**Develop neural network-based time series forecasting model**

**EX.No:9**

**DATE:**

**AIM:**

To develop a neural network-based time series forecasting model using a Multi-Layer Perceptron (MLP) to predict future values of electric production based on past historical data.

**ALGORITHM:**

1. Load and preprocess data (datetime conversion, missing value handling, normalization).
2. Transform time series data into supervised learning format using lag features.
3. Split the dataset into training and testing sets.
4. Build and compile a neural network model using Keras.
5. Train the model on the training data.
6. Make predictions using the trained model.
7. Visualize actual vs predicted values using line plots.
8. Interpret results and evaluate model performance.

**CODE:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

df = pd.read\_csv("Electric\_Production.csv")

df['DATE'] = pd.to\_datetime(df['DATE'])

df.set\_index('DATE', inplace=True)

data = df['IPG2211A2N'].values.reshape(-1, 1)

scaler = MinMaxScaler()

scaled\_data = scaler.fit\_transform(data)

def create\_dataset(data, lag=12):

X, y = [], []

for i in range(len(data) - lag):

X.append(data[i:i + lag])

y.append(data[i + lag])

return np.array(X), np.array(y)

lag = 12

X, y = create\_dataset(scaled\_data, lag)

split = int(len(X) \* 0.8)

X\_train, X\_test = X[:split], X[split:]

y\_train, y\_test = y[:split], y[split:]

model = Sequential()

model.add(Dense(64, activation='relu', input\_shape=(lag,)))

model.add(Dense(32, activation='relu'))

model.add(Dense(1))

model.compile(optimizer='adam', loss='mse')

model.fit(X\_train, y\_train, epochs=100, batch\_size=16, verbose=0)

y\_pred = model.predict(X\_test)

y\_pred\_inv = scaler.inverse\_transform(y\_pred)

y\_test\_inv = scaler.inverse\_transform(y\_test)

plt.figure(figsize=(12, 6))

plt.plot(y\_test\_inv, label='Actual')

plt.plot(y\_pred\_inv, label='Predicted', linestyle='--')

plt.title('Neural Network Time Series Forecast')

plt.xlabel('Time')

plt.ylabel('Electric Production')

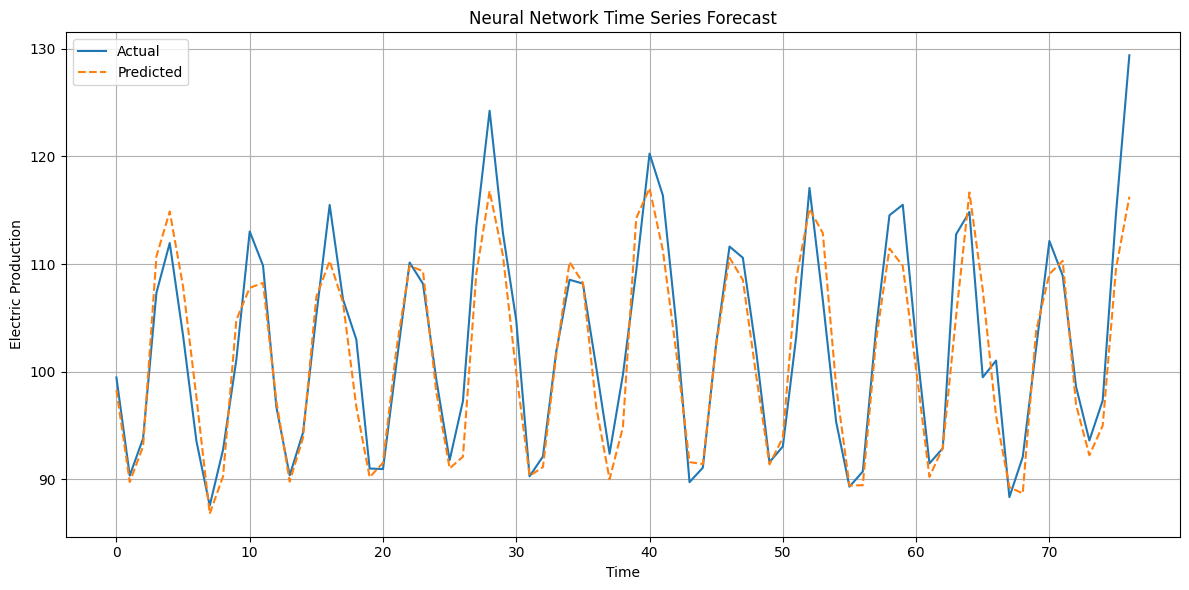
plt.legend()

plt.grid(True)

plt.tight\_layout()

plt.show()

**OUTPUT:**

****

**RESULT:**

Thus the program has been completed and verified successfully.