# Flowchart - Hydrological Modeling System

## Step 1: Load Configuration

* User loads XML/YAML config file
* ConfigParser loads and validates:
* - Basin name
* - Date range
* - Grid type (UGRID / REGULAR)
* - Model type (PINN / XPINN / Curriculum Learning)

## Step 2: Load Rainfall & Discharge

* RainLoader loads NetCDF rainfall files (2010–2022)
* Rain interpolated to UGRID via kriging
* DischargeProcessor loads CSV discharge data
* Optional: Resample rainfall/discharge (e.g., 30min)

## Step 3: Merge Data

* SpatialMerger joins:
* - Rain (interpolated)
* - Discharge (per station)
* - Static spatial features (DEM, SLOPE, etc.)
* Output: unified DataFrame with lat/lon/time/rain/Q/features

## Step 4: Feature Engineering

* FeatureEngineer performs:
* - Normalization (e.g., Z-score)
* - Compute TAU (normalized time)
* - Optional: Spatial Embedding
* - Feature Selection (manual or auto)
* Output: clean DataFrame ready for modeling

## Step 5: Sequence Building

* DatasetBuilder slices sequences of length L (e.g., 6 steps)
* Split by event intensity or region (Curriculum Learning)
* Create train/test loaders

## Step 6: Model Training

* ModelTrainer loads model type (e.g., LSTM + Physics)
* Trains on sequences with:
* - Weighted loss on discharge
* - Physics residual loss (e.g., Manning)
* Save trained model

## Step 7: Model Evaluation

* Evaluator:
* - Computes metrics (MSE, etc.)
* - Plots hydrographs (per station)
* - Generates rainfall-Q animation video

## Step 8: Staging Phase (Optional)

* StagingRunner (optional):
* - Compare architectures (LSTM, FCN)
* - Try alternative physics equations
* - Retrain on new events (e.g., 2024 data)

## Step 9: Future Prediction

* New rainfall forecast (or ensemble) is fed in
* Trained model predicts future discharge

## 🧭 מטרת המסמך:

המסמך מתאר את **זרימת העבודה (Flowchart)** של כל מערכת החיזוי שאתה בונה — משלב קלט הקונפיגורציה, דרך טעינת הדאטה, ועד הפקת תחזית עתידית.

הוא בנוי **כשלבים ברורים, אחד אחרי השני**, ומאפשר להבין כיצד כל רכיב בתהליך משתלב בתמונה הגדולה.

## 🟢 שלב 1: טעינת קונפיגורציה (Configuration)

* המשתמש בוחר קובץ תצורה (XML או YAML)
* הקובץ מגדיר:
  + איזה אגן נבחר
  + תאריכים לאימון
  + סוג הגריד (UGRID או רגולרי)
  + סוג המודל: PINN / XPINN / CURRICULUM

## 🌧️ שלב 2: טעינת נתוני גשם וספיקה

* נטענים קבצי NetCDF של גשם (גשם מרחבי ב־10 דקות)
* מבוצעת אינטרפולציה (לרוב קריגינג) לגריד שלך (UGRID)
* נטענים קובצי CSV של ספיקה
* ניתן לבצע resample ל־30 דקות או שעה אם צריך

## 🧩 שלב 3: מיזוג הדאטה

* מחברים הכל ל־DataFrame אחד אחיד:
  + גשם ← אחרי אינטרפולציה
  + ספיקה ← לפי תחנות
  + מידע מרחבי סטטי ← DEM, שיפועים, חספוס

## 🛠️ שלב 4: הנדסת פיצ'רים

* נרמול של ערכים (Z-score)
* חישוב זמן מנורמל (TAU)
* Spatial Embedding (אופציונלי)
* בחירת פיצ'רים — לפי ניתוח או ידנית

## 🔄 שלב 5: בניית רצפים לאימון

* יוצרים רצפים באורך קבוע (למשל 6 צעדים)
* מחלקים לפי עוצמת אירועים או לפי אזורים – לצורך **למידה הדרגתית**
* יוצרים train/test loader

## 🧠 שלב 6: אימון המודל

* נטען המודל המתאים (LSTM, FCN וכו')
* מתבצע אימון:
  + עם איבר loss רגיל על discharge
  + עם איבר loss פיזיקלי (למשל משוואת Manning)
* נשמר checkpoint של המודל

## 📊 שלב 7: הערכת המודל

* ניתוח ביצועים:
  + חישוב מדדים (MSE וכד')
  + ציור הידרוגרפים
  + הפקת סרטון: תנועת גשם + תגובת תחנות

## 🧪 שלב 8: ניסויים (Staging)

* בשלב זה מבצעים שיפורים:
  + השוואה בין סוגי מודלים
  + בדיקת משוואות פיזיקליות אחרות
  + התאמה לדאטה חדש (למשל 2024)

## 📈 שלב 9: תחזית עתידית

* מקבלים קלט חדש של גשם (חזוי או אנצמבל)
* מפעילים את המודל המאומן
* תחזית ספיקה לעתיד באגן

## ✅ למה זה חשוב:

* מאפשר הבנה שיטתית של כל המערכת
* בסיס לכתיבת הקוד ולמימוש בפועל
* עוזר להפריד בין ליבה (core) לבין ניסויים (staging)
* מאפשר תיעוד מסודר לצוות או לגיטהאב