Sinhgad Institutes

# Sinhgad College of Engineering, Pune

## Department of E&TC

B.E. E&TC - (Project)

**Group No: 79**

# Landslide Monitoring And Prediction System Using Machine Learning

(Sponsered By:- TDL TechSphere)


**By:**

| | |
|---|---|
| ***Aditi Sarraf*** | ***404A009*** |
| ***Parth Pol*** | ***404C047*** |
| ***Swaroop Sawale*** | ***404D013*** |

**Guide: Dr S.S.Patil(Internal)**


**2024-25, Sem: VII**

Sinhgad Institutes

# Contents

# Introduction

- The system is designed to continuously monitor soil moisture, temperature, humidity, and movement using a variety of sensors and provides it for real time monitoring .
- This data is processed by a microcontroller .To address this, we propose a machine learning algorithm that uses sensor data as input datasets to predict potential landslides and activate early-warning systems.
- Additionally, the system features a multi-stage alarm to provide timely warnings for potential landslides and floods. Provides an early warning system for landslides and floods, reducing potential damage and enhancing safety.

# Aim & Objectives

Sinhgad Institutes

❖ **Aim:**

- Develop a system to monitor soil conditions, including moisture, temperature, humidity, and movement, to predict potential landslides and provide early warnings.

❖ **Objectives:**

1. **Real-time Monitoring**: Continuously monitor soil moisture, temperature, humidity, and vibrations using various sensors.
2. **Multi-stage Alarm System**: Implement an alarm system that triggers at different stages based on the severity of detected conditions, indicating potential landslide risks.
3. **Data Processing and Prediction**: Use machine learning algorithms to process sensor data and predict landslides, providing timely alerts.
4. **Automated Preventive Measures**: Facilitate preventive actions to reduce disaster impacts by providing warnings and potentially activating safety measures.

# Literature Survey


1. **"Deep Learning-Based Landslide Susceptibility Mapping"** – Zhu, Q., Xu, X., Pan, B., Chen, F., & Zhao, Q. (2022), *Engineering Geology*, Elsevier.
2. **"Landslide Detection Using Remote Sensing and Machine Learning"** – Youssef, N., Pourghasemi, M., & Demirci, D. (2021), *Remote Sensing*, MDPI.
3. **"A Comparative Study of Machine Learning Algorithms for Landslide Susceptibility Mapping"** – Hong, X., Liu, Y., Wu, J., & Li, T. (2020), *Landslides*, Springer.
4. **"Early Warning of Landslides Using IoT and Edge Computing"** – Tan, J., Wang, C., & Zhang, L. (2021), *IEEE Internet of Things Journal*.
5. **"A Deep Learning Framework for Real-Time Landslide Detection Using Seismic Data"** – Zhao, F., Liu, H., & Chen, K. (2022), *Computers & Geosciences*, Elsevier.
6. **"Multi-Criteria Decision Analysis for Landslide Hazard Assessment Using GIS"** – Kumar, R., Singh, S., & Mehta, P. (2019), *Geocarto International*, Taylor & Francis.

Sinhgad Institutes

# Components Requirements

- ESP32
- Soil Moisture Sensor
- Accelerometer MPU6050
- Temperature sensor DHT11
- Vibration Sensor
- Ultrasonic Sensor
- Buzzer

# Hardware Requirements

**Hardware Requirements:**

- **Sensors:**
  - Soil Moisture Sensor: Measures water content in the soil.
  - Temperature and Humidity Sensor (DHT11) : Monitors temperature and humidity
  - Accelerometer (MPU6050): To get x, y, z acceleration helps detect movements or changes in the angle of the slope, which could indicate potential landslides.
  - Vibration Sensor (SW-420) : To detect sudden movements
- **Microcontroller/Processor:**
  - ESP32 : For reading sensor data, processing, controlling alarms and sending data to cloud platform ThingSpeak.
- **Output Devices:**
  - LEDs: Indicate different levels of alerts.
  - Buzzer: Sounds alarm when thresholds are exceeded.
  - Notification : ML model processes the data from sensors and makes prediction.

Sinhgad Institutes

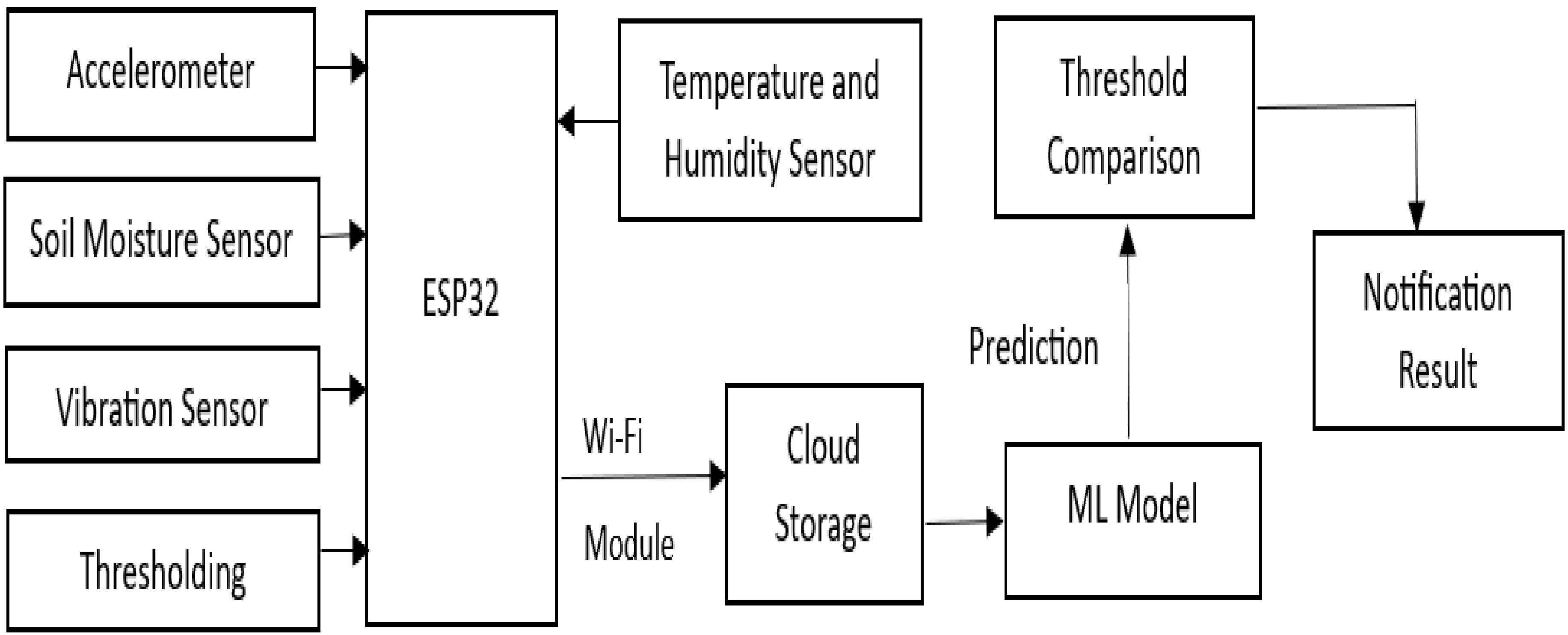
# Software Requirements

**Software Requirements:**

- **Libraries for ESP32 & Cloud :**
  - ESP32WiFi.h : Connects ESP32 to Wi-Fi..
  - WiFiClient.h: Enables TCP client for cloud communication
  - ThingSpeak.h : Sends sensor data to ThingSpeak cloud.
- **Sensor Libraries :**
  - DHT.h + Adafruit Unified Sensor : For DHT11 (temp & humidity).
  - Wire.h : I2C communication for MPU6050.
  - Adafruit_MPU6050.h: Reads accelerometer data.
- **Arduino IDE:**
  - Used to write, compile & upload code to ESP32.
  - Enables sensor interfacing via GPIO/analog pins.
  - Sends data every 15 – 20 secs to ThingSpeak using ThingSpeak.writeFields().

