Comparing Machine Learning Methods for Early Warning of Floods and Landslides in Thailand

Joanna Sophie Abraham, Supatta Labaiusuh, Ekkasit Ismael, and Chatree Nilnumpetch*

Department of Computer Science, Faculty of Science, Ramkhamhaeng University, Bangkok, 10240, Thailand

* Corresponding author. E-mail address: nchatree@ru.ac.th

Received: 2 April 2023; Revised: 28 September 2023; Accepted: 9 October 2023; Available online: 16 November 2023

Abstract

Flood disasters and landslides have a strong impact on people's lives, property, and the economy of the country. Heavy rainfall is the primary cause of these disasters. Therefore, prediction warnings is necessary for people to help them prepare for the disaster in time. This paper outlines the process used to identify appropriate models for prediction warnings for floods and landslides by comparing the recall performance of eight different models. The models were Rule-Based, K-nearest Neighbor, Decision Tree, Random Forest, Support Vector Machine, Naïve Bayes, Logistic Regression, and Multilayer Perceptron. The process involved five phases: data collection, data pre-processing, building a model, 5-fold cross-validation, and model evaluation. This study utilized a rainfall-related dataset collected by the Department of Water Resources in Thailand for training and testing the models. After the process was applied along with a detailed evaluation, it found that when 5-fold cross-validation was applied, better performance was achieved with Random Forest having the highest recall value at 74%, followed by Decision Tree, Multilayer Perceptron and Support Vector Machine. From these results, it can be concluded that the Random Forest model is suitable for predicting warnings and can be implemented in future works for developing an early warning application to reduce the aftermath of these disasters does helping achieve the Sustainable Development Goal 11.5.

Keywords: Machine Learning; Random Forest; Decision Tree; Early Warning; Landslides

Introduction

The monsoon significantly impacts the weather in Thailand, bringing with it persistent heavy rain, which is the main cause of floods and landslides (Department of Water Resources, Ministry of Natural Resources and Environment, 2009). According to the disaster statistics for the year 2021 released by the Department of Disaster Prevention and Mitigation (2022), there were more than 2.5 million people affected by floods and over 1.1 million households that suffered damage. In addition, landslides affected almost 2,000 people and damaged about 800 households. Therefore, it is crucial to be able to issue warning predictions for floods and landslides so that those in high-risk areas can receive reliable warnings in advance and make necessary arrangements.

Akter, Sadman, and Bala (2022) explained the process of machine learning. This requires the collection of data to be used in algorithms to solve problems like classification and prediction. It is connected to several related fields like data mining and artificial intelligence. There are seven stages in the machine learning process: data collection, data cleaning and data pre-processing, feature engineering, the definition of a machine learning model, training, performance evaluation and prediction/ inference. They further described that the machine learning process has four primary categories: identified-supervised, unsupervised, semi-supervised, and reinforcement learning.

Numerous studies have been conducted on landslide prediction. Yi, Zhang, Zhang, and Xu (2019) compared the performance of three models for mapping landslide susceptibility in China: Logistic Regression, Naïve Bayes, and Support Vector Machine. The models were trained using rasterized recognized landslides and non-landslide

data. To ensure a fair evaluation of performance for each model, the same training and test datasets were utilized for training and evaluating each model. The study found that the logistic regression model had the highest success and prediction rate at 86.09% and 85.94%, respectively, followed by the Naïve Bayes model with a success rate of 85.69% and a prediction rate of 84.94%. In a study conducted by Chang, Hwang, Liu, Wang, and Wang (2011), a combination of supervised and unsupervised classification algorithms using LiDAR data were used to monitor long-term crumbling areas in Taiwan that were induced by heavy rainfall. The supervised classification algorithms applied were Support Vector Machine and K-nearest Neighbor, while region-based segmentation was the unsupervised classification algorithm used to merge image elements. The results revealed that Support Vector Machine and K-nearest Neighbor had a producer accuracy of 85.68% and 84.72%, respectively, and user accuracy of 80.41% and 79.85%, respectively. They discovered that K-nearest Neighbor caused probable misinterpretation, but it identified minor objects better than Support Vector Machine. In a study by Martin and Chai (2022), three models, namely K-nearest Neighbor, Random Forest, and XGBoost, were evaluated and compared for predicting landslide susceptibility in Malaysia. The models were trained and tested using satellite images, locations with landslide records, and topographical information. Results showed that all three models achieved relatively high AUC scores. The AUC score for K-nearest Neighbor was the highest at 87.52%, followed by Random Forest at 84.34%, and XGBoost at 73.07%, respectively. It also determined that K-nearest Neighbor was particularly effective in identifying high and very high susceptible areas, whereas Random Forest and XGBoost demonstrated a high sensitivity to landslides. Deep Learning and Machine Learning approaches were proposed and compared for predicting landslide susceptibility in Turkey (Yilmaz, Teke, & Kavzoglu, 2022). The study used Depthwise Separable Convolutional Neural Networks as the Deep Learning model and Random Forest for Machine Learning. Their performances were evaluated based on the 13 factors that contribute to landslides. The results showed that both models achieved high levels of training and validation accuracy at 97% and 96%, respectively. However, after analyzing the accuracy assessment metrics, Deep Learning was found to perform better with an overall accuracy of 96%, while Machine Learning had an overall accuracy of 87%.

Several research projects have been done on disaster warnings in Thailand, but the majority of these concentrated on the use of sensors to measure the values of the factors causing these disasters, for example, measuring water level and rainfall for flood warning system in a tunnel (Jinsikhong, Musika, Amonlak, & Nonthakarn, 2022), measuring environmental and carbon dioxide values for wildfire alarm (Wongka, Montri, & Chaikhamwang, 2022) and measuring temperature, humidity and smoke volume for an automatic forest fire alarm system (Phomsak & Khruahong, 2021). The data collected from these sensors were then sent to be processed for further warning without the use of Machine Learning.

