

MEASURE ENERGY CONSUMPTION

Name: Vedagiri S

Naan Mudhalvan Id: au513521104054

Email Id: vedagirisankar2004@gmail.com

College: ANNAI MIRA COLLEGE OF ENGINEERING AND TECHNOLOGY

INTRODUCTION:

The concept of energy consumption is directly related to energy efficiency since higher consumption results in lower energy efficiency.

It's estimated that during an hour about 1,000 watts are consumed, so this measure is used to calculate the consumption of homes, businesses, or any other type of building in order to issue the corresponding bills.

There are various factors that directly influence energy consumption such as:

- The activity that takes place in the home or business.
- The number of people in a household or workers.
- The consumption habits of each person.
- The energy performance of household appliances.

With the right information and technology, it's possible to use energy more responsibly and efficiently. This results in a reduction in energy consumption and, therefore, in significant savings on utility bills.

DATA PREPROCESSING:

Load Modules

```
import numpy as np
```

```
import pandas as pd
```

```
from datetime import datetime
```

```
import warnings
```

```
warnings.filterwarnings('ignore')
```

Load Data

```
duq_df = pd.read_csv('data/DUQ_hourly.csv', index_col=[0], parse_dates=[0])
```

Sort Data

```
duq_df.sort_index(inplace=True)
```

Identify Duplicate Indices

```
duplicate_index = duq_df[duq_df.index.duplicated()]
```

Replace Duplicates with Mean Value

```
duq_df = duq_df.groupby('Datetime').agg(np.mean)
```

Set DatetimeIndex Frequency

```
duq_df = duq_df.asfreq('H')
```

Impute Missing Values

```
duq_df['DUQ_MW'] = duq_df['DUQ_MW'].interpolate(limit_area='inside', limit=None)
```

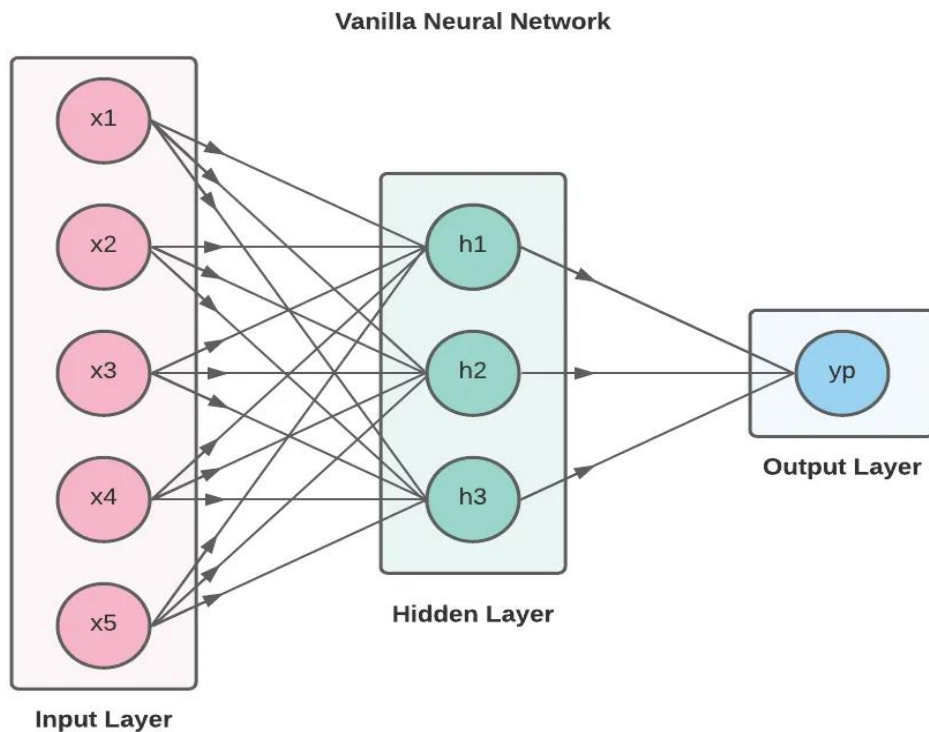
Create Train & Test Series

```
train_series = duq_df.loc[(duq_df.index >= datetime(2016, 1, 1)) & (duq_df.index < datetime(2018, 8, 1)), 'DUQ_MW']
```

```
test_series = duq_df.loc[(duq_df.index >= datetime(2018, 8, 1)), 'DUQ_MW']
```

LSTM Neural Networks:

To understand **Long Short-Term Memory (LSTM) Neural Networks**, we first have to understand **Recurrent Neural Networks (RNNs)**. Understanding RNNs is simpler once we have a grasp of how a simple, fully-connected, or ‘vanilla’, neural network works. Vanilla neural networks take an input vector with a fixed size, passing it to one or more hidden layers of neurons where an activation function is applied to the dot product of the input vector and a vector of weights. Weight vectors are adjusted through backpropagation during the training process. From the hidden layers, information is passed to an output vector which is used to make a prediction based on the input vector. It is assumed that all elements of the input vector are independent of one another.



