Introduction:

Creating a comprehensive flood monitoring system with sensor data, threshold alerts, database storage, predictive modeling, automatic emergency alarms, and alert mechanisms is a complex task. Below is a simplified example using Pythonscript.

Program:

```
Import random

Import time

Import sqlite3

From collections import deque


# Simulated water level sensor (replace with real sensor data)

Def simulate_water_level_sensor():

   Return random.uniform(0.0, 10.0)


# Function to save data to a database

Def save_to_database(location, water_level):

   Conn = sqlite3.connect("flood_data.db")

   Cursor = conn.cursor()

   Cursor.execute("INSERT INTO water_level_data (location, water_level, timestamp) VALUES (?, ?, DATETIME('now'))", (location, water_level))

   Conn.commit()

   Conn.close()


# Predictive modeling parameters

Num_readings_for_prediction = 5  # Number of recent readings for prediction

Rising_threshold = 0.1  # Threshold for rising water levels


# Function to predict rising water levels

Def predict_rising_water_level(readings):
```

```
    If len(readings) < num_readings_for_prediction:
        Return False
    Return all(readings[i] < readings[I + 1] for I in range(len(readings) – 1))

# Function to trigger an emergency alarm
Def trigger_emergency_alarm(location, water_level):
    Print(f”EMERGENCY ALARM: Flood at {location} is predicted with a water level of {water_level} meters.”)

# Main monitoring loop
Def main():
    Location = “Your Location”  # Replace with your location name

    Readings = deque(maxlen=num_readings_for_prediction)  # Keep track of recent water level readings

    While True:
        Water_level = simulate_water_level_sensor()
        Print(f”Current water level at {location}: {water_level} meters”)

        # Save data to the database
        Save_to_database(location, water_level)

        Readings.append(water_level)

        If water_level > 5.0:  # Adjust the threshold as needed for initial alert
            Print(f”Flood Alert at {location}! Water level: {water_level} meters”)

        If predict_rising_water_level(readings):
            Trigger_emergency_alarm(location, water_level)
```

```
    Time.sleep(60)  # Simulate reading data every minute


If __name__ == “__main__”:
    Conn = sqlite3.connect(“flood_data.db”)
    Cursor = conn.cursor()
    Cursor.execute(‘’’CREATE TABLE IF NOT EXISTS water_level_data (location TEXT, water_level REAL, timestamp DATETIME)’’’)
    Conn.commit()
    Conn.close()


    Main()
```

Conclusion:

In this script, we store water level data in a database, use predictive modeling to detect rising water levels, and trigger an emergency alarm when a consistent rise is detected