- **ML Data Processing**
  - NumPy : For numerical ops.
  - Pandas : For data cleaning & analysis.
  - Matplotlib : For data visualization

# Block Diagram



# METHODOLOGY

**Implementation Steps :-**

- **1.Sensor Integration:**
  - Connect all sensors to the microcontroller.
  - Ensure proper calibration of sensors to get accurate readings.
- **2. Microcontroller Programming:**
  - Write code to read data from sensors.
  - Implement logic to check if readings exceed thresholds.
  - Control the buzzer for alarm signaling.
- **3.ML model:** The model is trained on historical data from similar landslide-prone areas. The dataset contains features such as soil moisture, temperature, Rainfall, slope angle and soil type. If values exceed the threshold in the ML model, the buzzer will be activated.
- **4. System Alert:** When the machine learning model detects that the conditions for a landslide are met (based on thresholds of moisture, temperature , soil type), a buzzer will be triggered.

Calculation:- For movements in give direction

Accelerometer: for prone areas ,steep slopes , loose soil , ground movement.

Also detect tilt or inclination:-

Accleration in y-axis $a_y = 0.4g$

Accleration in x–axis $a_x = 0.3g$

Accleration in z–axis $az = 0.8g$

Total Acceleration A= $\sqrt{(0.3)^2 + (0.4)^2 + (0.8)^2} = 0.943g$

$1g = 9.8\, m/s^2$

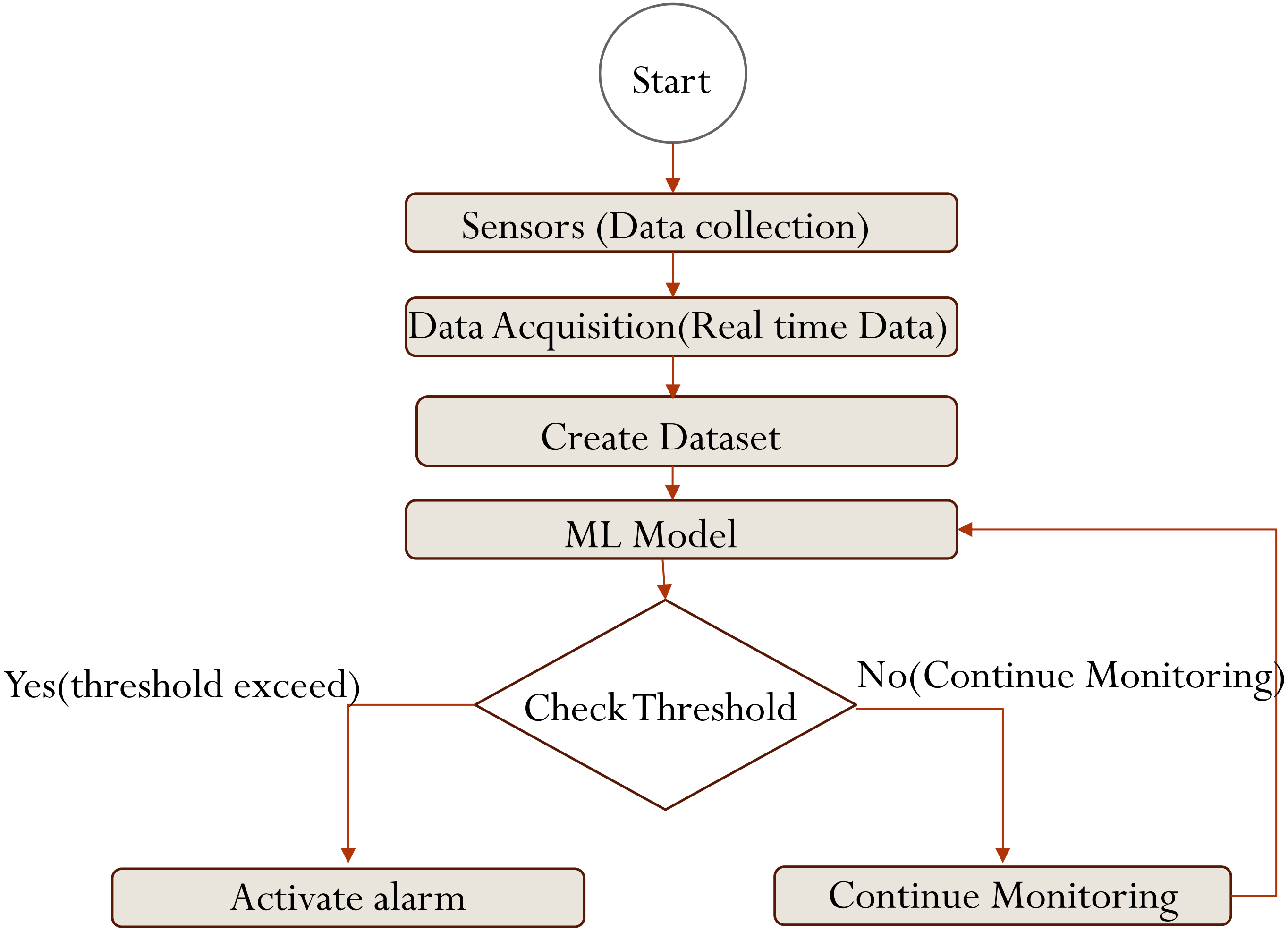
$=0.943 \times 9.8 = 9.2\, m/s^2$

Tilt (using Z-Axis):-

$\theta = \arctan\left[\frac{a_x}{a_z}\right] = \theta = \arctan\left[\frac{0.3}{0.8}\right] = 20.56^0$

