Analyzing the Role of Fusing Image and Time Series Data in Forecasting Rainfall-Induced Landslides Configuration Manual

MSc Research Project
MSc Artificial Intelligence

Siddhi Sawant
Student ID: x23181753

School of Computing
National College of Ireland

Supervisor: Prof. Abdul Shahid

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Siddhi Sawant
Student ID:	x23181753
Programme:	MSc Artificial Intelligence
Year:	2024-2025
Module:	MSc Research Project
Supervisor:	Prof. Abdul Shahid
Submission Due Date:	12/12/2024
Project Title:	Analyzing the Role of Fusing Image and Time Series Data in Forecasting Rainfall-Induced Landslides Configuration Manual
Word Count:	954
Page Count:	8

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Siddhi Sawant
Date:	12th December 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Analyzing the Role of Fusing Image and Time Series Data in Forecasting Rainfall-Induced Landslides Configuration Manual

Siddhi Sawant
x23181753

1 Introduction

Detailed instructions on how to setup a system or device are included in a configuration manual. The manual's objective is to completely outline how to conduct the research study. Additionally, it specifies the machine configuration required to build and operate the models. The steps include setting up both the minimal setup recommended for a project to succeed and the applications and packages that are required.

2 Project File details

Data preparation, exploration, modeling and evaluation were performed were conducted using Google Colab.

Case-study-1-File1 in google Colab: using 'GLC_EDA_v1.ipynb' fetch the Global_Landslide_Catalog_
from the GLC_DATASET used for data exploration of GLC dataset and generate a precipitation data and write to this location GLC_Dataset/GLCprecipitationdata.csv.

Case-study-2-File2 in google Colab : Using GLCTrainMultipleModelsV2.ipynb fetch the GLC dataset concatenated with precipitation dataset from this location GLC_Dataset/GLCprecipita
to perform modelling and evaluation of sequential models.

Case-study-3-File3 in google Colab : Using TrainResNet50V2_FN.ipynb fetch the .npy file from this location HR_GLDD_Dataset/NPY_Files to perform modelling and evaluation of Computer vision model.

Case-study-4-File4 in google Colab : Using TrainCombinedModelV2.ipynb fetch the trained models from location Models/CNNMODEL fetch ResNet_model.h5 , from /Models/LSTMMODEL/With_TPU/WO_SAMPLING/ fetch LSTMModelWSV1.h5 and from Models/BiLSTMMODEL/With_TPU/WO_SAMPLING/ fetch BiLSTMModelWSV1.h5 . Also fetch datasets from GLC_Dataset/GLCprecipitationdata.csv and from HR_GLDD_Dataset/NPY_
fetch .npy files from modelling and evaluating Combined (multi-modal) model.

3 System Specification

A system specification is a written document outlining a system's technical properties and requirements, including its components, functionalities, design, and other technical characteristics. Figure 3. illustrates the system configuration used in this project, while

Figure 1. and Figure 2. shows the Google Colab specifications. A minimum of 8 GB RAM is sufficient for basic operation, but the recommended configuration ensures smooth execution.

