Risk Assessment of Railway Tracks in Floodplain area using Digital Surface Model and Computer Vision

Authors: Watcharapong Wongkaew, Wachira Muenyoksakul, Krittiphong Manachamni

Advisors: Asst. Prof. Dr. Chayut Ngamkhanhong, Dr. Tanawat Tangjarusritaratorn

Chulalongkorn University

18 July 2023

Railway track flooding

Sources: SRT Hua Hin Station Master; (Oct., 2022)

Extensive damage to railway infrastructure

Risk to the safety of railway rolling stocks along with passengers

Needs to understand the risk and prediction of railway track flooding in the regions

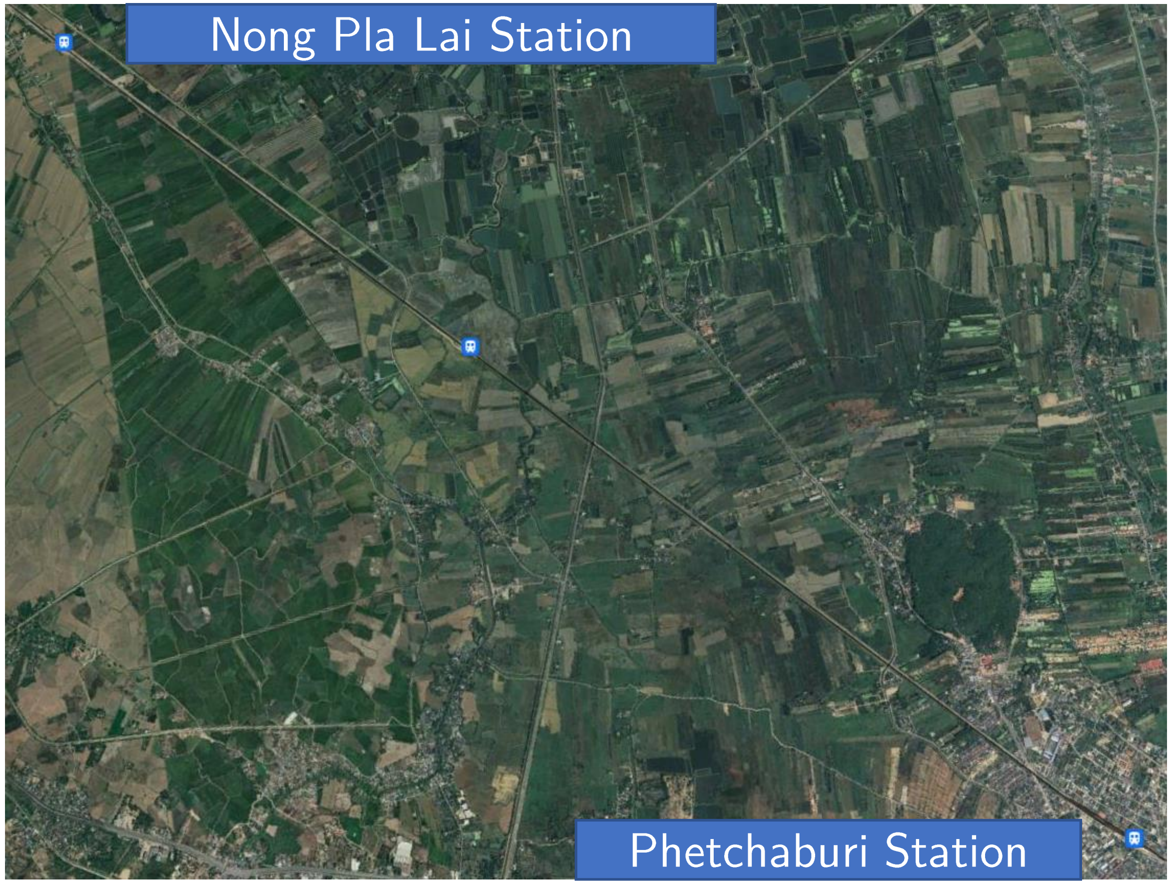
Area of interests

Area

- Phetchaburi Province
- 11 km. of railway tracks between 3 stations

Features

- Floodplain area
- Repeated flood over the course of 11 years



Data Collection

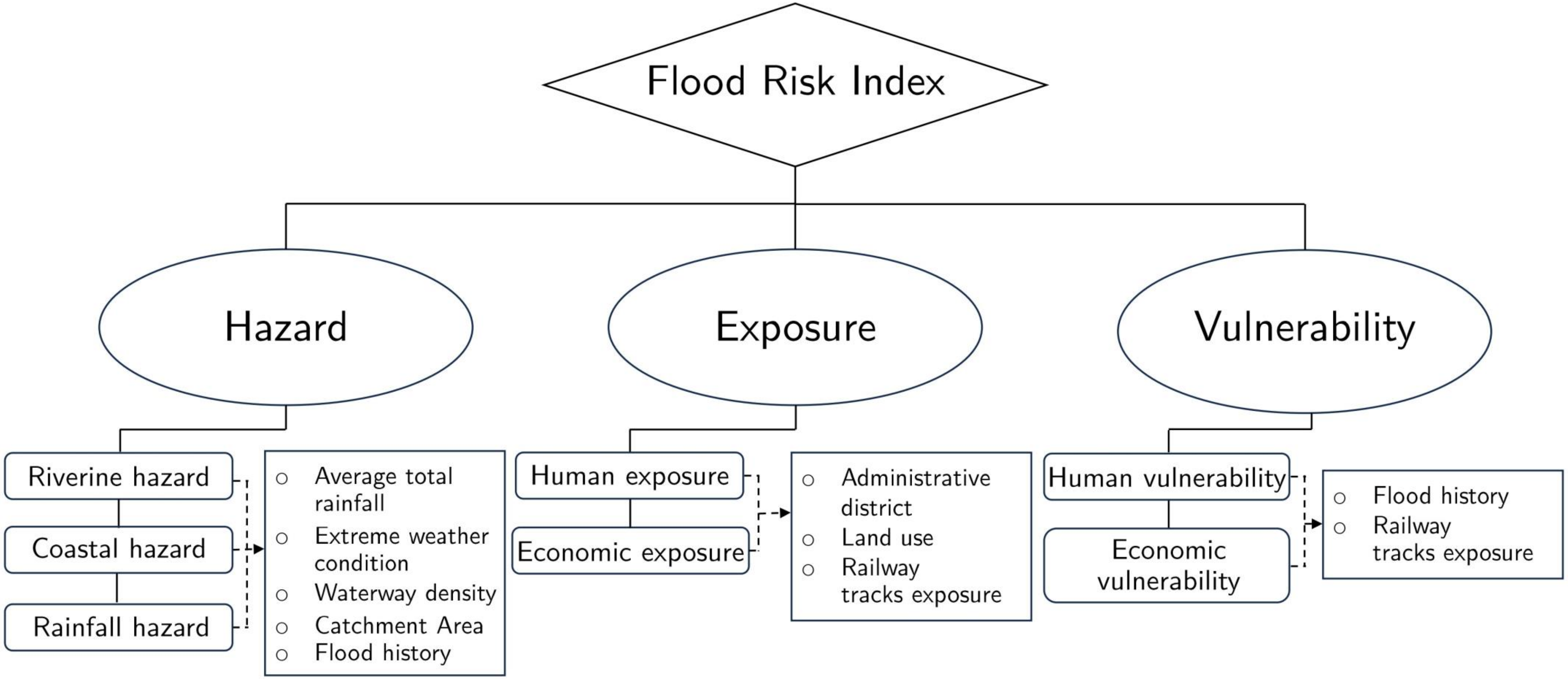
Data	Sources	Boundary of data
Digital Surface Model	GTOPO30/ NASA SRTM DEM30	-15 – 1506 m.
Average Total Rainfall in rainy season	Thai Meteorological Department	0 – 13.6 cm.
Elevation/ Slope	Contour maps from Department of Geology, Chulalongkorn University NASA SRTM DEM30	numeric
Railway Tracks	Department of Geography, Chulalongkorn University	-
Flood history	Geo-Informatics and Space Technology Development Agency	0 – 9
Waterway density	Department of Geology, Chulalongkorn University	-
Catchment Area	Department of Geology, Chulalongkorn University	0 – 2,210 km^2