Sinhgad Institutes

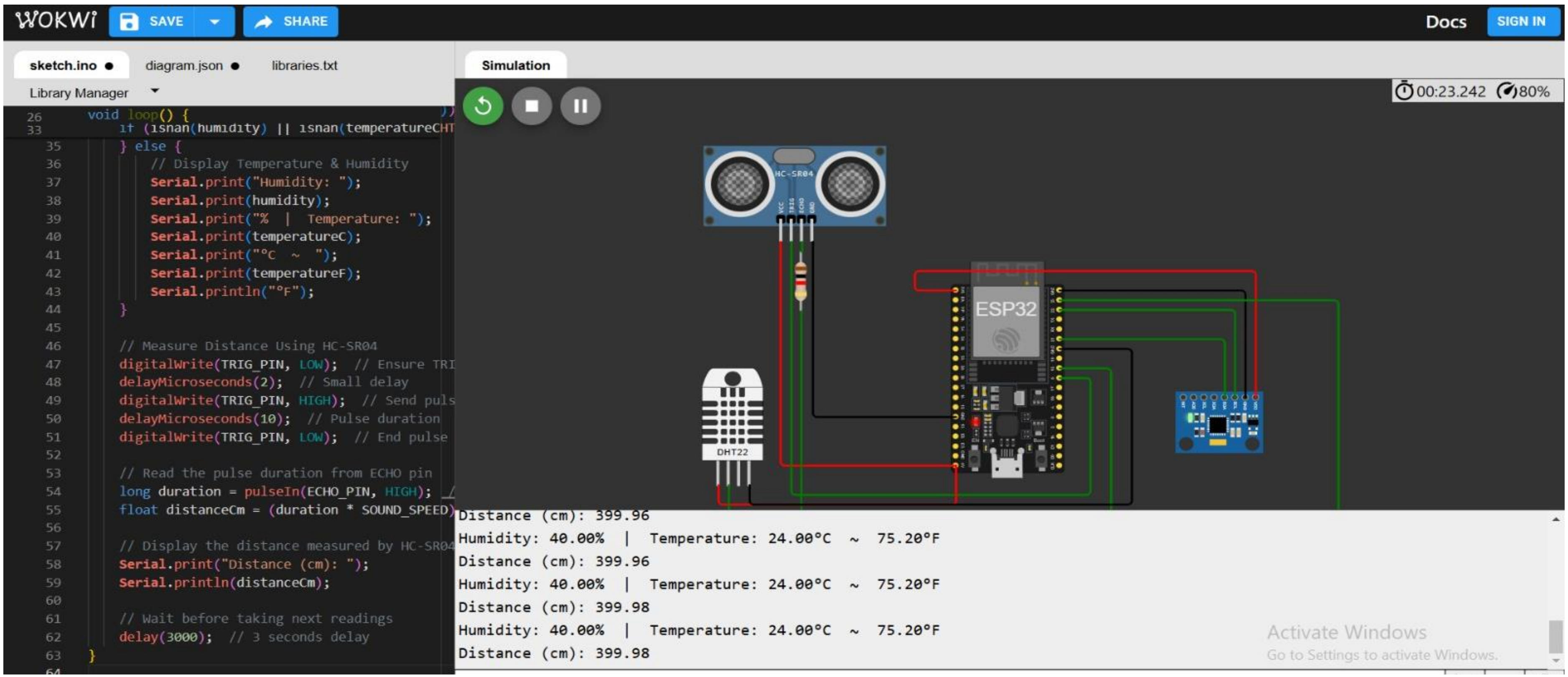
# Algorithm(Flowchart)