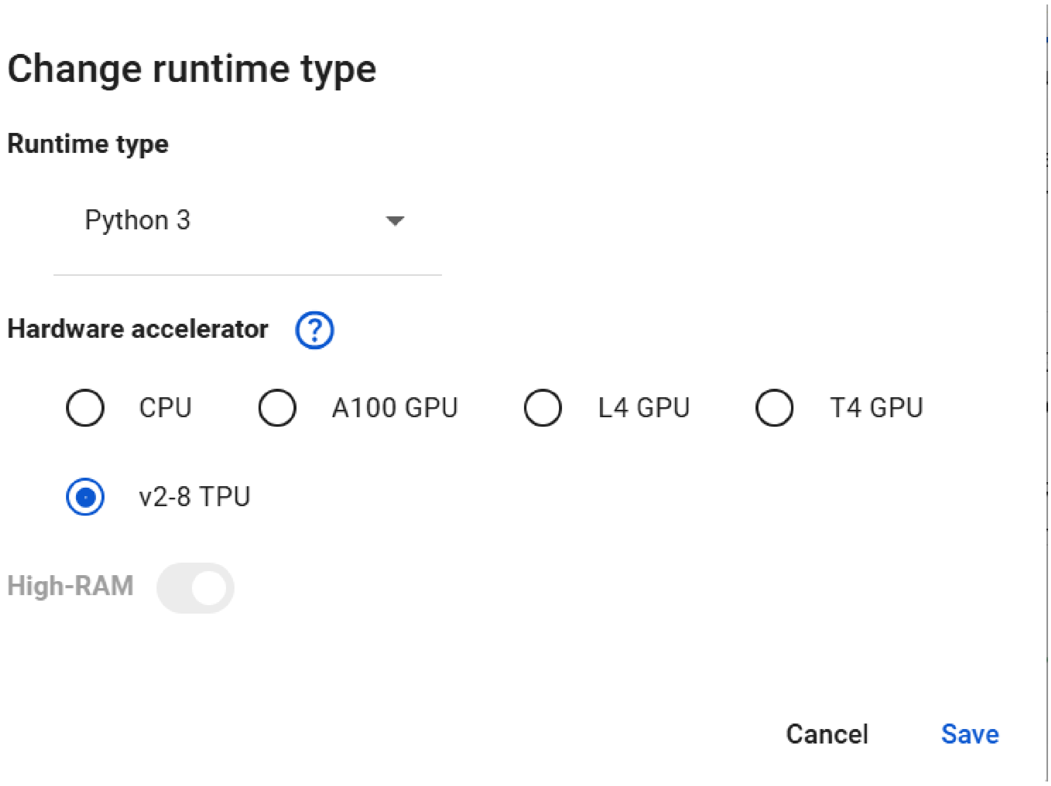

Figure 1: Laptop Specification

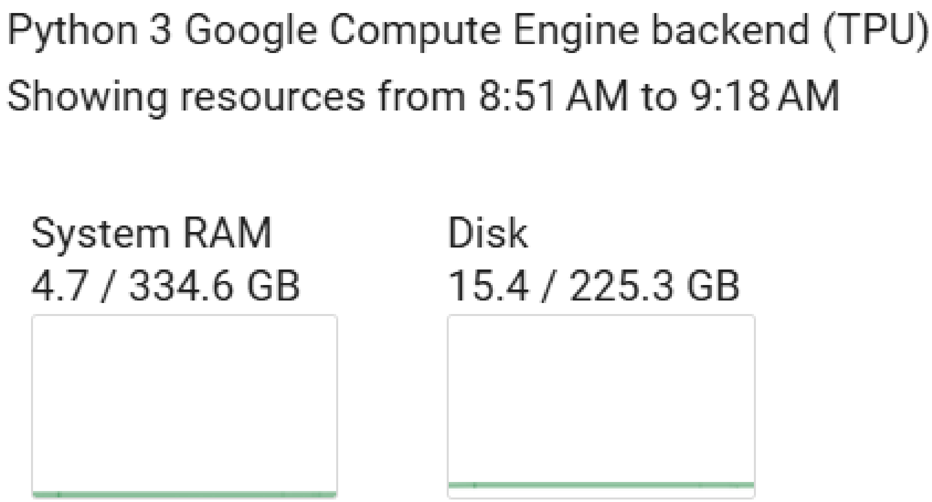

Figure 2: Laptop Specification

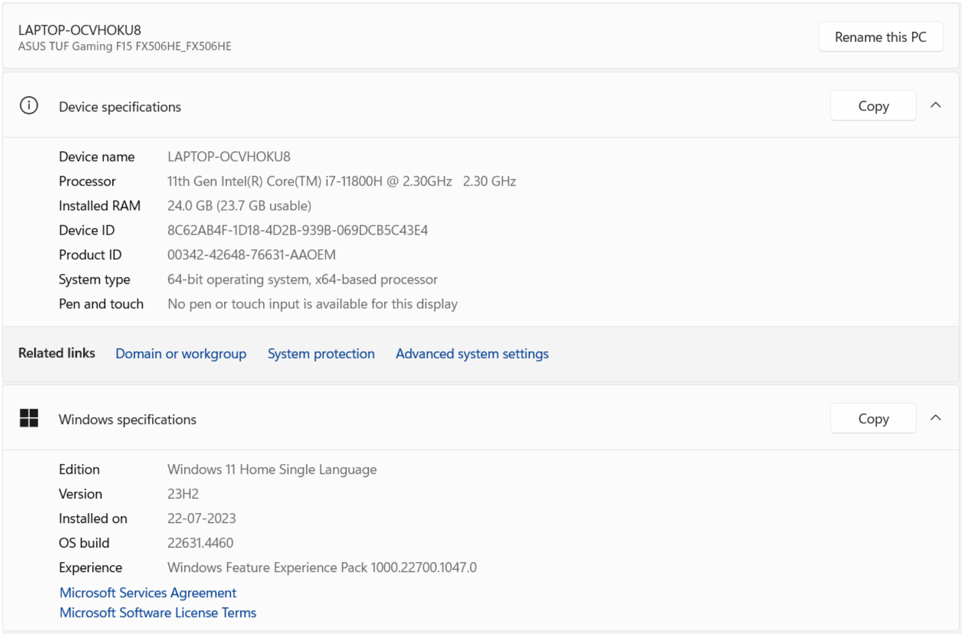

Figure 3: Laptop Specification

4 Software Used

- Google Drive: For storing dataset, source code and generated models.
- Google Collab: Used for Exploration ,processing , modelling and evaluation.
- Microsoft excel : Used for initial exploration.
- Microsoft word : Used for drafting initial report.

5 Dataset Description

In the project two dataset are used Time-series data and Image dataset. Below is description ,location of the dataset online and the saved in google drive with folder structure. These dataset are downloaded in the Google Drive and fruther accessed in Google colab for exploration, processing , modelling and evaluation.

- Global Landslide Catalog (GLC) [1]: is an online database of rainfall triggered landslides recorded from 1988 to 2017 by NASA's open portal since 2007 and comprises of 11,033 landslide events. Used in the project as base for rainfall Induced landslide events.Kirschbaum et al.; 2010(Kirschbaum et al.; 2015)

  Google drive folder structure : /content/gdrive/MyDrive/Data/GLC_Dataset

- CHIRPS [2] [3] : From Climate Hazards Group InfraRed Precipitation with Station data (CHIRPS), precipitation data is fetched for the landslide events by accessing CHIRPS file System(Funk et al.; 2015)

[1] https://catalog.data.gov/dataset/global-landslide-catalog-export
[2] https://www.chc.ucsb.edu/data/chirps
[3] https://data.chc.ucsb.edu/products/CHIRPS-2.0/global_monthly/netcdf/byYear/

Google drive folder structure : /content/gdrive/MyDrive/Data/GLC_Dataset