Extreme Weather Condition	Thai Meteorological Department	0 – 20 days

Data Collection

Elevation, slopes from the contour and DSM

Drainage basin, drainage network, minor and major streams

Flood Risk Index Calculation



Flood Risk Index Calculation

To calculate the Flood Risk Index or FRI, the methodology that we used is to combine 3 components of risk as hazard in form of Flood Hazard Score (FHS), exposure as Flood Exposure Score (FES) and vulnerability as in Flood Vulnerability Score (FVS) into index in the form of Equation which gives us the index to interpret and into risk map

$$FRI = 0.5(FHS) + 0.3(FES) + 0.2(FVS)$$

All of the data in Table 2 is interpreted into one score as in FHS, FES and FVS which can be obtained by Equation

$$FHS = f(Elevation, Slope, Precipitation, Thunderstorm)$$
$$FES = f(ArialExtent, RailTrackElevation)$$
$$FVS = f(FloodHistory)$$

Which *f(x)* is functions of variable that model automatically tunes
FRI will be classified into 4 categories ranging from 0 – 1

Flood Risk Index Calculation

Flood risk index range	Description	Intensity (Height relative to rail head)	Frequency (Number of flood in monsoon season)
< 0.01	Very small risk to flood	< 0.01	< 4
0.01 – 0.25	Small risk to flood	0.01 - 0.25	5 - 8
0.25 – 0.50	Vulnerable to flood	0.25 - 0.50	9 - 12
0.50 – 0.75	High vulnerability to flood	0.50 - 0.75	13 - 16
0.75 – 1.00	Very high vulnerability to flood	> 0.75	> 16