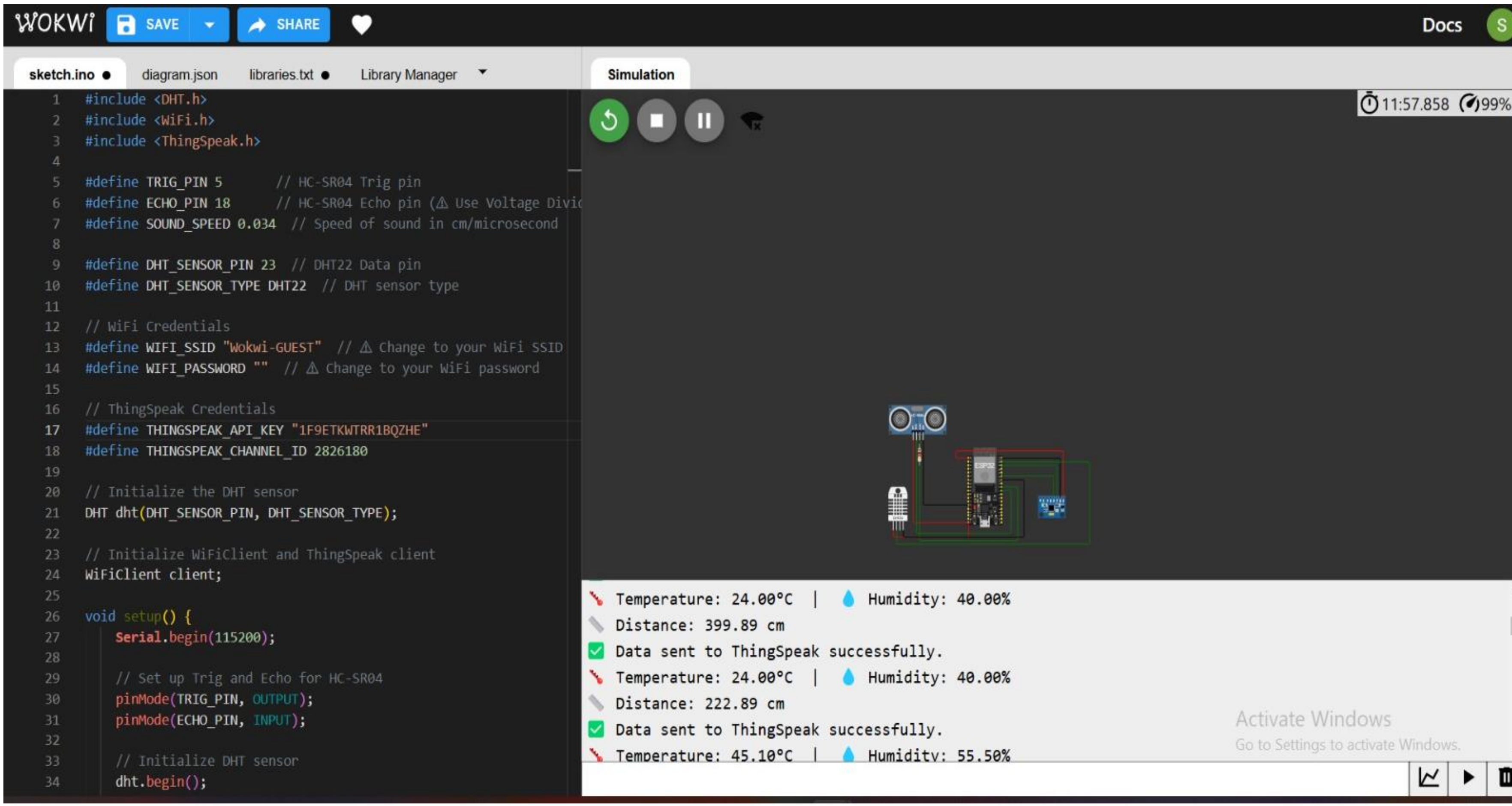
Sinhgad Institutes

# Results

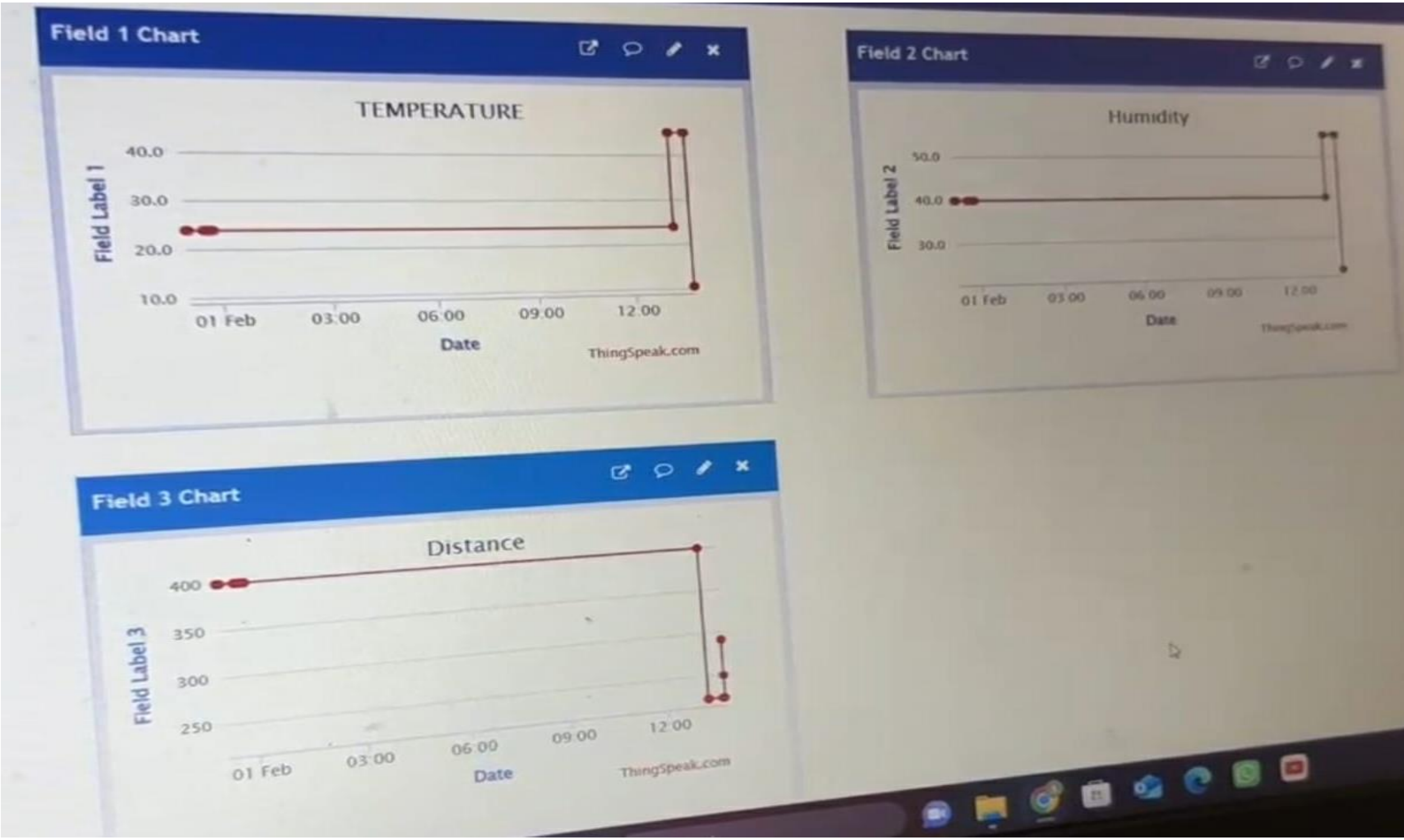
## Simulations and IOT Integration



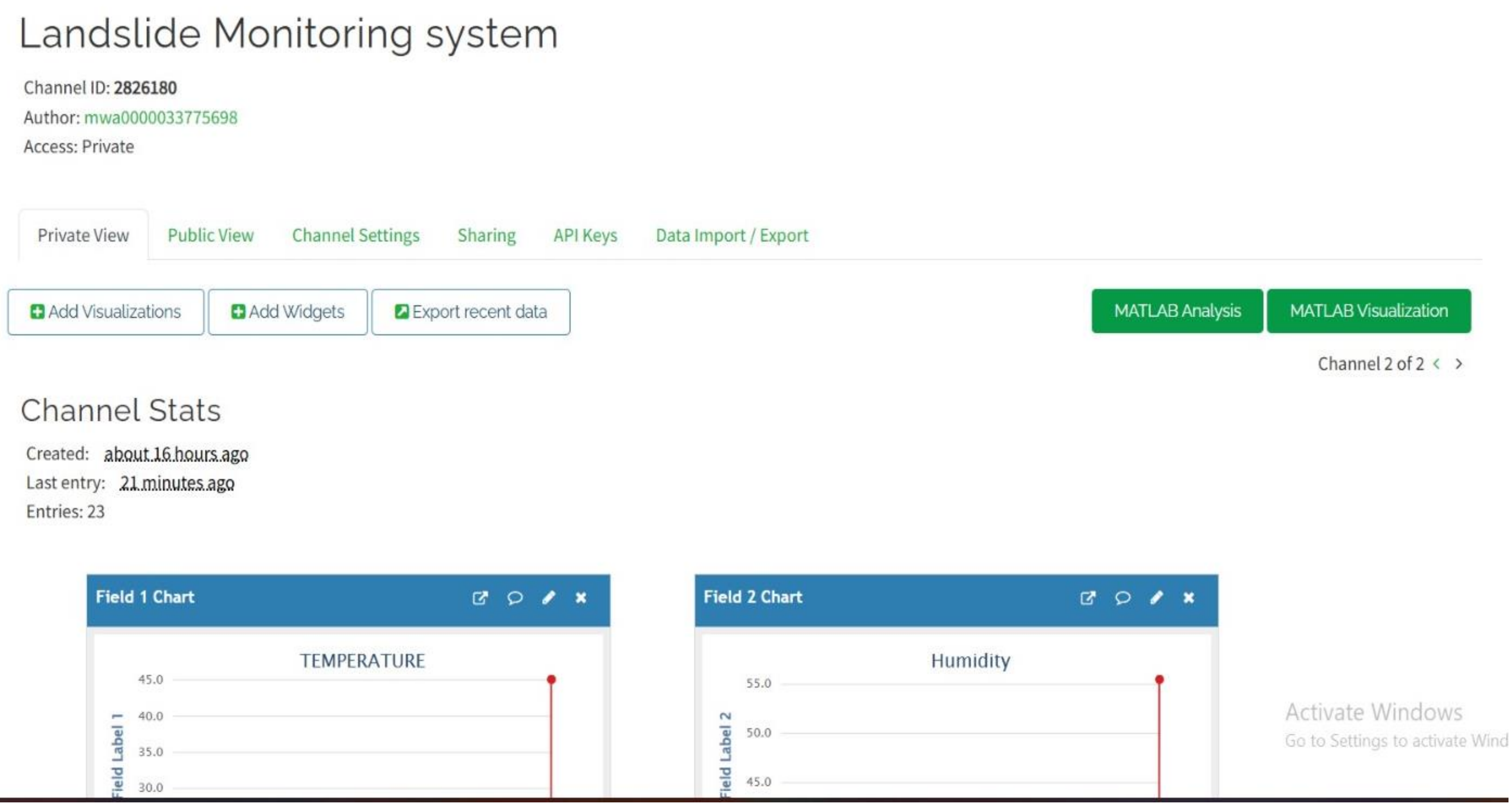
Sinhgad Institutes

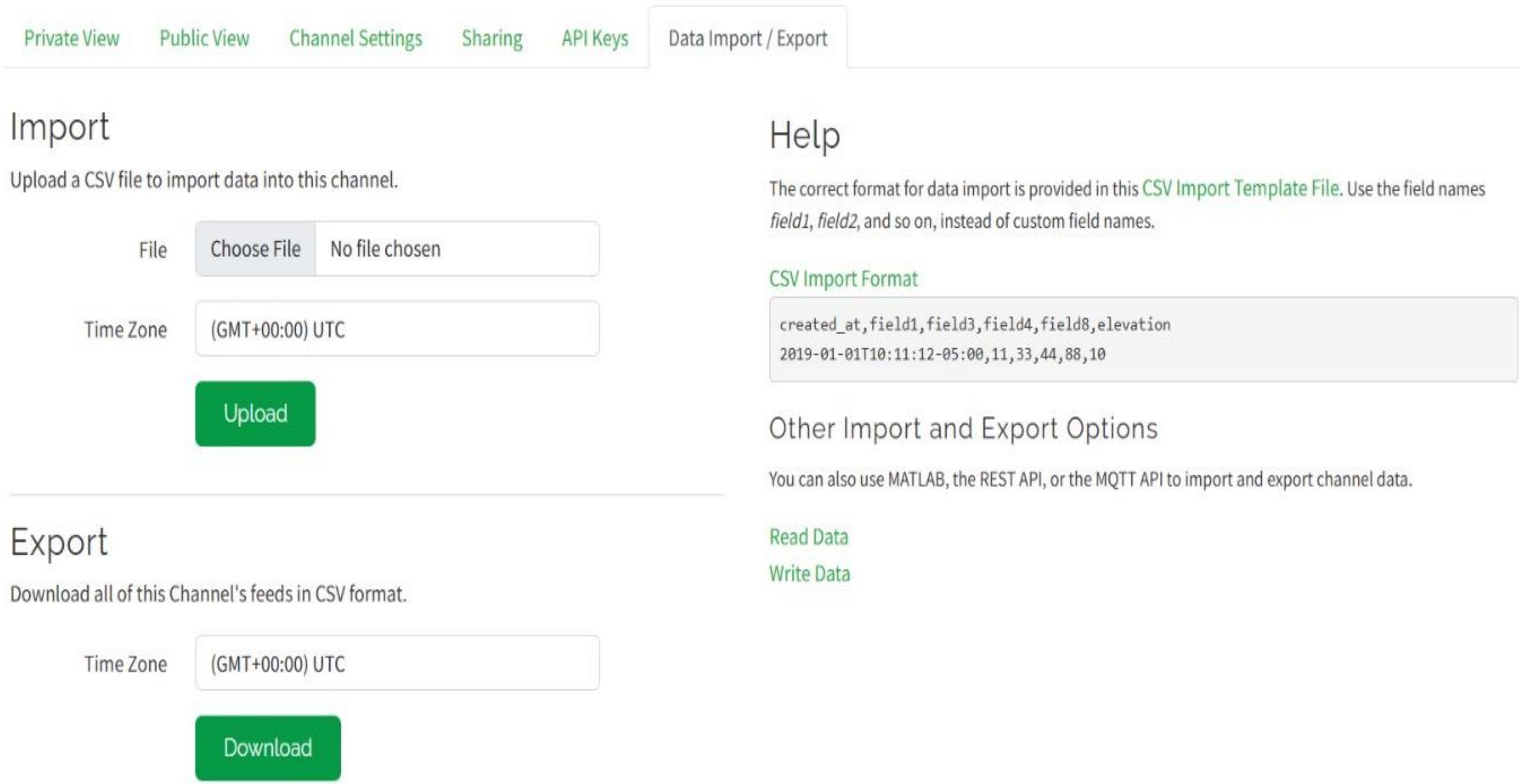


Sinhgad Institutes

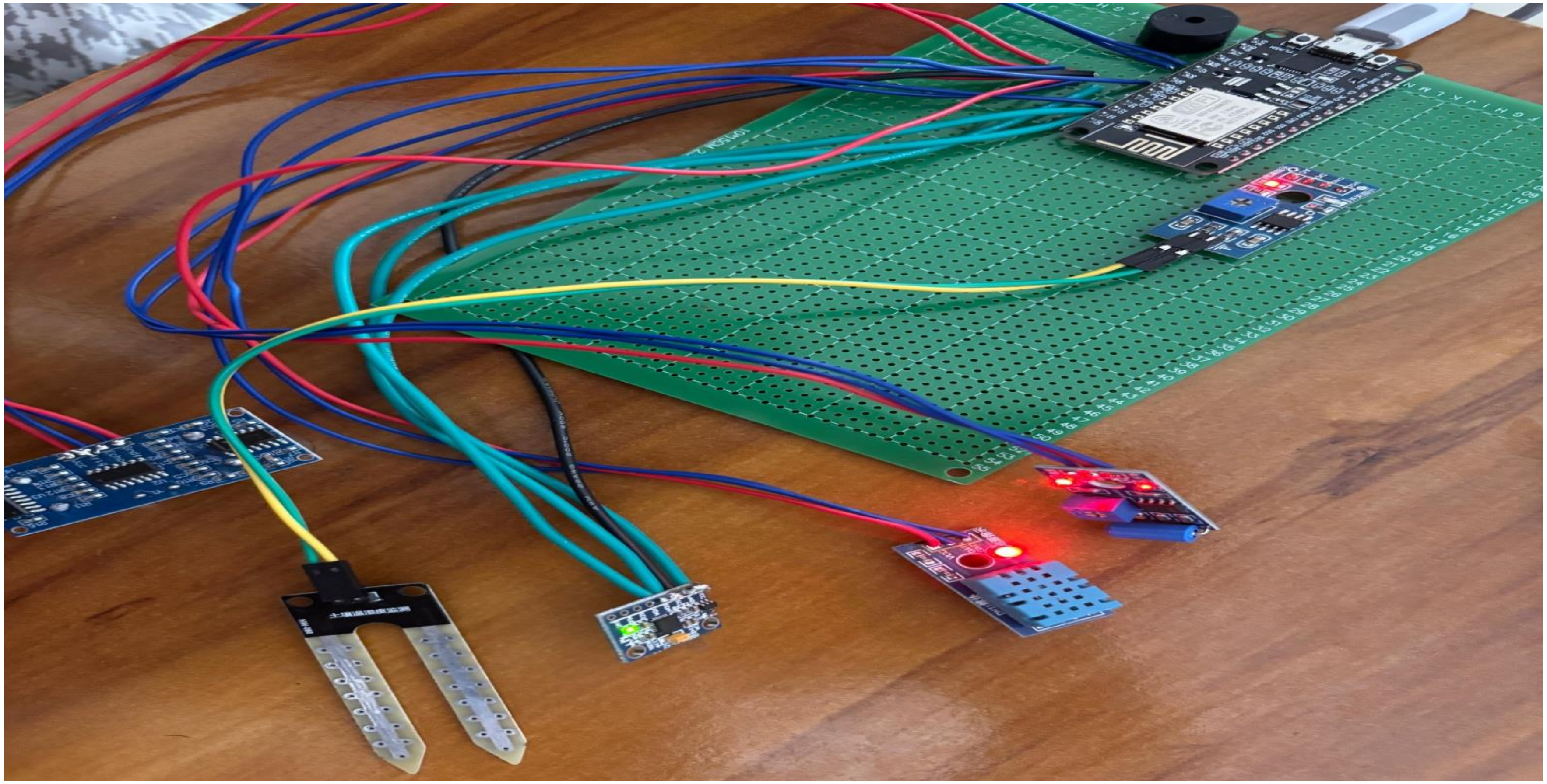


Sinhgad Institutes




Private View
Public View
Channel Settings
Sharing
API Keys
Data Import / Export
Import
Upload a CSV file to import data into this channel.
File
Choose File
No file chosen
Time Zone
(GMT+00:00) UTC
Upload
Export
Download all of this Channel's feeds in CSV format.
Time Zone
(GMT+00:00) UTC
Download
Help
The correct format for data import is provided in this CSV Import Template File. Use the field names field1, field2, and so on, instead of custom field names.
CSV Import Format
created_at,field1,field3,field4,field8,elevation
2019-01-01T10:11:12-05:00,11,33,44,88,10
Other Import and Export Options
You can also use MATLAB, the REST API, or the MQTT API to import and export channel data.
Read Data
Write Data