There are also local studies related to landslides, such as a study on landslide warnings in Thailand that used the process of selecting a time frame that corresponds the most to landslides occurrences and set rainfall boundaries by separating it into before and after landslides occurrence (Thaiyuenwong, Nuthimthong, Detmak, & Soralump, 2013). Rangsiwanichpong (2020) utilized the Geographic Information System together with the Multiple Logistic Regression model to compute the probability of landslides using maximum daily rainfall data. The artificial neural network was adopted to predict areas that are susceptible to landslides in Uttaradit province (Inruang & Chaipimonplin, 2558), while Pomthong & Asavasuthirakul (2017) utilized the Geographic Information System to analyze areas prone to landslides in Phetchabun province. Such studies did not apply

machine learning for predicting early warning for floods and landslides. Machine learning was only applied for predicting landslide susceptibility. Therefore, this paper chose to use machine learning models in training and testing to make predictions.

The goal of this paper is to compare eight models and find suitable models for prediction warnings for floods and landslides in Thailand. It is imperative to have a historical rainfall dataset that has been used for previous warnings to train and test the models and make predictions. The performance of each model is used to determine the models that are appropriate for making predictions and can be utilized as part of developing an early warning application in the future.

In the next section, the methods and materials applied in this research were explained including the process that was created to identify the appropriate models for prediction warnings. The results were then presented and discussed. Finally, the paper was concluded by summarizing the content and mentioning the limitations and possible future works.

Methods and Materials

In this section, there are five stages involved in identifying the appropriate models. These include gathering data, preparing the data for analysis, building and configuring each model, performing cross-validation on each chosen model, and evaluating its performance. The diagram shown in Figure 1 describes the mentioned process.

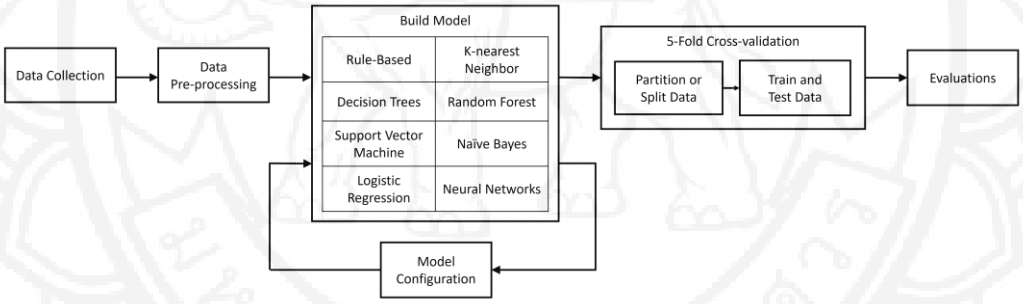

Figure 1 Process diagram for identifying suitable models

1. Data Collection

The first step was data collection. As rainfall is the primary factor behind floods and landslides in Thailand (Department of Water Resources, Ministry of Natural Resources and Environment, 2009), it is crucial to use a dataset related to rainfall. The Department of Water Resources in Thailand has been continuously collecting rainfall data from across the country and uploading it into the Early Warning System (http://ews.dwr.go.th/ews/index.php?page=warn_list_new.php). This study used the downloaded dataset, which contains 11 attributes and 13,992 rows. The collected data spanned from 2013 to 2022; however, it contained missing values that needed to be addressed.

2. Data Pre-processing

Before any model training could take place, the collected dataset was rearranged. It required pre-processing to replace missing values and remove outliers, inconsistencies, and irrelevant features that were not useful in the

process. First, the missing values were replaced using the K-nearest Neighbor imputation method. Next, outliers were identified by locating data points outside the whiskers of the boxplot represented by the small black diamond shapes as shown in Figures 2a and 2c. They were removed by applying the interquartile range method. After applying this method, each feature was plotted again as a boxplot, it revealed that no outliers were present anymore, as depicted in Figures 2b and 2d.