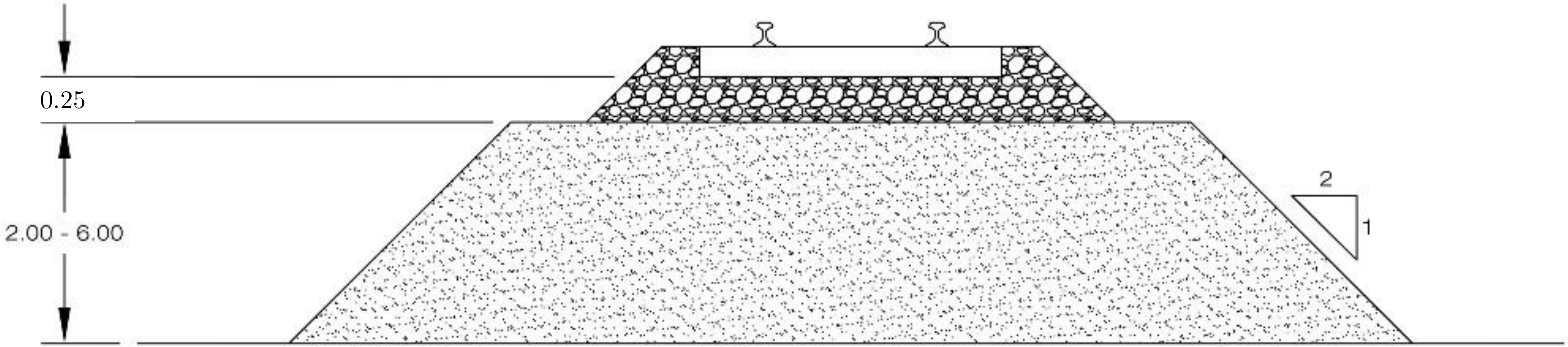
Risk Index Interpretation

Idealize track condition

Engineering Parameters	Data
Track Gauge	1 Meter
Rail Section	Bs100a
Sleeper Type	Prestress Concrete
Sleeper Dimension	200 X 50 X 25 cm
Ballast Material	Andesite, Rhyolite
Ballast Depth	25 cm
Embankment Height	2-6 m