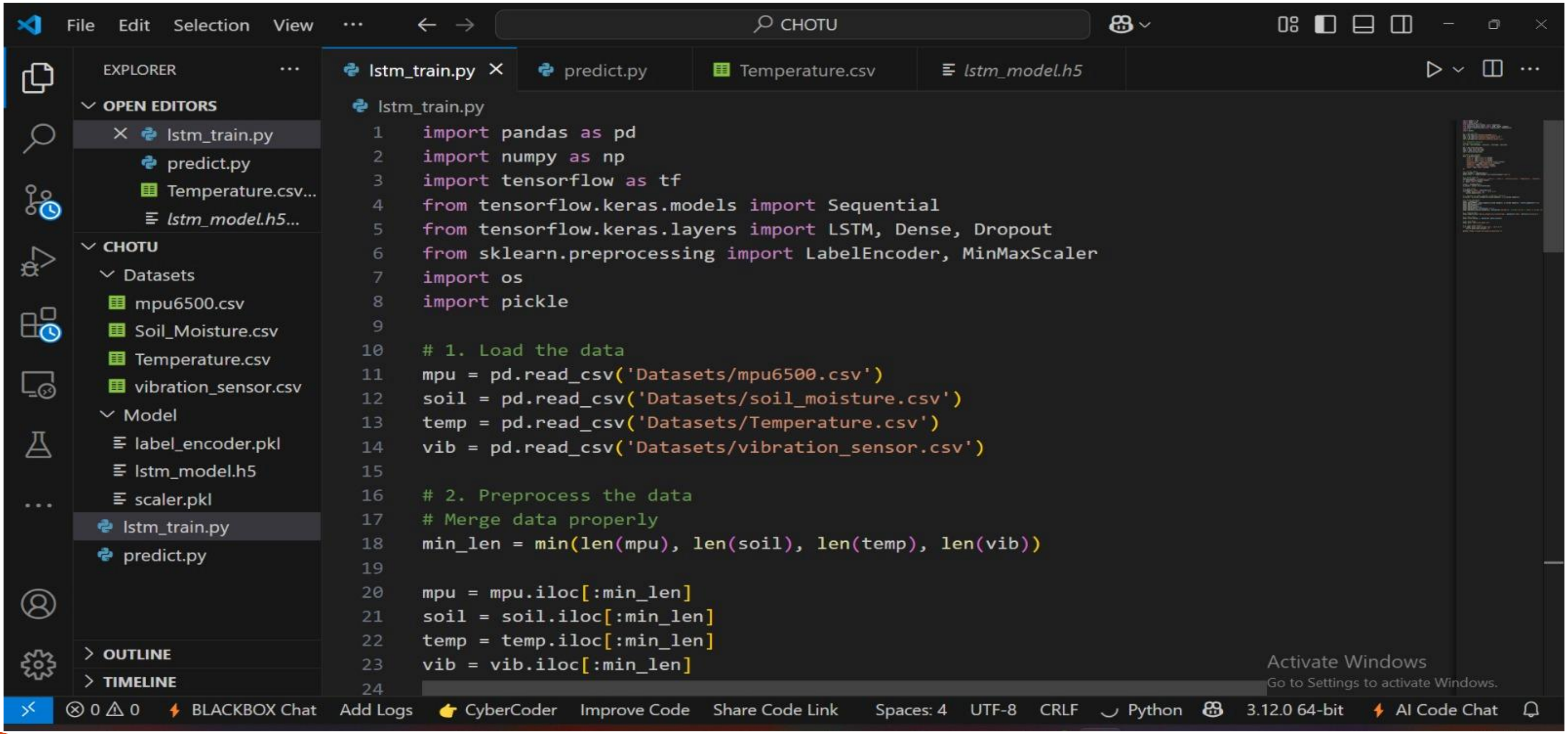
# Hardware Integration



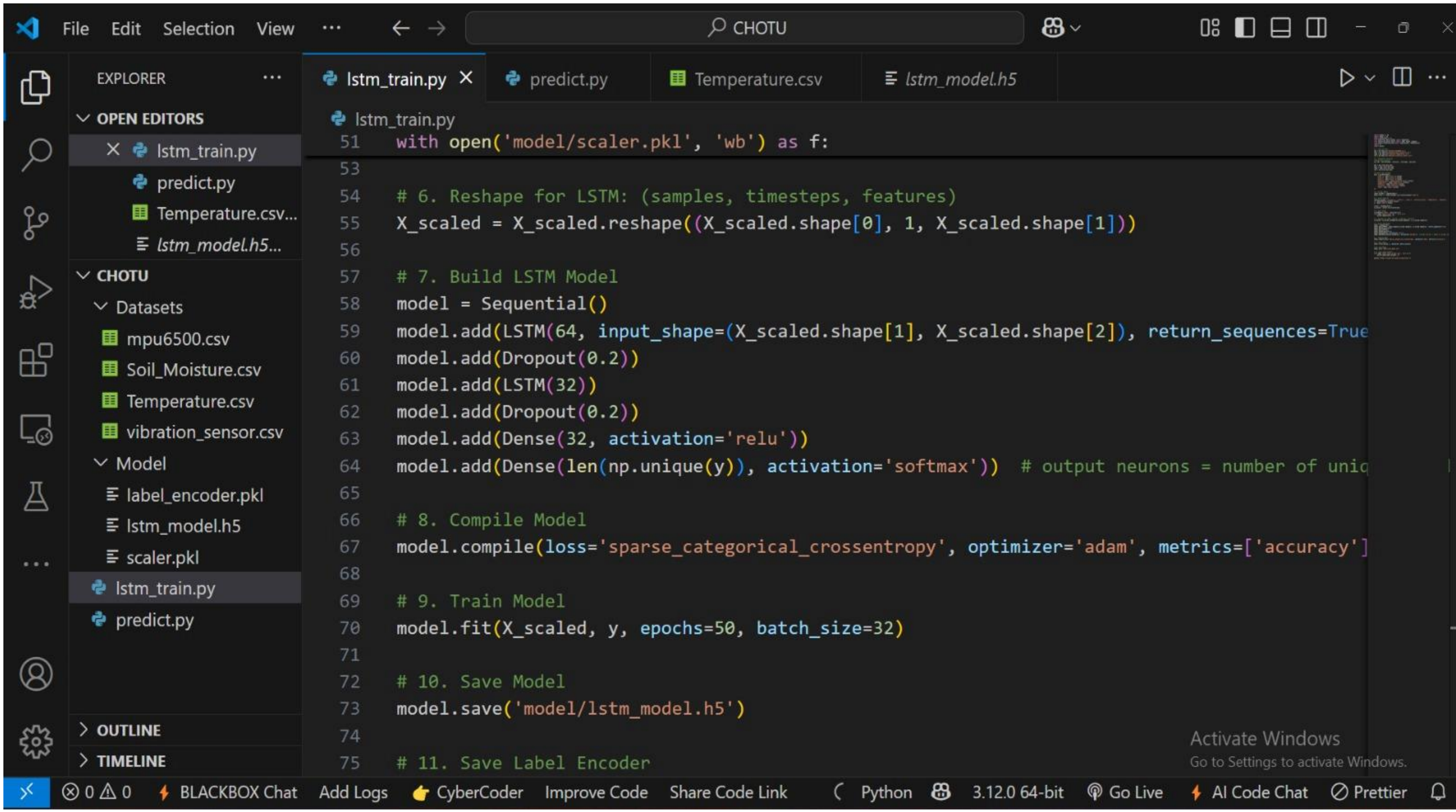
Sinhgad Institutes

# ML Model :

## Training Script :



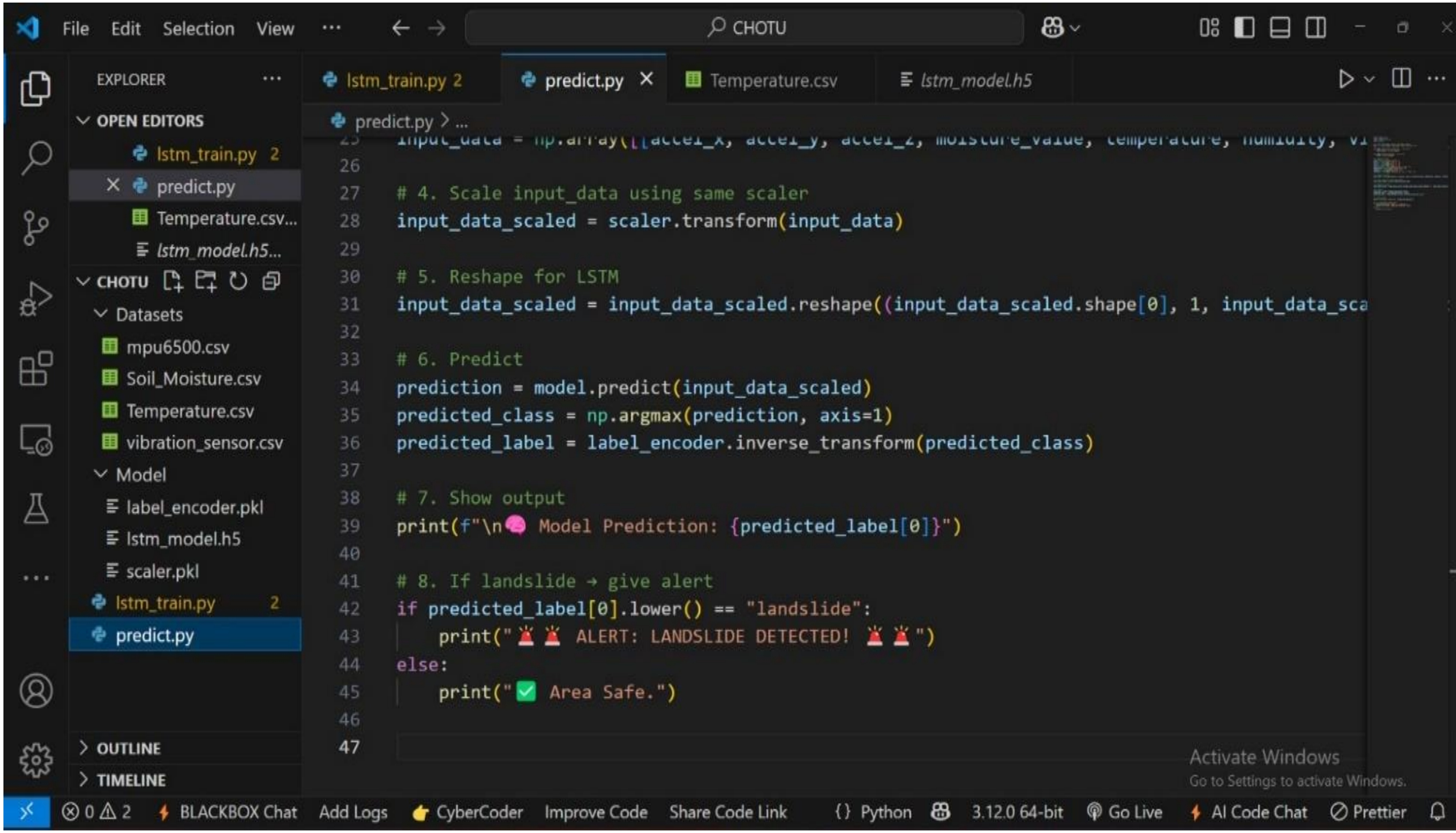

```python
import pandas as pd
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
from sklearn.preprocessing import LabelEncoder, MinMaxScaler
import os
import pickle

# 1. Load the data
mpu = pd.read_csv('Datasets/mpu6500.csv')
soil = pd.read_csv('Datasets/soil_moisture.csv')
temp = pd.read_csv('Datasets/Temperature.csv')
vib = pd.read_csv('Datasets/vibration_sensor.csv')

# 2. Preprocess the data
# Merge data properly
min_len = min(len(mpu), len(soil), len(temp), len(vib))

mpu = mpu.iloc[:min_len]
soil = soil.iloc[:min_len]
temp = temp.iloc[:min_len]
vib = vib.iloc[:min_len]
```


File Edit Selection View
CHOTU
EXPLORER
OPEN EDITORS
lstm_train.py
predict.py
Temperature.csv...
lstm_model.h5...
CHOTU
Datasets
mpu6500.csv
Soil_Moisture.csv
Temperature.csv
vibration_sensor.csv
Model
label_encoder.pkl
lstm_model.h5
scaler.pkl
lstm_train.py
predict.py
OUTLINE
TIMELINE
lstm_train.py
predict.py
Temperature.csv
lstm_model.h5
with open('model/scaler.pkl', 'wb') as f:
# 6. Reshape for LSTM: (samples, timesteps, features)
X_scaled = X_scaled.reshape((X_scaled.shape[0], 1, X_scaled.shape[1]))
# 7. Build LSTM Model
model = Sequential()
model.add(LSTM(64, input_shape=(X_scaled.shape[1], X_scaled.shape[2]), return_sequences=True
model.add(Dropout(0.2))
model.add(LSTM(32))
model.add(Dropout(0.2))
model.add(Dense(32, activation='relu'))
model.add(Dense(len(np.unique(y)), activation='softmax')) # output neurons = number of uniq
# 8. Compile Model
model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy']
# 9. Train Model
model.fit(X_scaled, y, epochs=50, batch_size=32)
# 10. Save Model
model.save('model/lstm_model.h5')
# 11. Save Label Encoder
Activate Windows
Go to Settings to activate Windows.
0 0 BLACKBOX Chat Add Logs CyberCoder Improve Code Share Code Link Python 3.12.0 64-bit Go Live AI Code Chat Prettier