- HR_GLDD : High-Resolution Global landslide Detector Database (HR-GLDD), a high-resolution (HR) satellite dataset, containing image dataset in .npy files.Meena et al.; 2022 Google drive folder structure : /content/gdrive/MyDrive/Theses_Landslides/Data /HR_GLDD_Dataset

6 Project Development

After completing all these steps, you can launch Google Collab, open the source codes file from Code folder, then connect to the specified GPU, next you will have the option to run all of them simultaneously or each cell individually. The command "pip install package-name" should be used if a package needs to be installed.

6.1 Importing Libraries

The packages used in the project are displayed below as per the source code. The cloud platform comes with several necessary libraries already installed. If necessary, additional libraries should be imported.

For GLC_EDA_v1.ipynb the packages used are shown in figure Figure 4

```
[ ]  import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
     import plotly.express as px
     import xarray as xr
```

```
[ ]  from google.colab import drive
```

Figure 4: Packages used for GLC_EDA_v1.ipynb

For GLCTrainMultipleModelsV2.ipynb the packages used are shown in figure Figure 5

```
import pandas as pd
import numpy as np
from google.colab import drive
from sklearn.preprocessing import MinMaxScaler
from imblearn.over_sampling import SMOTE
import matplotlib.pyplot as plt
from sklearn.model_selection import TimeSeriesSplit
from keras.models import Sequential
from keras.layers import Bidirectional,LSTM,Dense,Dropout
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
from tensorflow.keras.metrics import Precision, Recall, AUC
```