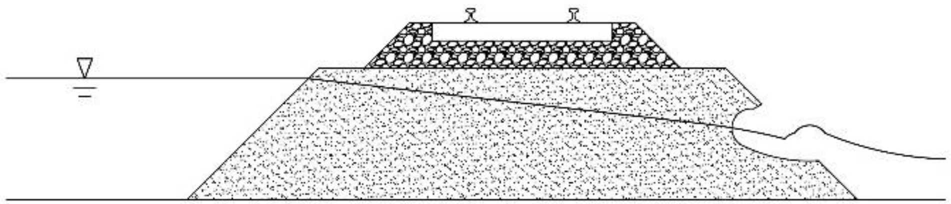
Risk Index Interpretation

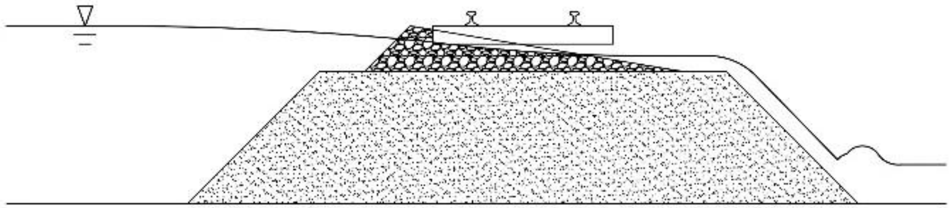
Idealize track condition

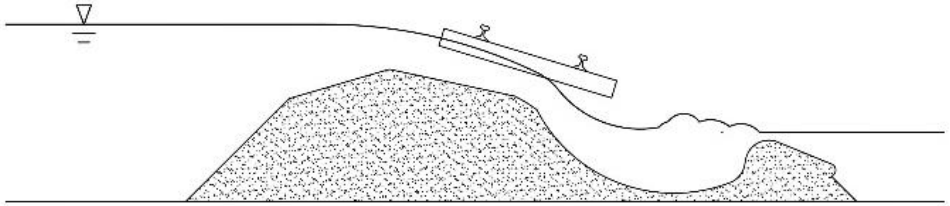


Risk Index Interpretation

Track damage interpretation: One-sided flood

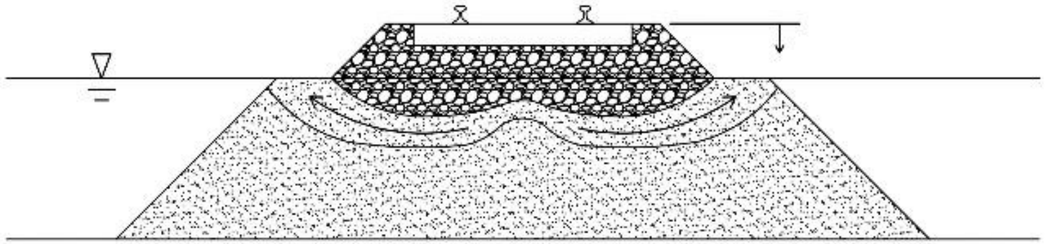

(a.) Embankment seepage


(b.) Ballast washaway

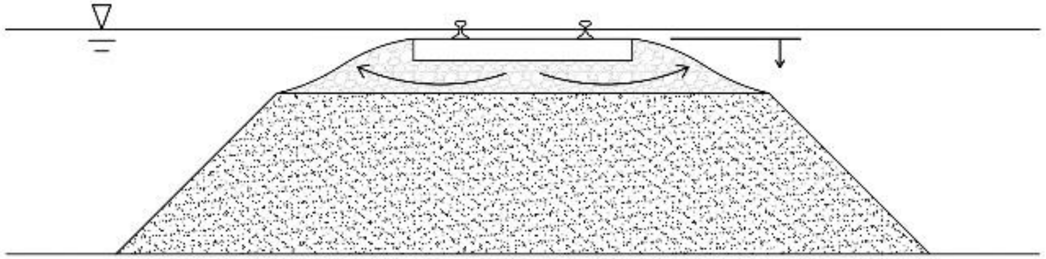

(c.) Embankment scour

Risk Index Interpretation

Track damage interpretation: Two-sided flood



(a.) Mud pumping and track settlement due to embankment material movement



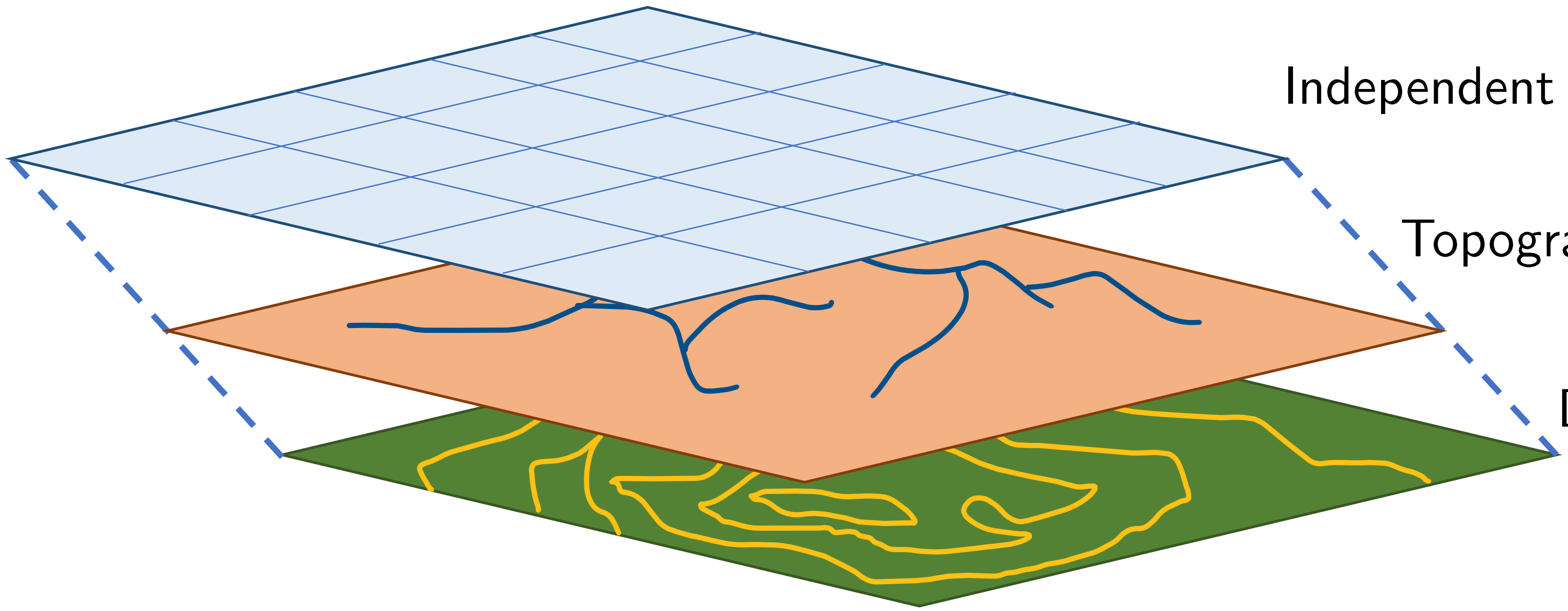
(b.) Track settlement due to reduction of ballast interlocking

Model/ Estimators

Model	Type of processing
Multinomial Logistic Regressor*	Classifier
Multinomial Naïve Bayes*	Classifier
Multi-Layer Perceptron (MLP)*	Fully connected ANN
Convolutional Neural Network based	Backpropagation
Transformers based model	Attention-based