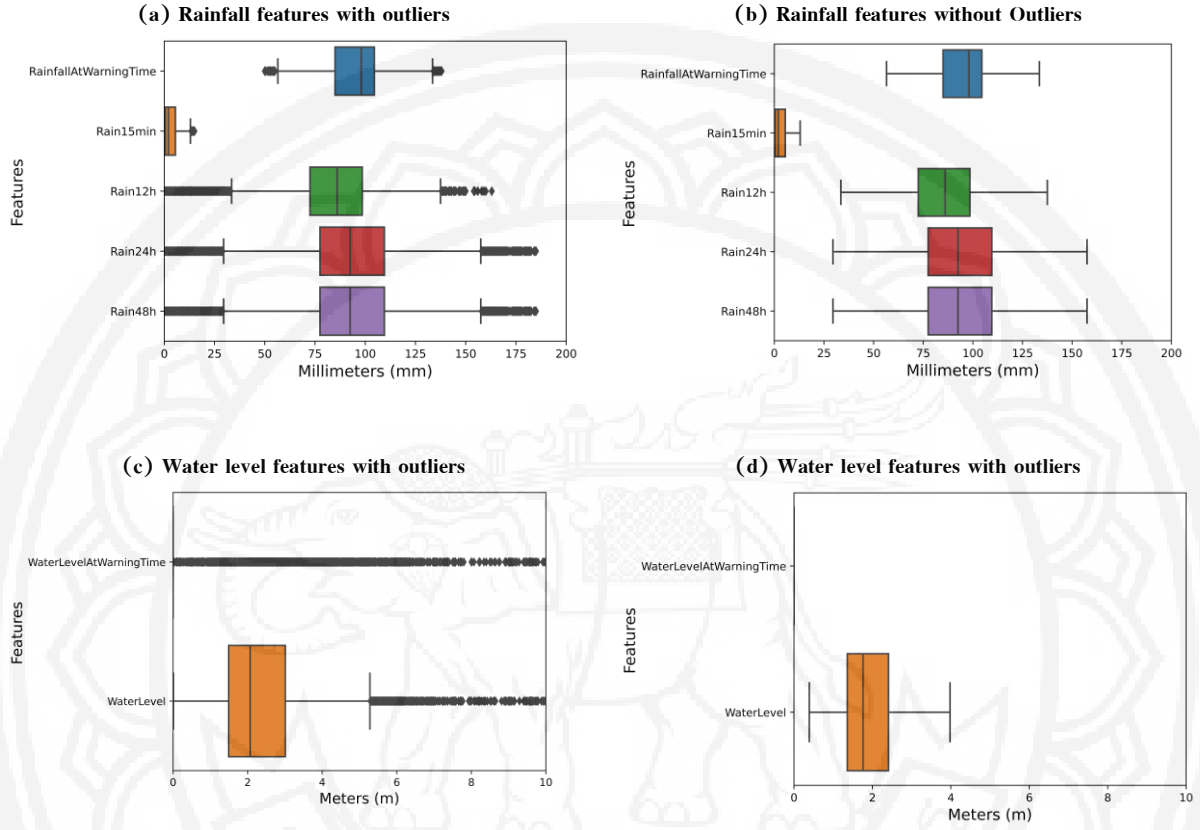

Figure 2 Boxplot of data features

Features that were not related to the amount of rainfall, such as Date, Time, and StationID were detected by checking the data type of each feature, if the data type did not correspond to the data type that is related to rainfall, then it is removed from the dataset. The dataset originally contained six warning types, which were divided into three classes for both rainfall and water level: vigilant, warning, and evacuate. After eliminating the outliers, only rainfall-related classes remained in the dataset. The warning types that remained in the dataset were, namely vigilant (0), warning (2) and evacuate (4). A violin plot (Figure 3) was created for each feature where each class was represented by a different color, blue, for "WarningType" 0, orange for "WarningType" 2, and green for "WarningType" 4. Violin plots were created to determine how the data was distributed for each class in each feature. The appearance of each shape in the violin plot represents the density of the data. The wide areas of the shape indicate a large concentration of data, or in other words, there were a large number of data points in those areas. In contrast, narrow areas meant that there were fewer data points. The density within each feature of a given class was also utilized to set rules or conditions in the Rule-Based model. The violin plot for the attribute "WaterLevelAtWarningTime" (Figure 3b) had a value of 0.0 for all warning types. Moreover, the

data concentration for the attribute "Rain15min" (Figure 3c) was similar across all three warning types, making it a challenge to determine a threshold. Therefore, these two attributes were eliminated from the dataset. The remaining attributes are seen in Table 1. Finally, further analysis of the dataset, discovered that the dataset was imbalanced (Acharya, Ghosh, Kang, Munasinghe, & Binita, 2022). This could affect the model's performance in prediction warnings. To solve this problem, an oversampling approach known as Synthetic Minority Oversampling Technique or SMOTE (Sreevidya, Abhilash, Paul, & Rejithkumar, 2021) was utilized to reduce bias in the model prediction. After pre-processing the data with multiple methods, 9,001 rows remained.

