Phase 2 report

Exploring more advanced time series forecasting techniques like Prophet or deep learning models for improved accuracy in predicting future electricity prices.

Time series forecasting

Time series forecasting is the process of using historical data to predict future values of a time series. Time series data is data that is collected over time, such as daily sales figures, monthly unemployment rates, or hourly temperature readings.

There are many different time series forecasting techniques available, each with its own strengths and weaknesses. Some of the most common techniques include

- Autoregressive integrated moving average (ARIMA) models
- Exponential smoothing models
- Prophet
- Deep learning models

In this project we uses Deep learning Models

Deep learning models:

Deep learning models are a type of machine learning model that can be used for a variety of tasks, including time series forecasting. Deep learning models are typically trained on large amounts of data, and they are able to learn complex patterns in the data. This makes them well-suited for forecasting electricity prices, which are often influenced by a variety of factors, including weather, demand, and supply

1)Collecting and preparing data

Collect a large dataset of historical electricity prices and other relevant factors, such as weather data, demand data, and supply data. Clean and prepare the data for training your model.

2)Choose a deep learning model

There are many different deep learning models that can be used for time series forecasting. Some popular choices include convolutional

- Neural networks (CNNs)
- Recurrent neural networks (RNNs),
- Long short-term memory (LSTM) networks

Choose a model that is appropriate for the characteristics of your data

3)Train the model

Train your model on the historical data. This process can be computationally expensive, so it is important to have access to powerful computing resources.

4)Evaluate the model

Evaluate the performance of your model on a held-out test set. This will help you to assess the accuracy of your model and identify any areas where it can be improved

5)Deploy the model

Once you are satisfied with the performance of your model, you can deploy it to production. This may involve integrating your model into a software application or making it available as a web service

Python libraries for deep learning:

- **TensorFlow:**

TensorFlow is an open-source software library for numerical computation using data flow graphs. It is used for machine learning and deep learning. TensorFlow can be used to build and train a variety of deep learning models, such as neural networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs).

- **PyTorch:**

  PyTorch is an open-source machine learning library based on the Torch library. It is used for building and training deep learning models. PyTorch is known for its flexibility and ease of use.

- **Keras:**

Keras is a high-level Python API for deep learning. It can be used to build and train deep learning models using TensorFlow or PyTorch. Keras is a popular choice for beginners and for prototyping deep learning models.