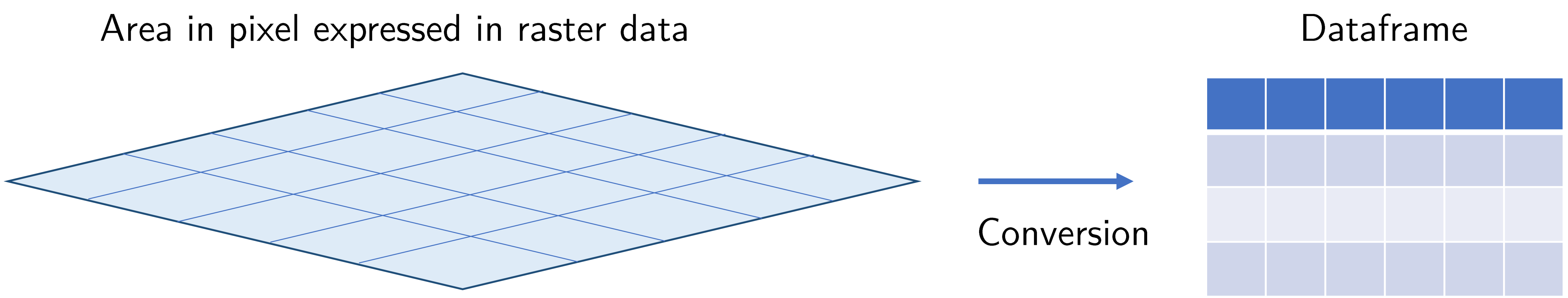
*Model use for **only** classifying flood on rail

Model inputs



Input stacked and digitalized onto one map and use ... function to see the area that flooded

Model inputs



The digitalized data that we interest will be transfer in to dataframe which have attributes of the input data as column and row as each pixel

Model parameters

Train-test-split of data: since data is not that much, we will be splitting and engineering data to both validate and test more efficiently

Splitting Type	Number of years
Training	10
Validation	2
Tests	4
Total flooded	16
Total Data	30

Model metric

F1 Score, Precision & Recall

		True Label (Data)	
		Positive	Negative
Prediction Label	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

$$\text{Precision} =$$

$$\text{Recall} =$$

$$\text{F1 Score} =$$

Review

Most data layers are normalized so that data from different sources can be compared qualitatively. The final risk map is calculated using Equation (1), multiplying the flood hazard, flood exposure, and flood vulnerability indices.

$$\mathrm{FRI} = \mathrm{FHI} \times \mathrm{FEI} \times \mathrm{FVI} \tag{1}$$

where:
FHI = Flood Hazard Index (-)
FEI = Flood Exposure Index (-)
FVI = Flood Vulnerability Index(-)

Review

The composition of these vulnerabilities are formulated in an equation shown

$$FVI = \frac{A + U + L}{3}$$

FVI = Vulnerability Index
A = Age composition
U = Urbanization
L = Literacy

The calculation for exposure is shown in Equation.

$$FEI = W_p \frac{C + 0.5R + U + S_d + H_d + R_d}{5.5}$$

FEI = Flood Exposure Index
W_p = Population data
C = Cropland
R = Rice
U = Urban
S_d = School distance
H_d = Hospital distance
R_d = Road distance

Results

Multilayer Perceptron results stand out in predicting flood occurring both training and testing set. Low recall show that model still missed some case in predicting flood event. Due to 4 support in dataset can't show the true performance of the model.

```
In [139]: y_pred = clf.predict(data)
          print(classification_report(y_data, y_pred))
```

	precision	recall	f1-score	support
0	0.90	1.00	0.95	9
1	1.00	0.86	0.92	7
accuracy			0.94	16
macro avg	0.95	0.93	0.94	16
weighted avg	0.94	0.94	0.94	16

```
In [140]: y_pred = clf.predict(X_test)
          print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.67	1.00	0.80	2
1	1.00	0.50	0.67	2
accuracy			0.75	4
macro avg	0.83	0.75	0.73	4
weighted avg	0.83	0.75	0.73	4

Results

Multi Layer Perceptron (MLP) performs the best in both validation set, and testing set in prediction flood occurring in interest area.

	Validation set					Testing set				
	Precision	Recall	F1 score	IoU	Dice Index	Precision	Recall	F1 score	IoU	Dice Index
Logistic Regression (Macro-average)	0.68	0.69	0.68	-	-	0.50	0.50	0.50	-	-
Naïve Bayes (Macro-average)	0.54	0.53	0.52	-	-	0.27	0.35	0.40	-	-
MLP (Macro-average)	0.95	0.93	0.94	-	-	0.83	0.75	0.73	-	-

Issues & Limitations

As demonstrated, we face a challenge due to the lack of sufficient hydrological data to accurately predict floods and propose effective FRI in this paper.

To address this, we intend to develop a model that incorporates seasonal variations and geological factors, in addition to annual data, to enhance our comprehension of the system.