Figure 5: Packages used for GLCTrainMultipleModelsV2.ipynb

For TrainResNet50V2_FN.ipynb the packages used are shown in figure Figure 6
For TrainCombinedModelV2.ipynb the packages used are shown in figure Figure 7

```
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.applications import ResNet50V2
from tensorflow.keras import layers, models
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau, ModelCheckpoint
from tensorflow.keras.layers import Input
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import GlobalAveragePooling2D, Dense, Dropout
```

```
[ ] from google.colab import drive
```

Figure 6: Packages used for TrainResNet50V2_FN.ipynb

```
import pandas as pd
import numpy as np
from google.colab import drive
import matplotlib.pyplot as plt
import tensorflow as tf
import keras
from tensorflow.keras.models import load_model
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau, ModelCheckpoint
from tensorflow.keras.layers import Dense, Dropout
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Dense, Concatenate
from keras.regularizers import l1_l2
```

Figure 7: Packages used for TrainCombinedModelV2.ipynb

6.2 Importing Files

```
[ ] from google.colab import drive
```

```
[ ] drive.mount('/content/gdrive')

Mounted at /content/gdrive
```

Figure 8: Process to Mount and Fetch Document in Google Collab

```
[ ] glcdata= pd.read_csv('/content/gdrive/MyDrive/Theses_Landslides/Data/GLC_Dataset/Global_Landslide_Catalog_Export.csv')
```

Figure 9: Process to fetch file in Google Collab

In this research project, once the required files mentioned in project file details are processed in Google Collab. To load dataset file, the following procedure must be followed. Figure 8 shows how to mount the Google drive to the collab. Only after one can fetch the data that has been put in the drive, the path should be set to the location where the file is as shown in Figure 9

6.3 Processing

- Treatment of Missing Values: Missing values are checked and treated if required.
- Null Values handling: The utilities IsNull() and sum() are employed to check for null values, while the duplicated () is used to examine duplicates.
- Normalization: MinMaxScaler techniques are used to normalize the data.

6.4 Modeling

Modelling in the research includes construction of sequential models like LSTM , BI-LSTM , computer vision Models like ResNet50v2 and Multi-modal feature fusion model.

6.4.1 Modelling Sequential Models

Figure 10 shows construction and fitting the model Long Short-term Memory with required Hyper-parameter tuning and implementing Callbacks and evaluating using cross validation.

```
earlystopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
lrate = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=3, min_lr=0.000001)

epochs=50
batch_size=32

for train_i, val_i in tssplit.split(X_sampled, y_sampled):
    X_train, X_val = X_sampled[train_i], X_sampled[val_i]
    y_train, y_val = y_sampled[train_i], y_sampled[val_i]

    mlstm= Sequential()

    mlstm.add(Bidirectional(LSTM(64, return_sequences=True), input_shape=(X_train.shape[1], X_train.shape[2])))

    mlstm.add(Dropout(0.3))

    mlstm.add(Bidirectional(LSTM(32)))

    mlstm.add(Dense(1, activation='sigmoid'))

    mlstm.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy', Precision(), Recall(), AUC()])

    mlstm.fit(X_train, y_train,validation_data=(X_val, y_val),epochs=epochs,batch_size=batch_size,callbacks=[earlystopping, lrate])

    val_loss, val_accuracy, val_precision, val_recall, val_auc = mlstm.evaluate(X_val, y_val)

    print(f"Validation Loss: {val_loss}, Validation Accuracy: {val_accuracy}, Validation Precision: {val_precision},Validation Recall: {val_recall}, Validation AUC: {val_auc} ")
```

Figure 10: Model Creation of LSTM

Figure 11 shows construction and fitting the model Bidirectional Long Short-Term Memory with required Hyper-parameter tuning and implementing Callbacks and evaluating using cross validation.

```
earlystopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
lrate = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=3, min_lr=0.000001)

epochs=50
batch_size=32

for train_i, val_i in tssplit.split(X_sampled, y_sampled):
    X_train, X_val = X_sampled[train_i], X_sampled[val_i]
    y_train, y_val = y_sampled[train_i], y_sampled[val_i]

    mlstm= Sequential()

    mlstm.add(Bidirectional(LSTM(64, return_sequences=True), input_shape=(X_train.shape[1], X_train.shape[2])))

    mlstm.add(Dropout(0.3))

    mlstm.add(Bidirectional(LSTM(32)))

    mlstm.add(Dense(1, activation='sigmoid'))

    mlstm.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy', Precision(), Recall(), AUC()])

    mlstm.fit(X_train, y_train,validation_data=(X_val, y_val),epochs=epochs,batch_size=batch_size,callbacks=[earlystopping, lrate])

    val_loss, val_accuracy, val_precision, val_recall, val_auc = mlstm.evaluate(X_val, y_val)

    print(f"Validation Loss: {val_loss}, Validation Accuracy: {val_accuracy}, Validation Precision: {val_precision},Validation Recall: {val_recall}, Validation AUC: {val_auc} ")
```