Sinhgad Institutes

# Prediction Script

```python
import numpy as np
import tensorflow as tf
import pickle

# 1. Load saved model, scaler, and label encoder
model = tf.keras.models.load_model('model/lstm_model.h5')

with open('model/label_encoder.pkl', 'rb') as f:
    label_encoder = pickle.load(f)

with open('model/scaler.pkl', 'rb') as f:
    scaler = pickle.load(f)

# 2. Take user input
print("Enter the sensor values:")
accel_x = float(input("Accel X: "))
accel_y = float(input("Accel Y: "))
accel_z = float(input("Accel Z: "))
moisture_value = float(input("Moisture Value: "))
temperature = float(input("Temperature: "))
humidity = float(input("Humidity: "))
vibration = int(input("Vibration (0 = No, 1 = Yes): "))

# 3. Prepare the data
```


File Edit Selection View
CHOTU
EXPLORER
OPEN EDITORS
lstm_train.py 2
predict.py
Temperature.csv...
lstm_model.h5...
CHOTU
Datasets
mpu6500.csv
Soil_Moisture.csv
Temperature.csv
vibration_sensor.csv
Model
label_encoder.pkl
lstm_model.h5
scaler.pkl
lstm_train.py 2
predict.py
OUTLINE
TIMELINE
lstm_train.py 2
predict.py
Temperature.csv
lstm_model.h5
predict.py > ...
# 4. Scale input_data using same scaler
input_data_scaled = scaler.transform(input_data)
# 5. Reshape for LSTM
input_data_scaled = input_data_scaled.reshape((input_data_scaled.shape[0], 1, input_data_sca
# 6. Predict
prediction = model.predict(input_data_scaled)
predicted_class = np.argmax(prediction, axis=1)
predicted_label = label_encoder.inverse_transform(predicted_class)
# 7. Show output
print(f"\n🧠 Model Prediction: {predicted_label[0]}")
# 8. If landslide → give alert
if predicted_label[0].lower() == "landslide":
print("🚨🚨 ALERT: LANDSLIDE DETECTED! 🚨🚨")
else:
print("✅ Area Safe.")
Activate Windows
Go to Settings to activate Windows.
0 2 BLACKBOX Chat Add Logs CyberCoder Improve Code Share Code Link
{} Python 3.12.0 64-bit Go Live AI Code Chat Prettier

# Output :

```
Epoch 47/50
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step - accuracy: 0.9895 - loss: 0.0216
Epoch 48/50
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - accuracy: 1.0000 - loss: 0.0089
Epoch 49/50
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step - accuracy: 1.0000 - loss: 0.0089
Epoch 50/50
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step - accuracy: 0.9817 - loss: 0.0269
3/3 ━━━━━━━━━━━━━━━━━━━━ 1s 151ms/step
Prediction: 0.0059, Predicted label: 0, Actual label: 0
Prediction: 0.0077, Predicted label: 0, Actual label: 0
Prediction: 0.0126, Predicted label: 0, Actual label: 0
Prediction: 0.0297, Predicted label: 0, Actual label: 0
Prediction: 0.1141, Predicted label: 0, Actual label: 0
Prediction: 0.5524, Predicted label: 1, Actual label: 0
Prediction: 0.9727, Predicted label: 1, Actual label: 1
Prediction: 0.9982, Predicted label: 1, Actual label: 1
Prediction: 0.9995, Predicted label: 1, Actual label: 1
Prediction: 0.9997, Predicted label: 1, Actual label: 1
```

# Applications and Future Scope

| Applications | Future Scope |
| --- | --- |
| **1. Real-Time Data Analysis**: Continuously monitors and processes sensor data using embedded systems for immediate threat detection.. | **1. AI-Powered Predictive Models**: Future improvements could include more advanced AI models that predict disasters with higher accuracy. |
| **2. Machine Learning for Predictive Analytics**: Uses historical data and machine learning models (e.g., Random Forest, Decision Trees) to predict landslides based on environmental sensor readings. | **2. IoT Integration**: Integration with IoT devices for real-time monitoring and remote alerts via mobile apps. |
| **3. Early Warning Systems**: Provides early alerts for potential disasters, helping reduce loss of life and property. | **4. Cloud Computing & Big Data**: Leveraging cloud platforms for large-scale data storage and analytics, enabling more scalable, real-time processing of multi-sensor data across larger areas. |
| **4. Disaster Management**: Assists government and agencies in planning evacuation and resource deployment during emergencies. | **1. Integration with Image Processing**: Use satellite imagery or drones to perform real-time image analysis for terrain changes and landslide risk mapping. |

Sinhgad Institutes

# Conclusions

- By utilizing sensors, a ESP32, and an LSTM model, the system continuously collects and analyzes data, activating alarms if thresholds are exceeded to warn nearby individuals.
- The integration of IoT with machine learning not only enhances early warning accuracy but also enables remote data access and historical trend analysis through cloud platforms like ThingSpeak, laying the groundwork for intelligent, data-driven disaster management systems.
- This scalable, real-time monitoring solution demonstrates significant potential for disaster prevention and mitigation, supporting the safety of communities in landslide-prone areas.

Sinhgad Institutes

# Refrences


1. **Zhu, Q., Xu, X., Pan, B., Chen, F., & Zhao, Q.** (2022). *Deep Learning-Based Landslide Susceptibility Mapping*. Engineering Geology, Elsevier.
2. **Youssef, N., Pourghasemi, M., & Demirci, D.** (2021). *Landslide Detection Using Remote Sensing and Machine Learning*. Remote Sensing, MDPI.
3. **Hong, X., Liu, Y., Wu, J., & Li, T.** (2020). *A Comparative Study of Machine Learning Algorithms for Landslide Susceptibility Mapping*. Landslides, Springer.
4. **Tan, J., Wang, C., & Zhang, L.** (2021). *Early Warning of Landslides Using IoT and Edge Computing*. IEEE Internet of Things Journal.
5. **Zhao, F., Liu, H., & Chen, K.** (2022). *A Deep Learning Framework for Real-Time Landslide Detection Using Seismic Data*. Computers & Geosciences, Elsevier.
6. **Kumar, R., Singh, S., & Mehta, P.** (2019). *Multi-Criteria Decision Analysis for Landslide Hazard Assessment Using GIS*. Geocarto International, Taylor & Francis.
7. **ThingSpeak Documentation** : MathWorks. Available at: https://thingspeak.com
8. **Adafruit Documentation** : DHT11 & MPU6050 Libraries and Unified Sensor Library. Available at : https://learn.adafruit.com
9. **Arduino IDE**: Arduino Software Platform. Available at:https://www.arduino.cc/en/software
10. **Wokwi Simulator** : Online Arduino & ESP8266 Simulator. Available at:https://wokwi.com

# THANK YOU