```
In [128]: model.fit(X_train, y_train)
Out[128]: LogisticRegression(C=0.7)

In [131]: y_pred = model.predict(data)
          print(classification_report(y_data, y_pred))

              precision    recall  f1-score   support

           0       0.70      0.78      0.74         9
           1       0.67      0.57      0.62         7

    accuracy                           0.69        16
   macro avg       0.68      0.67      0.68        16
weighted avg       0.69      0.69      0.68        16

In [132]: y_pred = model.predict(X_test)
          print(classification_report(y_test, y_pred))

              precision    recall  f1-score   support

           0       0.50      0.50      0.50         2
           1       0.50      0.50      0.50         2

    accuracy                           0.50         4
   macro avg       0.50      0.50      0.50         4
weighted avg       0.50      0.50      0.50         4
```

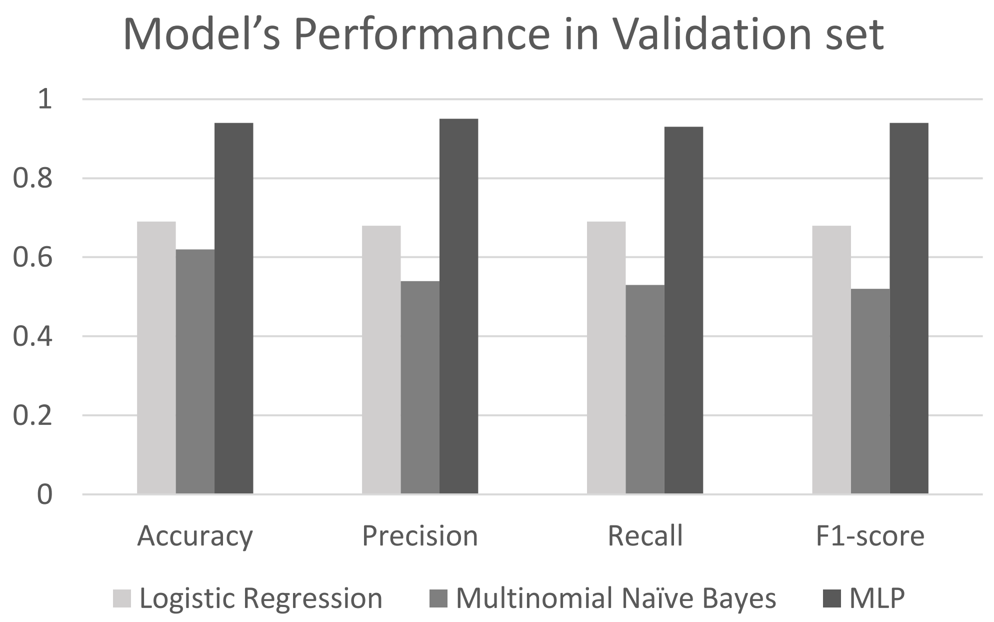
Discussion

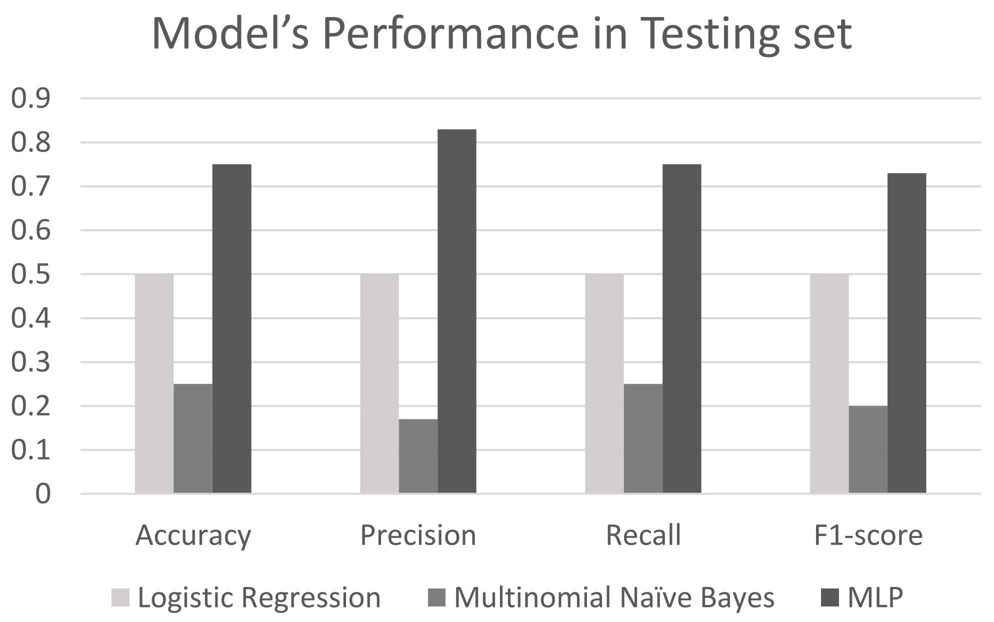
The results show that the Multi-Layer Perceptron (MLP) area has the highest precision, recall, and F1 score, indicating that the model is most effective when applied with this category of area. For others, overall results are not good due to the small area and number of years, and variation. Providing additional data, such as the month of the seasonal flood, may improve the results.

	Validation set					Testing set				
	Precision	Recall	F1 score	IoU	Dice Index	Precision	Recall	F1 score	IoU	Dice Index
Logistic Regression (Macro-average)	0.68	0.69	0.68	-	-	0.50	0.50	0.50	-	-
Naïve Bayes (Macro-average)	0.54	0.53	0.52	-	-	0.27	0.35	0.40	-	-
MLP (Macro-average)	0.95	0.93	0.94	-	-	0.83	0.75	0.73	-	-

Conclusions

In summary, MLP shows great performance in predicting floods in railways.





Conclusions

The digital surface model used in this study help in **identifying the damages to railway tracks, track structures, and surrounding areas** due to the influences of different flooding conditions.

This study is expected to provide us with an **understanding of the relationship between flooding conditions and railway embankment and track damages**, which can be used to **develop more effective flood risk management strategies**. With a remarkable potential, the findings of this study will contribute to the development of a more resilient railway infrastructure that can withstand the impacts of natural disasters.

Improvements

We train another model to represent chance of flood occur using same architect.

In [68]:
```
clf = MLPClassifier(
    hidden_layer_sizes=1000,
    activation='relu',
    solver='adam',
    learning_rate='adaptive',
    learning_rate_init=0.0001,
    max_iter=50,
    verbose=True).fit(X_train, y_train)
model.score(X_train, y_train)
```