Figure 11: Model Creation of Bi-LSTM

6.4.2 Modelling ResNet50V2

Figure 12 shows construction and fitting the model Bidirectional Long Short-Term Memory with required Hyper-parameter tuning and implementing Callbacks and evaluating.

```
earlystopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
lr = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=5, min_lr=0.000001)
checkpoint_cb = ModelCheckpoint(filepath='/content/gdrive/My Drive/Theses_Landslides/Models/CNN_MODEL/ResNet_model.h5',monitor='val_loss',save_best_only=True,verbose=1)
```

```
history=None
```

```
resnetm = Sequential()

resnetm.add(base_model)

resnetm.add(GlobalAveragePooling2D())

resnetm.add(Dense(256, activation='relu'))

resnetm.add(Dropout(0.5))

resnetm.add(Dense(1, activation='sigmoid'))
```

```
resnetm.compile(optimizer=Adam(learning_rate=0.0001),loss='binary_crossentropy',metrics=['accuracy'])
```

```
history=resnetm.fit(X_train, Y_train,validation_data=(X_val, Y_val),epochs=50,batch_size=32,callbacks=[earlystopping, lr,checkpoint_cb])
```

Figure 12: Model Creation of ResNet50V2

6.4.3 Modelling Multi-modal Feature Fusion

Figure 13 shows construction and fitting the model Multimodal Feature Fusion with regularization with required Hyper-parameter tuning and implementing Callbacks and evaluating.

```
earlystopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
lr = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=5, min_lr=0.000001)
checkpoint_cb = ModelCheckpoint(filepath='/content/gdrive/My Drive/Theses_Landslides/Models/CNN_MODEL/ResNet_model.h5',monitor='val_loss',save_best_only=True,verbose=1)
```

```
history=None
```

```
resnetm = Sequential()

resnetm.add(base_model)

resnetm.add(GlobalAveragePooling2D())

resnetm.add(Dense(256, activation='relu'))

resnetm.add(Dropout(0.5))

resnetm.add(Dense(1, activation='sigmoid'))
```

```
resnetm.compile(optimizer=Adam(learning_rate=0.0001),loss='binary_crossentropy',metrics=['accuracy'])
```

```
history=resnetm.fit(X_train, Y_train,validation_data=(X_val, Y_val),epochs=50,batch_size=32,callbacks=[earlystopping, lr,checkpoint_cb])
```

Figure 13: Model Creation of Multi-modal Feature Fusion

7 GitHub link

Following is the GitHub link for the dataset and code artifact [4]

References

Funk, C., Peterson, P., Landsfeld, M., Pedreros, D., Verdin, J., Shukla, S., Husak, G., Rowland, J., Harrison, L., Hoell, A. and Michaelsen, J. (2015). The climate hazards infrared precipitation with stations-a new environmental record for monitoring extremes, *Scientific Data* **2**.

[4] https://github.com/Siddhi0602/Theses_LandlsidePrediction/tree/main/Theses_23181753/Code

Kirschbaum, D. B., Adler, R., Hong, Y., Hill, S. and Lerner-Lam, A. (2010). A global landslide catalog for hazard applications: method, results, and limitations, *Natural Hazards* **52**(3): 561–575.

Kirschbaum, D., Stanley, T. and Zhou, Y. (2015). Spatial and temporal analysis of a global landslide catalog, *Geomorphology* . In press.

Meena, S. R., nava, L., Bhuyan, K., Puliero, S., Soares, L. P., Dias, H. C., Floris, M. and Catani, F. (2022). Hr-gldd: A globally distributed high resolution landslide dataset. **URL:** *https://doi.org/10.5281/zenodo.7189381*