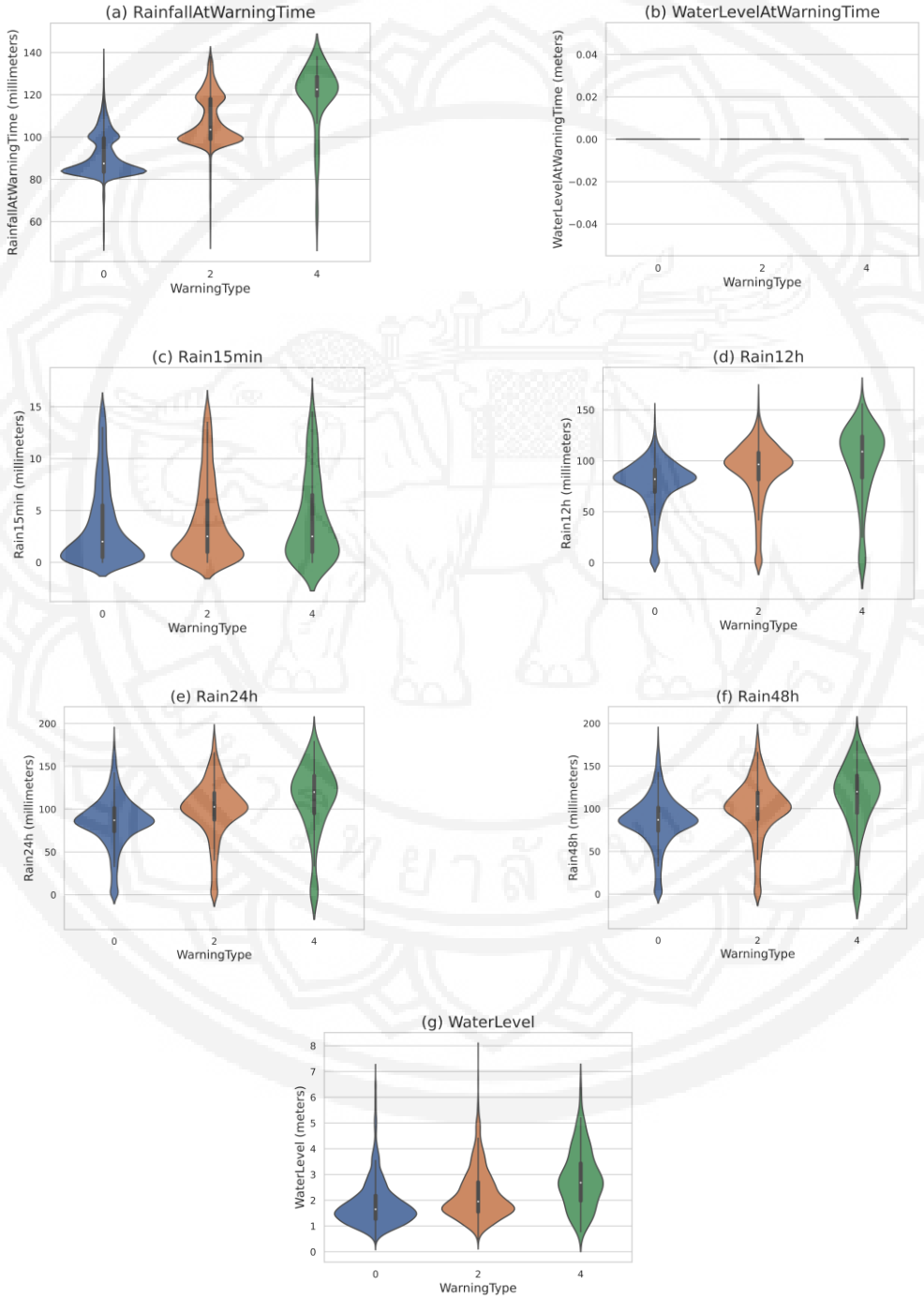

Figure 3 Violin plot for each feature

Table 1 Attributes in the dataset

Attributes	Types	Description
RainfallAtWarningTime	Float64	The accumulated rainfall in the past 12 hours at the time of warning.
Rain12h	Float64	The accumulated rainfall in 12 hours on the day of warning (7 p.m. of the day before warning until 7 a.m. of the day of warning).
Rain24h	Float64	The accumulated rainfall in 24 hours on the day of warning (7 a.m. of the day before warning until 7 a.m. of the day of warning).
Rain48h	Float64	The accumulated rainfall in 48 hours on the day of warning (7 p.m. of the 2 days before the day of warning until 7 a.m. of the day of warning).
WaterLevel	Float64	The level of water on the day of warning.
WarningType	Int64	The warning type for floods and landslides.

3. Building the Model

After pre-processing the data, the next step was to build the models that were used in training and testing the data. Eight models were chosen to evaluate the performance of predicting floods and landslides.

3.1 Rule-Based

Rule-Based is a model that uses conditional or if-else statements, also known as rules to execute a separate sequence of statements, depending on whether the expression is true. These rules are closely related between the inputs and the outputs of a problem. When an input is given, it is sent to be checked for which condition it meets, and an output is determined based on the condition that the input satisfies.

For the process of this model, the violin plot of each feature (Figure 3) was used to determine the criteria for the warning types. It was discovered that in the "RainfallAtWarningTime" feature (Figure 3a) for "WarningType" 4, the data mainly concentrated at approximately 120-130 mm. Based on this, the criterion for this "WarningType" was selected. Similarly, values above 95 were selected to classify "WarningType" 2, while values below that were the criterion for "WarningType" 0. After analyzing the features "Rain12h" (Figure 3d), "Rain24h" (Figure 3e) and "Rain48h" (Figure 3f), it was determined that from 110-120 mm onwards could be used to classify "WarningType" 4. For "WarningType" 2, values around 90 mm were chosen. Any values below that were selected to determine "WarningType" 0. For the "WaterLevel" feature (Figure 3g), values above 2 meters were utilized for "WarningType" 4, values above 1.5-2 meters for "WarningType" 2, and values below that for "WarningType" 0.

3.2 K-nearest Neighbor

K-nearest neighbor (KNN) is a supervised machine learning model used for classification and regression. It classifies a sample data point by calculating the distance using a distance metric or by looking at similarities between the K number of nearest data points surrounding it (Adnan et al., 2020; Chang et al., 2011; Cover & Hart, 1967).

Several parameters are used when creating this model, however, the number of nearest neighbors is the main hyperparameter that is frequently used to determine the model's best performance. In this study, the number of nearest neighbors was initialized at 5 and was then increased by 5 until it reached 50 (Table 2).

3.3 Decision Tree

The purpose of the Decision Tree model is to generalize or find patterns in the data by determining which tests best divide the instances into distinct classes, forming a tree (Kotsiantis, 2013). The tree represents the decision-making process. It asks a question and depending on the response (Yes or No), it further divides the tree into subtrees. Each internal node represents a feature in the dataset, each branch represents a decision rule, and each leaf node represents the outcome (Solehman, Azmi, & Setianingsih, 2019). There are two node types: Decision Nodes and Leaf Nodes. Decision Nodes have multiple branches and are used to make decisions. Leaf Nodes, on the other hand, represent the results of those choices and do not have any additional branches.

Two essential hyperparameters used in building a decision tree were the splitting criteria and the maximum depth (Table 2). The splitting criteria determine when and how to split a decision node. In this case, two splitting criteria were used, namely Gini and Entropy. The maximum depth is the maximum depth of the tree or the number of splits a tree can make before predicting the target. Initially, the tree had a maximum depth set at 5 which was then increased by 5 to determine whether a more complex tree would make a better prediction. Additionally, the splitting criteria were also switched between Gini and Entropy interchangeably.

3.4 Random Forest

Random Forest is another supervised machine learning model that is also used for classification. It consists of multiple decision trees that were merged using different subsets of data and features to help make better predictions (Acharya et al., 2022; Adnan et al., 2020) These predictions are made by computing the prediction for each tree and selecting the best outcome.

Four hyperparameters were used in this model (Table 2). The first is the criterion. This hyperparameter is a function used to determine when and how to split. It is also used to measure the quality of the split. The second is maximum features. It is used to determine the maximum number of features to be used in each tree in a random forest. The third feature is the number of estimators. It is used to assign the number of trees that will be used in a random forest. The fourth parameter is the minimum sample leaf, which sets the minimum number of samples in a leaf node after a split.

3.5 Multilayer Perceptron

A multilayer perceptron is a type of neural network comprised of three or more layers, namely the input layer, one or more hidden layers, and an output layer (Adnan et al., 2020). The input layer is responsible for receiving the data, while the hidden layers process and make the computation, and the output layer provides the result. The layers are composed of nodes where computation takes place. These nodes combine the input data with weights. In the training phase, the weights are modified using the backpropagation method so that the model can make precise predictions or classifications based on the input data.

Six hyperparameters were tuned. First was the number of neurons or nodes in each hidden layer. In this model, there were five layers, each layer consisted of 5, 15, 35, and 15 nodes, respectively. Second was the activation function for the hidden layer. Identity, logistic sigmoid, hyperbolic tan, and rectified linear unit were all the functions that were tuned for this hyperparameter. Third was the function for weight optimization, namely the stochastic gradient descent function, the limited-memory BFGS function, and the Adam Solver function. Fourth was the alpha value whose values were from 10^{-4} to 10^{-1}. An adaptive learning rate approach was used as the fifth hyperparameter and the final hyperparameter was the maximum number of iterations ranging from 100 to 600 (Table 2).