```
Iteration 50, loss = 0.59036261
```

In [71]:
```
yhat = clf.predict_proba(X_train)
# summarize the predicted probabilities
print('Predicted Probabilities: %s' % yhat)
```

```
Predicted Probabilities: [[0.63313764 0.36686236]
 [0.61104507 0.38895493]
 [0.64501515 0.35498485]
 [0.53245518 0.46754482]
 [0.60401511 0.39598489]
 [0.26872087 0.73127913]
 [0.50854322 0.49145678]
 [0.48903307 0.51096693]
 [0.49341669 0.50658331]
 [0.55203711 0.44796289]
 [0.49037988 0.50962012]
 [0.44180372 0.55819628]]
```

Improvements

We train another model to represent chance of flood occur using same architect.

```
In [68]: clf = MLPClassifier(
             hidden_layer_sizes=1000,
             activation='relu',
             solver='adam',
             learning_rate='adaptive',
             learning_rate_init=0.0001,
             max_iter=50,
             verbose=True).fit(X_train, y_train)
         model.score(X_train, y_train)
```

```
Iteration 50, loss = 0.59036261
```

```
In [69]: y_pred = clf.predict(data)
         print(classification_report(y_data, y_pred))
```

	precision	recall	f1-score	support
0	0.75	0.67	0.71	9
1	0.62	0.71	0.67	7
accuracy			0.69	16
macro avg	0.69	0.69	0.69	16
weighted avg	0.70	0.69	0.69	16

```
In [70]: y_pred = clf.predict(X_test)
         print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	1.00	0.50	0.67	2
1	0.67	1.00	0.80	2
accuracy			0.75	4
macro avg	0.83	0.75	0.73	4
weighted avg	0.83	0.75	0.73	4

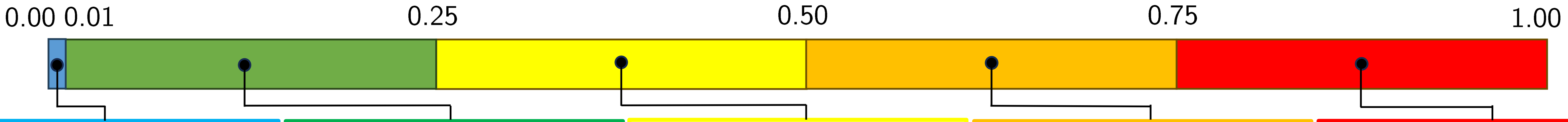
Acknowledgements

The authors would like to thank all of the data sources including Thai Meteorological Department, GISTDA, Department of Geology, Chulalongkorn University, and Department of Geography, Chulalongkorn University. Also, the authors would like to thank Associate Professor Dr. Pannee Cheewinsiriwat, and Dr. Phathinan Thaithatkul for the advice, methodology and the data on Geographic Information System (GIS) which we used QGIS and Python 3 in visualizing.