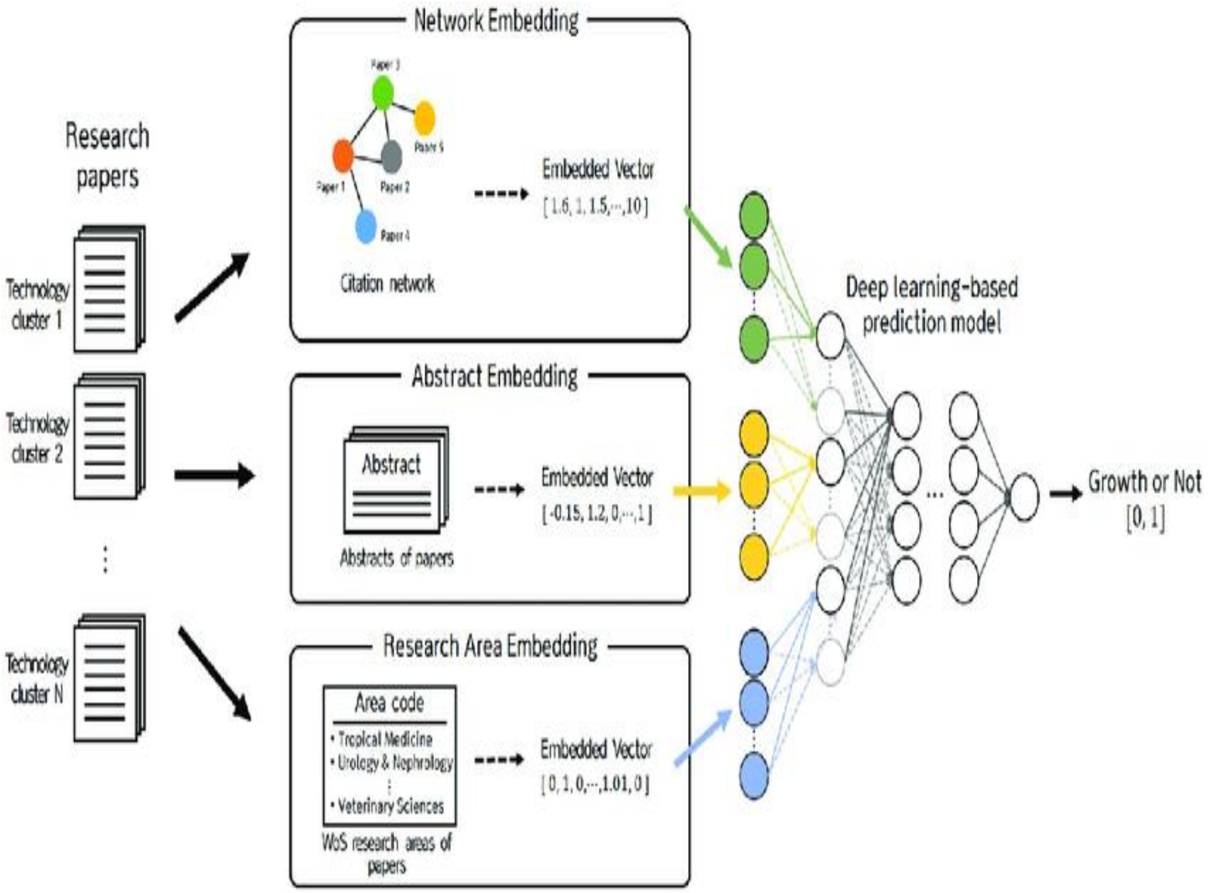
- **MXNet:**

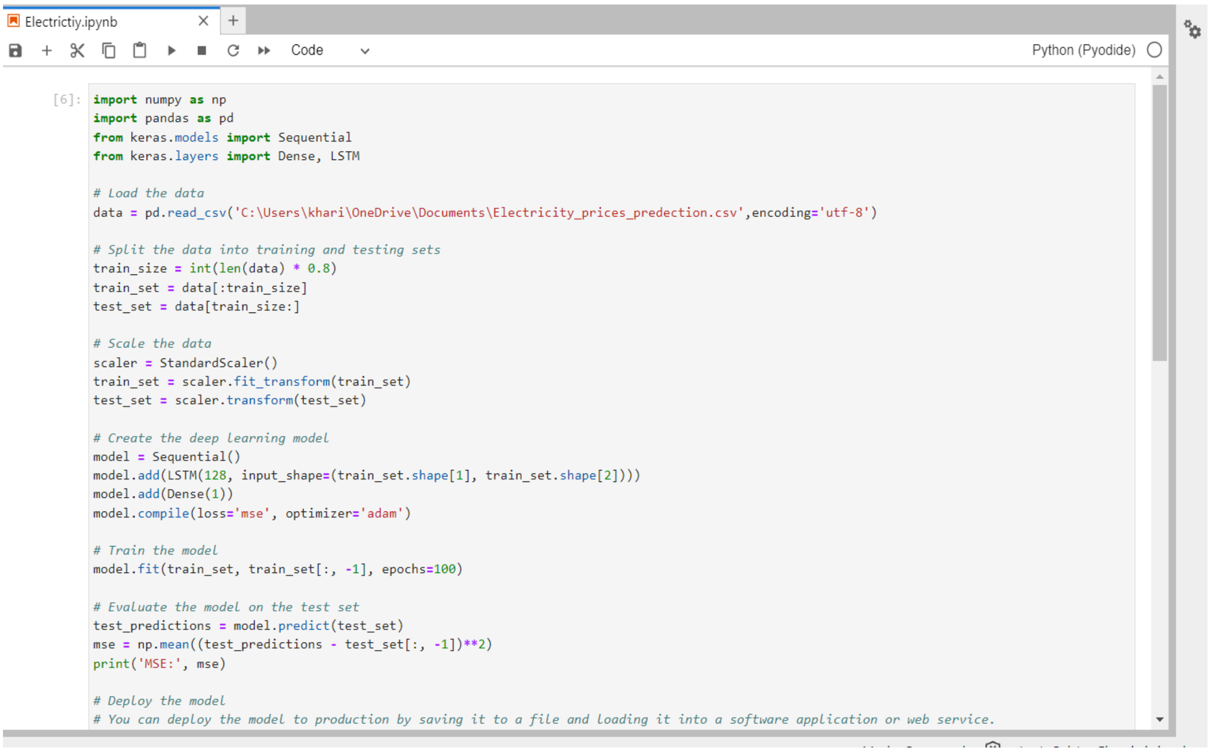
MXNet is an open-source machine learning library that is designed for scalability and efficiency. It can be used to build and train deep learning models on a variety of platforms, including CPUs, GPUs, and distributed systems.

Caffe2: Caffe2 is an open-source machine learning framework that is widely used in academia and industry. It can be used to build and train deep learning models on a variety of platforms, including CPUs, GPUs, and mobile devices.