3.6 Logistic Regression

Logistic Regression is another supervised machine learning model that uses multivariate analysis for estimating or predicting whether an event will happen (Habibie & Nurda, 2022; Yi et al., 2019). It contains a dependent variable and independent variables. The dependent variable must be binary, meaning either having a value of 1 or 0. The independent variables may have one or more variables that can be categorical or continuous.

For this model, the two hypermeters solver and multiclass were used (Table 2). Solver determines the algorithm that will be used for optimization, while multiclass is applied for a multiclass problem. The solver values that were tested were the limited-memory BFGS, the stochastic average gradient descent, the saga and the newton-cg. The value tested for multiclass was multinomial because this paper deals with a multi-class problem.

3.7 Support Vector Machine

Support Vector Machine is a supervised machine learning model used for classification. It chooses the optimal line called a hyperplane to classify the data points by choosing the line that separates the data and which is the furthest away from the closest data points (Adnan et al., 2020; Chang et al., 2011; Yi et al., 2019).

In this model, the Support Vector Classification was used to classify the data. There are several hyperparameters used for this model, with only three hyperparameters, kernel, gamma and C (Table 2), being selected to be used in the model. The kernel is a function used to find the hyperplane, while gamma is used in non-linear kernels to determine the distance of the influence of a single training point. C is used to determine the penalty for a data point that was misclassified. The functions that were utilized for kernel were the linear, polynomial, and radial basis functions, while the gamma and C values ranged from 10^{-4} to 1000.

3.8 Naïve Bayes

The Naïve Bayes model is a probabilistic approach suitable for general classification expectations that use a series of probabilistic computations to find the best-fitted classification for the given data (Habibie & Nurda, 2022; Yang, 2018). Naïve Bayes also assumes that each data feature makes an independent and equal contribution to the outcome.

Gaussian Naïve Bayes model was used for this paper to determine the parameters best suited for the model to achieve the best performance (Table 2), the prior parameter values were adjusted from [0.3, 0.3, 0.4] to [0.3, 0.4, 0.3] to [0.4, 0.3, 0.3] to decide how much priors split to equal the test data that was appropriate. The variances smoothing was also considered as a hyperparameter in this model. It was initialized at 1e-9 and was adjusted until it reached 1e-1.

Each of these models was built, and then the features in the dataset were divided into independent features and dependent features, which were then split into an 80% training set and a 20% testing set. To ensure that each class or "WarningType" in the training set had an equal representation, an oversampling technique known as SMOTE was utilized. The training set was then trained in the model and the testing set was used for model prediction. The performance of each model was evaluated by creating a confusion matrix (Figure 4) and calculating the performance metrics such as recall, f1-score, and precision from the values in the confusion matrix. In the confusion matrix (Figure 4), there were 323 misclassified instances between classes 0 and 2, which is a significant number. The reason behind this was when the dataset was plotted using a pair plot, it was discovered that the majority of the data for classes 0 and 2 overlapped each other which affected the models' ability to predict warnings for these two classes correctly.

After evaluation, the hyperparameters of each model that were mentioned above were tuned to compare and achieve the best performance possible as seen in Table 2. These hyperparameters were applied in training and testing the models five times, each time with a different split of train and test set, to obtain the recall values for each round. The recall values varied in each of the training and testing procedures for the model because the distribution of each class differs in each split. This caused uncertainty as to which split would deliver the best performance. If the class distribution in a split was good, then the recall performance would also be good. On the other hand, if the class distribution in a split was poor, then the recall performance would also be poor. Therefore, 5-fold cross-validation was applied to each model in the next phase as a way to be certain of the performance of all the models.

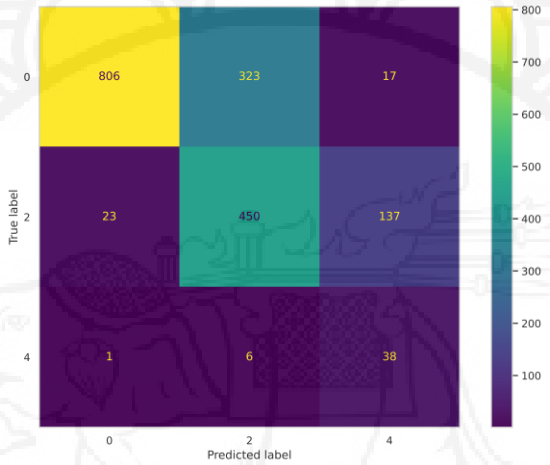

Figure 4 Confusion Matrix

Table 2 Summary of hyperparameter tuning for each machine learning model

Machine Learning Models	Hyperparameter	Tuning Range	Optimum
K-nearest Neighbor	Number of nearest neighbors	5,10,15,20,25,30,35,40,45,50	40
Decision Tree	Splitting criteria	Gini, Entropy	Gini
	Maximum depth	5,10,15,20	5
Random Forest	Criterion	Gini, Entropy	Gini
	Maximum features	Square root	Square root
	Number of estimators	25,50,100,150,200	100
	Minimum sample leaf	1,25,50,100,150,200	150
Multilayer Perceptron	Hidden layer sizes	(5,15,35,15)	(5,15,35,15)
	Activation function	Identity, logistic sigmoid, hyperbolic tan, rectified linear unit	Rectified linear unit
	Solver	Stochastic gradient descent, limited-memory BFGS, Adam Solver	Adam solver
	Maximum iteration	100,200,300,400,500,600	400
	Learning rate	Adaptive	Adaptive
	Alpha	10^{-4},10^{-3},10^{-2},10^{-1}	10^{-1}

Table 2 (Cont.)

