**Step-by-Step Learning Plan**

**1️⃣ ARIMA (Autoregressive Integrated Moving Average)**

**Objective:** Master time series fundamentals—trend, seasonality, and stationarity.

* **Concepts to Cover:**
  + Understanding ACF (Autocorrelation Function) & PACF (Partial Autocorrelation Function).
  + Stationarity & Differencing (making data stationary).
  + ARIMA (p, d, q) parameter tuning.
* **Practical Implementation:**
  + Dataset: **Kenyan Tomato Price Dataset** (can find similar on Kaggle).
  + Libraries: pandas, statsmodels, matplotlib.
* **Key Skills:**
  + Model fitting and diagnostics.
  + Forecasting and evaluating performance using RMSE or MAPE.

**2️⃣ Prophet (by Meta/Facebook)**

**Objective:** Quick, robust forecasting with minimal tuning.

* **Concepts to Cover:**
  + Handling trends, seasonality, and holidays.
  + Adding custom seasonality (for agriculture cycles in Kenya).
  + Decomposing time series components.
* **Practical Implementation:**
  + Dataset: **Sales Forecasting Dataset for Kenyan Supermarkets.**
  + Libraries: fbprophet or prophet (depending on your Python version).
* **Key Skills:**
  + Handling missing data and outliers.
  + Visualizing forecasts with confidence intervals.

**3️⃣ LSTM (Long Short-Term Memory Networks)**

**Objective:** Capture long-term dependencies in complex time series data.

* **Concepts to Cover:**
  + Understanding RNN vs LSTM (why LSTM handles long sequences better).
  + Data preparation: scaling, windowing (sliding windows), reshaping for LSTM.
  + Model architecture: layers, units, dropout, batch size, epochs.
* **Practical Implementation:**
  + Dataset: **Weather Forecasting Dataset (for agriculture decisions in Kenya).**
  + Libraries: TensorFlow, Keras, NumPy.
* **Key Skills:**
  + Hyperparameter tuning.
  + Evaluating performance (validation loss, overfitting, underfitting).

**4️⃣ XGBoost for Time Series**

**Objective:** Apply gradient boosting for time series forecasting.

* **Concepts to Cover:**
  + Feature engineering: lag features, rolling statistics, date-time features.
  + Difference between time series forecasting and classification/regression.
  + Hyperparameter tuning using GridSearchCV.
* **Practical Implementation:**
  + Dataset: **Energy Consumption Dataset (power usage in Kenya).**
  + Libraries: xgboost, scikit-learn, pandas.
* **Key Skills:**
  + Creating lagged features.
  + Preventing data leakage (train-test split in time series).

**5️⃣ CNN for Time Series (1D Convolution)**

**Objective:** Extract temporal patterns from time series using convolution.

* **Concepts to Cover:**
  + Difference between CNN for images (2D) and time series (1D).
  + Designing 1D convolutional layers for feature extraction.
  + Combining CNN with LSTM for hybrid models (advanced).
* **Practical Implementation:**
  + Dataset: **Network Traffic Dataset (anomaly detection for cybersecurity).**
  + Libraries: TensorFlow, Keras.
* **Key Skills:**
  + Designing filters/kernels for time sequences.
  + Using pooling layers for dimensionality reduction.

**⏱️ Weekly Learning Schedule (4 Weeks)**

| **Week** | **Focus Area** | **Key Deliverable** |
| --- | --- | --- |
| Week 1 | ARIMA + Prophet | Build tomato price forecast & sales dashboard |
| Week 2 | LSTM | Develop weather forecasting model |
| Week 3 | XGBoost for Time Series | Energy consumption forecasting project |
| Week 4 | CNN for Time Series | Anomaly detection in network traffic |