSTM(100, input\_shape=input\_shape,

return\_sequences=True),

Flatten(),

Dense(200, activation='relu'),

Dropout(0.1),

Dense(1)

])

model\_checkpoint = tf.keras.callbacks.ModelCheckpoint(

'multivariate\_lstm.h5', monitor=('val\_loss'), save\_best\_only=True)

optimizer = tf.keras.optimizers.Adam(lr=6e-3, amsgrad=True)

multivariate\_lstm.compile(loss=loss,

optimizer=optimizer,

metrics=metric)

history = multivariate\_lstm.fit(train, epochs=120,

validation\_data=validation,

callbacks=[early\_stopping,

model\_checkpoint])

plot\_model\_rmse\_and\_loss(history)

multivariate\_stacked\_lstm = tf.keras.models.load\_model('multivariate\_stacked\_lstm.h5')

forecast = multivariate\_stacked\_lstm.predict(X\_test)

multivariate\_stacked\_lstm\_forecast = scaler\_y.inverse\_transform(forecast)

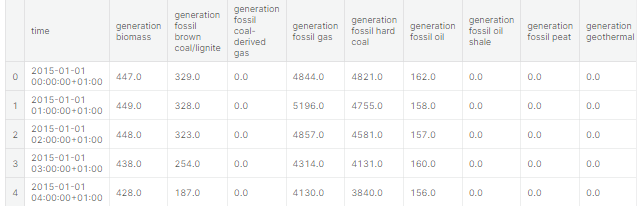
rmse\_mult\_stacked\_lstm = sqrt(mean\_squared\_error(y\_test\_inv,

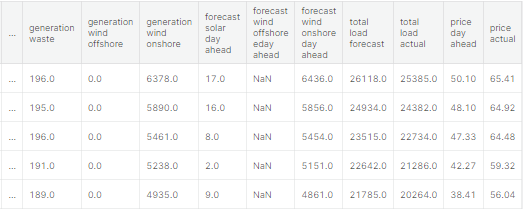
multivariate\_stacked\_lstm\_forecast))

print('RMSE of hour-ahead electricity price multivariate Stacked LSTM forecast: {}'

.format(round(rmse\_mult\_stacked\_lstm, 3)))

**DATASET :**





**TIME SERIES :**

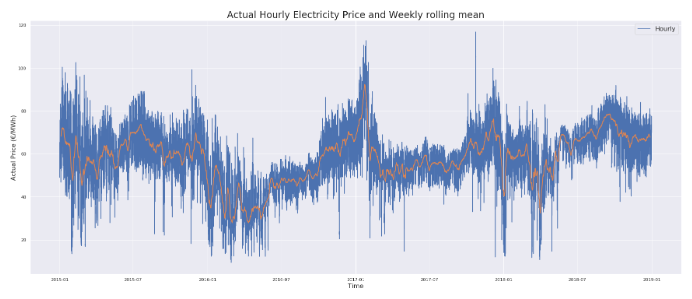
rolling = df\_final['price actual'].rolling(24\*7, center=True).mean()

ax = plot\_series(df\_final, 'price actual', label='Hourly', ylabel='Actual Price (€/MWh)',

title='Actual Hourly Electricity Price and Weekly rolling mean')

ax.plot(rolling, linestyle='-', linewidth=2, label='Weekly rolling mean')

plt.show()



# Plot the electricity price (monthly frequence) along with its 1-year lagged series

monthly\_price = df\_final['price actual'].asfreq('M')

ax = plot\_series(series=monthly\_price, ylabel='Actual Price (€/MWh)',

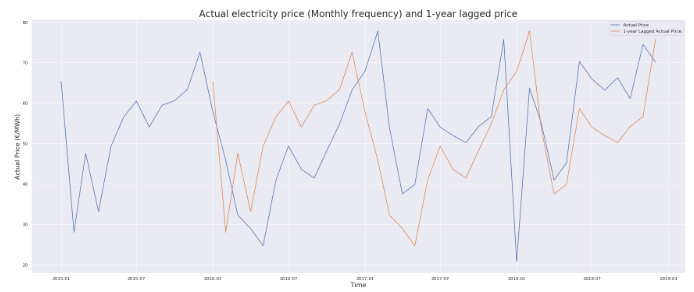
title='Actual electricity price (Monthly frequency) and 1-year lagged price')

shifted = df\_final['price actual'].asfreq('M').shift(12)

ax.plot(shifted, label='Hourly')

ax.legend(['Actual Price', '1-year Lagged Actual Price'])

plt.show()



**CONCLUSION :**

Summarize the key findings.Highlight the significance of your innovative approach.Recommendations suggest potential future improvements or refinements.Discuss scalability and real- world implementation.Include code snippets, if applicable.Share any additional resources or references used.References.List all the sources, papers, and references you used.Once you've created this document, you can share it for assessment as per the provided instructions. It should serve as a comprehensive guide to your innovative approach for predicting future electricity prices using Prophet and deep learning models.