Appendix

Results

FRI Interpretation



0.00 – 0.01
Very small risk to flood
- No or negligible flood over the years
- Drainage system can drain out all water after severe weather
- No potential damage to railway track

0.01 – 0.25
Small risk to flood
- Minor and infrequent flood can be occurred over the years
- Drainage system can drain out most to all water after severe weather
- Very low potential damage to railway track

0.25 – 0.50
Medium risk to flood
- Some flood can be occurred over the years
- Drainage system can drain out some water after severe weather
- Have some potential damage to railway track

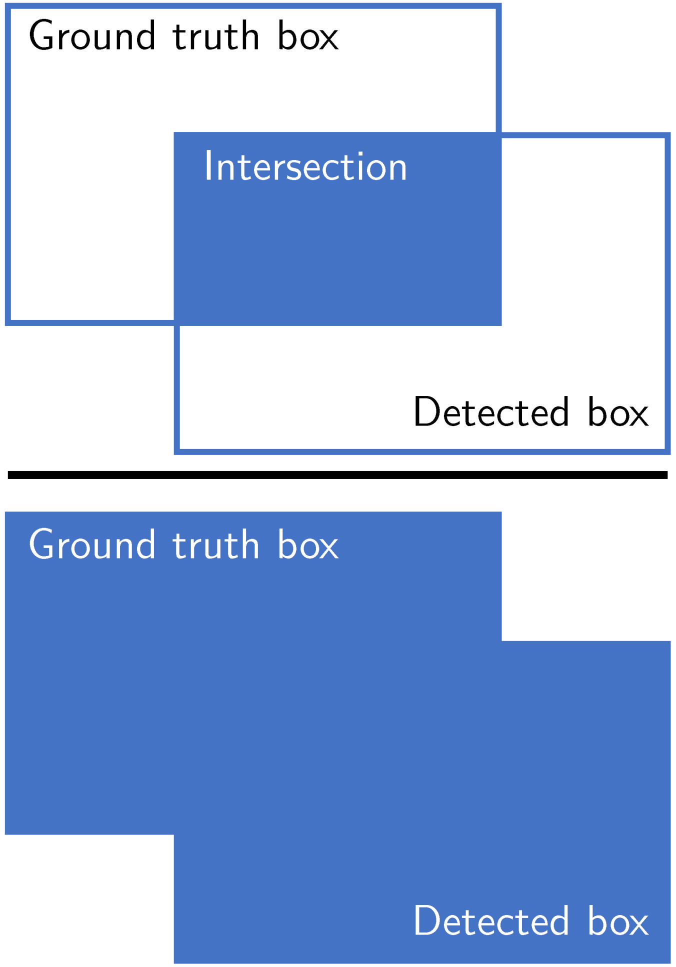
0.50 – 0.75
High risk to flood
- Frequent flood can be occurred over the years
- Drainage system problem can be occurred after severe weather
- High potential damage to railway track

0.75 – 1.00
Very high risk to flood
- Major or multiple flood over the years
- Drainage system cannot handle water after severe weather
- Very high potential damage to railway track

Model metric

Intersect over Union

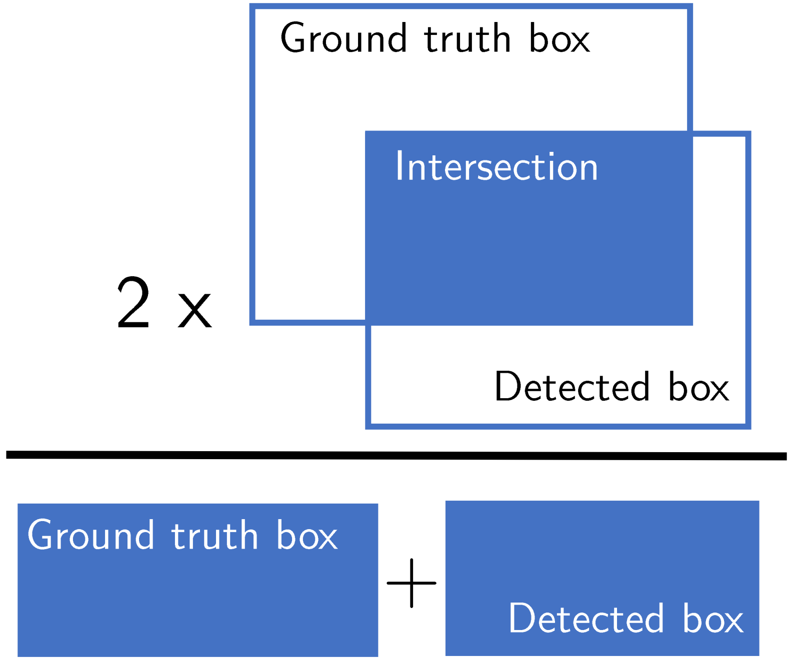
$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} =$$



Model metric

Dice Index

$$\text{Dice Index} = \frac{\text{2 x Area intersection of both box}}{\text{Area of Truth box + detected box}} =$$



Results

Prediction results on FRI show that most of our study area are Medium Risk area

FRI Range	Arial Extent
Low Risk area	0.12
Medium Risk area	0.35
High Risk area	0.25
Repeated Flood area	0.28

Improvements

Model mostly report chance of flood occuring in

In [68]:
```
clf = MLPClassifier(
    hidden_layer_sizes=1000,
    activation='relu',
    solver='adam',
    learning_rate='adaptive',
    learning_rate_init=0.0001,
    max_iter=50,
    verbose=True).fit(X_train, y_train)
model.score(X_train, y_train)
```