Machine Learning Models	Hyperparameter	Tuning Range	Optimum
Logistic Regression	Solver	Limited-memory BFGS, Stochastic average gradient descent, saga, newton-cg	Newton-cg
	Multiclass	multinomial	multinomial
Support Vector Machine	Kernel	Linear, polynomial, radial basis function	Linear
	Gamma	10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100, 1000	10^{-3}
	C value	10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100, 1000	10^{-3}
Naïve Bayes	Variances smoothing	1e-9 – 1e-1	1e-4
	Priors	[0.3, 0.3, 0.4], [0.3, 0.4, 0.3], [0.4, 0.3, 0.3]	[0.3, 0.4, 0.3]

4. 5-Fold Cross-validation

Each model, except for Rule-Based, used the cross-validation function to provide an accurate performance for itself by partitioning the data into 5 folds equally and dividing it into 2 sets: train and test, at a ratio of 4:1, or 80% train data and 20% test data, respectively. By portioning and dividing the data this way, the possibility of overfitting can be avoided. There were 5 splits, meaning that every fold was trained and tested to see how well the model works for different inputs. In this phase, the optimal hyperparameters from the previous phase were chosen and an oversampling method was applied to the training fold to ensure an equal amount of data across all types of warning ("WarningType"). Cross-validation was then applied to each model and the average macro performance of precision, recall and f1-score were computed.

5. Model Evaluation

Each model was evaluated by calculating the macro average values of recall, f1-score, and precision. Recall calculates the model's capability to predict true values out of all actual values whereas f1-score is used to calculate the model's predictive performance which can be used to determine which model performed the best. Precision is the measure of how well the model predicts true values out of all the positive values. In this paper, recall is the most critical value as this work focuses on finding the most suitable model to use for predicting warning types for floods and landslides. These disasters affect people's lives which is why the recall value must be high for the credibility of predicting natural disasters. However, other values such as f1-score and precision are still important in analyzing the overall performance of the model.

Each model that was used in this study was evaluated using macro average performance metrics. The macro performance or scoring is used to measure the model's performance for multiclass classifications. The metrics were precision, recall and f1-score (Naviamos & Niguidula, 2020). The macro average was determined by calculating the precision, recall and f1-score for each class, or in this case the "WarningType", and then taking the average by dividing the metric values by the total number of classes. The model's performance is fairly evaluated across all classes, regardless of the imbalance in the dataset.

These metrics were computed using the values of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) from the confusion matrix. Precision assesses how well the model predicts true

values from all the positive values. Recall measures the model's ability to predict true values out of all other possible values. F1-score is used to identify the model's accuracy by taking the recall and precision values into account. The macro average of precision, recall, and f1-score is computed by the sum of each of these metrics in each round of training and testing. Then, it is divided by the number of classes.

The primary objective of this study is to identify the most effective models for predicting appropriate flood and landslide warning types. It is essential to achieve a high macro average recall value as these natural disasters have an immense impact on people's lives. In addition to this, other values such as the macro average f1-score, and macro average precision are also considered to evaluate the overall performance of the models. A comparison between no cross-validation and cross-validation was made to see how well each model performed.

Results and Discussion

After utilizing a dataset consisting of 9,001 rows to train and test each model, and the hyperparameters of each model were tuned with and without the 5-fold cross-validation, the performance for Rule-Based, Random Forest, Decision Tree, Multilayer Perceptron, K-nearest Neighbor, Logistic Regression, Support Vector Machine and Naïve Bayes were evaluated using macro scoring of recall, f1-score, and precision. The Rule-Based model had a macro recall score of 48%, a macro f1-score of 50%, and a macro precision score of 60%, respectively. Figure 5 displays the recall performance for when the models were trained and tested five times without the 5-fold cross-validation. It was discovered that each model yielded different recall values in each round because of the difference in the class distribution in each round of splitting. Figure 6 aims to compare the performance with and without the 5-fold cross-validation. Table 3 presents the results of the evaluation of each model. When the 5-fold cross-validation method was not applied, the recall values for each model were relatively close to each other, ranging from 64%-73%. This made it challenging to determine which models were suitable for prediction warnings for floods and landslides because these values only represented how well each model performed for a particular combination of data once. It did not represent the model's performance for the entire dataset. However, applying the 5-fold cross-validation method gave a more accurate performance for each model. The first four models in Table 3 displayed acceptable recall values ranging from 71%-74%, while the other three models still had recall values below 70%.

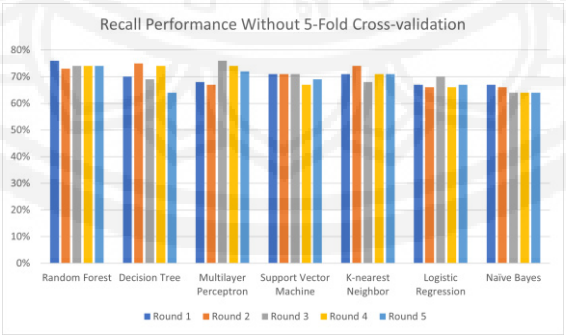

Figure 5 Recall performance without the 5-fold cross-validation in each round

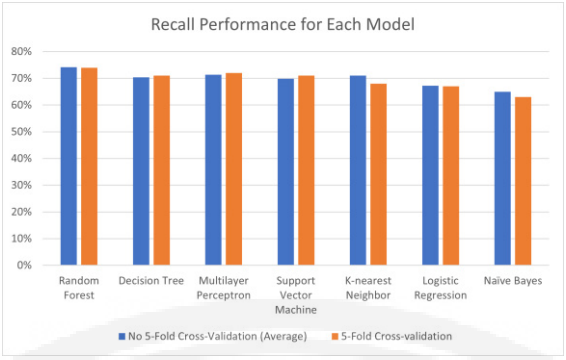

Figure 6 Recall performance for each model with and without the 5-fold cross-validation

Table 3 Macro average performance result of each model with and without 5-fold cross-validation

	No 5-Fold Cross-validation (Worst Values)			5-Fold Cross-validation		
Model	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Random Forest	**58%**	**73%**	**60%**	**58%**	**74%**	**58%**
Decision Tree	**54%**	**64%**	**55%**	**56%**	**71%**	**55%**
Multilayer Perceptron	**54%**	**67%**	**53%**	**55%**	**72%**	**56%**
Support Vector Machine	**52%**	**67%**	**51%**	**54%**	**71%**	**53%**
K-nearest Neighbor	55%	68%	55%	55%	68%	54%
Logistic Regression	52%	66%	52%	51%	67%	51%
Naïve Bayes	51%	64%	51%	50%	63%	50%