Tensorflow:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM
from tensorflow.keras.preprocessing.sequence import
TimeseriesGenerator
from sklearn.preprocessing import MinMaxScaler
```

```
np.random.seed(73)
```

```
# Apply MinMaxScaler to training and test data. Fit to training data, use
to transform train + test
```

```
scaler = MinMaxScaler(feature_range=(0,1))
scaled_train_values = scaler.fit_transform(train_series.values.reshape(-
1,1))
train_series_scaled = pd.Series(data=scaled_train_values.reshape(1,-
1)[0], index=train_series.index)
```

```
# Create TimeseriesGenerator using training data and selected
hyperparameters
```

```
n_input = 24
n_features= 1
sampling_rate = 1
stride = 1
batch_size = 1
```

```
train_generator = TimeseriesGenerator(scaled_train_values,
                                     scaled_train_values,
                                     length=n_input,
                                     sampling_rate=sampling_rate,
                                     stride=stride,
                                     batch_size=batch_size)
```

```
# Build and fit model
```

```
model = Sequential()
```

```

model.add(LSTM(100, activation='relu', return_sequences=False,
input_shape=(n_input, n_features)))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')

model.fit(train_generator, epochs=80)

# Generate predictions

lstm_preds = []

batch = scaled_train_values[-n_input:]
current_batch = batch.reshape((1, n_input, n_features))

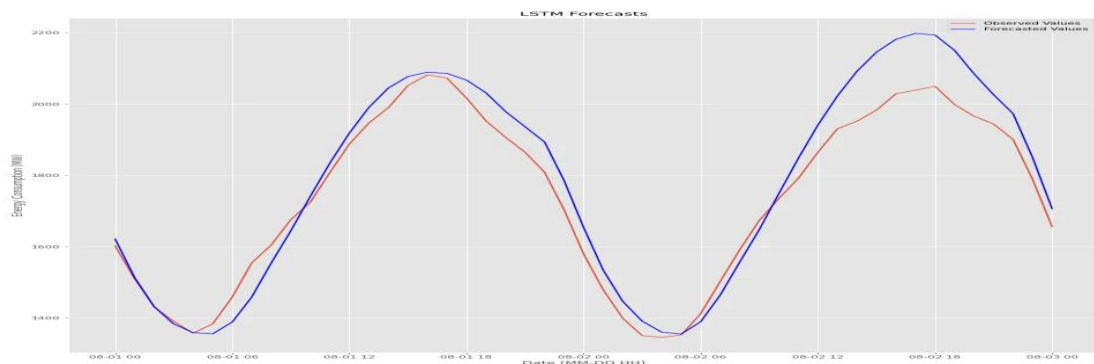
for i in range(len(test_series)):
    pred = model.predict(current_batch)[0]
    lstm_preds.append(pred)
    current_batch = np.append(current_batch[:, 1:, :], [[pred]], axis=1)

# Evaluate performance

lstm_preds = scaler.inverse_transform(lstm_preds)
lstm_preds = [item for sublist in lstm_preds for item in sublist]

test_score = mean_squared_error(test_series.values, lstm_preds)
print(test_score)

```



CONCLUSION:

- **Data-Driven Insights:** Machine learning algorithms effectively analyzed vast datasets, providing nuanced insights into energy consumption patterns. This data-driven approach has the potential to uncover hidden inefficiencies and anomalies that were previously challenging to detect.
- **Predictive Capabilities:** The predictive models developed in this project enable real-time forecasting of energy consumption, allowing organizations to make informed decisions and adjustments proactively.
- **Cost Savings:** By identifying energy-saving opportunities and enabling dynamic load management, machine learning can lead to substantial cost savings, making it a valuable investment for businesses and homeowners alike.
- **Environmental Impact:** Reducing energy consumption not only cuts costs but also contributes to a greener, more sustainable future by lowering carbon emissions and environmental impact.
- **Scalability:** Machine learning models can be scaled to accommodate various industries, from smart homes to large industrial complexes, making the approach versatile and adaptable to different contexts.
- **Recommendations:** As a result of this project, we recommend the integration of machine learning-based energy management systems in both residential and commercial settings, along with continuous model refinement and data collection.

in essence, this project has showcased the potential of machine learning in transforming the way we understand and manage energy consumption. The application of machine learning in this domain offers not only cost-effective solutions but also a means to support environmental sustainability goals. Its scalability and adaptability make it a powerful tool for organizations and individuals seeking to optimize energy usage.