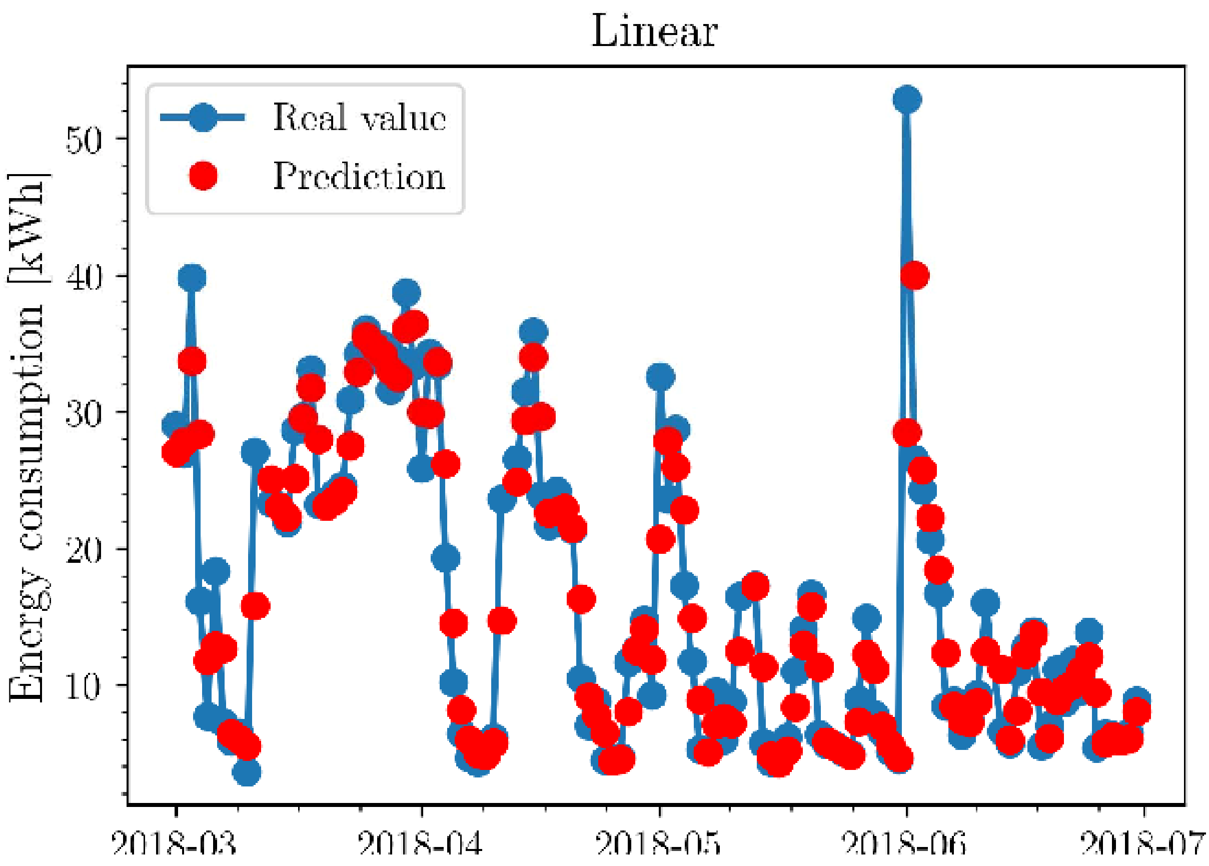
Data set Link:

https://www.kaggle.com/datasets/chakradharmattapalli/electricity-price-prediction


Research papers
Technology cluster 1
Technology cluster 2
⋮
Technology cluster N
Network Embedding
Paper 3
Paper 5
Paper 1
Paper 2
Paper 4
Citation network
Embedded Vector
[1.6, 1, 1.5,···,10]
Abstract Embedding
Abstract
Abstracts of papers
Embedded Vector
[-0.15, 1.2, 0,···,1]
Research Area Embedding
Area code
• Tropical Medicine
• Urology & Nephrology
⋮
• Veterinary Sciences
WoS research areas of papers
Embedded Vector
[0, 1, 0,···,1.01, 0]
Deep learning-based prediction model
Growth or Not
[0, 1]



```
import numpy as np
import pandas as pd
from keras.models import Sequential
from keras.layers import Dense, LSTM

# Load the data
data = pd.read_csv('C:\Users\khari\OneDrive\Documents\Electricity_prices_predection.csv',encoding='utf-8')

# Split the data into training and testing sets
train_size = int(len(data) * 0.8)
train_set = data[:train_size]
test_set = data[train_size:]

# Scale the data
scaler = StandardScaler()
train_set = scaler.fit_transform(train_set)
test_set = scaler.transform(test_set)

# Create the deep learning model
model = Sequential()
model.add(LSTM(128, input_shape=(train_set.shape[1], train_set.shape[2])))
model.add(Dense(1))
model.compile(loss='mse', optimizer='adam')

# Train the model
model.fit(train_set, train_set[:, -1], epochs=100)

# Evaluate the model on the test set
test_predictions = model.predict(test_set)
mse = np.mean((test_predictions - test_set[:, -1])**2)
print('MSE:', mse)

# Deploy the model
# You can deploy the model to production by saving it to a file and loading it into a software application or web service.
```



Conclusion

Our project will contribute to the field of electricity price prediction by developing a deep learning model that is accurate and efficient. Your model could be used by a variety of stakeholders, including electricity producers, consumers, and grid operators, to make better decisions about electricity generation, consumption, and transmission