When the 5-fold cross-validation was applied, the recall performance for each model was better as it represented the performance of the entire dataset. The average recall performances for each model were computed from the values in each fold, therefore, it did not matter whether the classes were distributed equally in each fold or partition. As shown in Figure 6, the performance that was calculated with and without the 5-fold cross-validation was comparatively close to each other because when the cross-validation was used, the average performance for each metric was calculated automatically after training and testing. The results provided in Table 3 indicated that when data were randomly separated into training and testing sets without using 5-fold cross-validation, the training set might have had a particular combination of data that resulted in a good performance for the model, but it did not necessarily mean that the model performed well for the entire dataset. Therefore, 5-fold cross-validation was applied for more reliable performance. In 5-fold cross-validation, the data was divided into equal partitions and the model ran multiple times depending on the number of specified folds, each time using a different subset of data for training and testing. This ensured the model learned from different inputs of data. After training and testing five times, the average performance for each model was computed, resulting in a more accurate performance evaluation. As seen in Table 3, the results of using the 5-fold cross-validation revealed that all models performed relatively acceptably except for K-nearest Neighbor, Logistic Regression, Naïve Bayes and Rule-Based. This was the case because the first four models were complex and were able to capture the non-linear relationships in the dataset. However, the other models were simpler and had made certain assumptions about the dataset, for example, the Naïve Bayes model assumed that each feature was independent of the other and made an equal contribution to the output of the dataset, which was not the case. The findings for K-nearest

Neighbor, Logistic Regression, Naïve Bayes and Rule-Based were all below 70%, indicating that these models were not suitable for predicting this type of dataset. When analyzing the first four models that utilized the cross-validation technique in Table 3, it was found that their performance scores were acceptable as their recall values were at a high level, ranging from 71%-74, showing that these four models were appropriate for prediction warnings for floods and landslides. Although their values were appropriate, it is still low because of certain characteristics of the dataset, such as the correlation between features and the lack of data related to other factors causing floods and landslides. Geographical information system data and other factors including types of rock, the slope of the area, conditions of land use and land cover, and soil characteristics can assist in prediction warnings more precisely.

Conclusion and Suggestions

The objective of this research was to find the most suitable models for predicting three warning types of floods and landslides; vigilant (0), warning (2) and evacuate (4). The eight models for comparison were Rule-Based, K-nearest Neighbor, Decision Tree, Random Forest, Support Vector Machine, Naïve Bayes, Logistic Regression, and Multilayer Perceptron. The performance of the first four models, Random Forest, Decision Tree, Multilayer Perceptron, and Support Vector Machine, generally had high performance, but when 5-fold cross-validation was applied, a more reliable evaluation of the performance showed that the recall values of these four models were 71% or higher. This suggests that with further training and testing on a larger dataset, these models could be effective for predicting floods and landslides and allowing appropriate warnings to be issued. The detailed evaluation of the performance of the K-nearest Neighbor, Logistic Regression, and Naïve Bayes models, using the 5-fold cross-validation, concluded that these three models were not suitable for this problem as their performance was similar and low, and did not change.

However, there are some limitations in this research. The dataset used was not large enough, and therefore the values of the individual performance scoring of recall, f1-score, and precision were still not high enough. To improve these values, future work can focus on applying a bigger dataset consisting of other factors causing floods and landslides when training and testing the models or combining any of the first four models in Table 3 to form a hybrid model.

Acknowledgements

We would like to take this opportunity to express our sincere appreciation to two of our academic advisors, Associate Professor Dr. Annupan Rodtook and Dr. Chadaporn Keatmanee, for their invaluable guidance, support, and feedback throughout the research process. Thanks also to Mr Roy I. Morien of the Naresuan University Graduate School for his editing of the grammar, syntax and general English expression in this manuscript.

References

Acharya, V., Ghosh, A., Kang, I., Munasinghe, T., & Binita, K. (2022). Landslide Likelihood Prediction using Machine Learning Algorithms. *2022 IEEE International Conference on Big Data (Big Data)*, *17-20 December 2022* (pp. 5395-5403). Osaka, Japan: Institute of Electrical and Electronics Engineers. http:dx.doi.org/10.1109/BigData55660.2022.10020433

Adnan, M., Rahman, M., Ahmed, N., Ahmed, B., Rabbu, M., & Rahman, R. (2020). Improving Spatial Agreement in Machine Learning-Based Landslide Susceptibility Mapping. *Remote Sensing*, *12*, 3347. http:dx.doi.org/10.3390/rs12203347

Akter, T., Sadman, Y., & Bala, S. (2022). Use Machine Learning Technologies in E-learning. *2022 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, *1-4 June 2022* (pp. 1-4). Toronto, ON, Canada: Institute of Electrical and Electronics Engineers. http:dx.doi.org/10.1109/IEMTRONICS55184.2022.9795843

Chang, K.-T., Hwang, J.-T., Liu, J.-K., Wang, E.-H., & Wang, C.-I. (2011). Apply two hybrid methods on the rainfall-induced landslides interpretation. *2011 19th International Conference on Geoinformatics*, *24-26 June 2011* (pp. 1-5). Shanghai, China: Institute of Electrical and Electronics Engineers. http:dx.doi.org/10.1109/GeoInformatics.2011.5980950

Cover, T., & Hart, P. (1967, January). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21-27. http:dx.doi.org/10.1109/TIT.1967.1053964

Department of Disaster Prevention and Mitigation. (2022). *Popho phoei sarup sathiti satharanaphai nai rop pi 64 [DDPM reveals the summary of disaster statistics in the year 2021]*. Retrieved from http://relation.disaster.go.th/

Department of Water Resources, Ministry of Natural Resources and Environment. (2009). *Rai-ngan kansueksa wichai patchai siang lae withichiwit chumchon nai phuenthi siangphai din thalom lae nam lak - din thalom [Research report: risk factors and community lifestyles in landslide prone areas and floods-landslides]*. Bangkok: Department of Water Resources.

Habibie, M., & Nurda, N. (2022). Performance Analysis and Classification using Naive bayes and Logistic Regression on Big Data. *2022 1st International Conference on Smart Technology, Applied Informatics, and Engineering (APICS)*, 23-24 August 2022 (pp. 48-52). Surakarta, Indonesia: Institute of Electrical and Electronics Engineers. http:dx.doi.org/10.1109/APICS56469.2022.9918793

Inruang, W., & Chaipimonplin, T. (2558, January-December). The Prediction of Landslides Risk Areas in Uttaradit Province by Applying Geo-Informatics Technology with an Artificial Neural Network. *Journal of Social Sciences Srinakharinwirot University*, *18*(18), 191-207.

Jinsikhong, P., Musika, K., Amonlak, W., & Nonthakarn, D. (2022). Rabop chaeng tuean phai namthuam lae prap radap nam thang lot umong [Flood warning system and water level adjustment in the tunnel]. *The 10th Asia Undergraduate Conference on Computing : AUC2*, 24 February 2022 (pp. 2473-2474). Chon Buri: Kasetsart University Sriracha Campus.

Kotsiantis, S. (2013). Decision trees: a recent overview. *Artificial Intelligence Review*, *39*, 261-283. http:dx.doi.org/10.1007/s10462-011-9272-4

Martin, D., & Chai, S. (2022). A Study on Performance Comparisons between KNN, Random Forest and XGBoost in Prediction of Landslide Susceptibility in Kota Kinabalu, Malaysia. *2022 IEEE 13th Control and System Graduate Research Colloquium (ICSGRC), 23 July 2022* (pp. 159-164). Shah Alam, Malaysia: Institute of Electrical and Electronics Engineers. http:dx.doi.org/10.1109/ ICSGRC55096.2022.9845146

Naviamos, M., & Niguidula, J. (2020). A Study on Determining Household Poverty Status: SVM Based Classification Model. *3rd International Conference on Software Engineering and Information Management (ICSIM '20) 12-15 January 2020* (pp. 79-84). New York, NY, USA: Association for Computing Machinery. http:dx.doi.org/10.1145/3378936.3378969

Phomsak, W., & Khruahong, S. (2021). Rabop tuean fai pa attanomat duai theknoloyi lo ra [Automatic Forest Fire Alarm System with LoRa Technology]. *The 9th Asia Undergraduate Conference on Computing : AUCC, 25-26 February 2022* (pp. 1032-1038). Prachuap Khiri Khan: Rajamangala University of Technology Rattanakosin Wang Klai Kangwon Khet Campus.

Pomthong, S., & Asavasuthirakul, D. (2017). Kan wikhro phuenthi siangphai chak din thalom nai changwat phetchabun [Analysis of Landslide Risk Area in Phetchabun Province]. *Journal of Geoinformation Technology of Burapha University, 2*(3), 41-52.

Rangsiwanichpong, P. (2020). Application of Geographic Information System with the Probability of Landslide Model for Assessing Landslide Hazard in the Eastern Thailand. *The Journal of King Mongkut's University of Technology North Bangkok, 30*(4), 560-569.

Solehman, M., Azmi, F., & Setianingsih, C. (2019). Web-Based Flood Warning System Using Decision Tree Method. *2019 International Conference on Advanced Mechatronics, Intelligent Manufacture and Industrial Automation (ICAMIMIA), 9-10 October 2019* (pp. 1-6). Batu, Indonesia: Institute of Electrical and Electronics Engineers. http:dx.doi.org/10.1109/ICAMIMIA47173.2019.9223370

Sreevidya, P., Abhilash, C., Paul, J., & Rejithkumar, G. (2021). A Machine Learning-Based Early Landslide Warning System Using IoT. *2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE), 15-16 January 2021* (pp. 1-6). NaviMumbai, India: Institute of Electrical and Electronics Engineers. http:dx.doi.org/10.1109/ICNTE51185.2021.9487669

Thaiyuenwong, S., Nuthimthong, P., Detmak, W., & Soralump, S. (2013). Kan tuean phai din thalom chak kan truat wat pariman nam fon nai phuenthi phukhao thang phaknuea khong prathet thai [Landslide Warning by Rainfall Monitoring on Mountainous Area in Northern Thailand]. *The 18th National Convention on Civil Engineering, 8-10 May 2013* (pp. 401-406). Chiangmai: Chiang Mai University.

Wongka, M., Montri, W., & Chaikhamwang, S. (2022). Khrueang tonbaep chaeng tuean fai pa doi chai theknoloyi khrueakhai rai sai lo ra [Wildlife Alarm Model Using LoRaWAN Technology]. *The 10th Asia Undergraduate Conference on Computing : AUC2, 24 February 2022* (pp. 677-685). Chon Buri: Kasetsart University Sriracha Campus.

Yang, F.-J. (2018). An Implementation of Naive Bayes Classifier. *2018 International Conference on Computational Science and Computational Intelligence (CSCI), 13-15 December 2018* (pp. 301-306). Las Vegas, NV, USA: IEEE Computer Society. http:dx.doi.org/10.1109/CSCI46756.2018.00065

Yi, Y., Zhang, Z., Zhang, W., & Xu, C. (2019). Comparison of Different Machine Learning Models For Landslide Susceptibility Mapping. *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, *28 July 2019- 2 Augest 2019* (pp. 9318-9321). Yokohama, Japan: Institute of Electrical and Electronics Engineers. http:dx.doi.org/10.1109/IGARSS.2019.8898208

Yilmaz, E., Teke, A., & Kavzoglu, T. (2022). Performance Evaluation of Depthwise Separable CNN and Random Forest Algorithms for Landslide Susceptibility Prediction. *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, *17-22 July 2022* (pp. 5477-5480). Kuala Lumpur, Malaysia: Institute of Electrical and Electronics Engineers. http:dx.doi.org/10. 1109/ IGARSS46834.2022.9883954