Minute-scale Prediction of Soil Movement using Machine-Learning Techniques

K Agrawal[1], S Agrawal[2], P Chaturvedi[1, 3, *], N Mali[1,4], VU Kala[1,4], V Dutt[1]

[1]*Applied Cognitive Science Laboratory, Indian Institute of Technology, Mandi – 175005, India*

[2]*Centre for Converging Technologies, University of Rajasthan, Jaipur – 302004, India*

[3]*Defence Terrain Research Laboratory, Deference Research and Development Organization, Delhi – 110054, India*

[4]*Construction Material laboratory, Indian Institute of Technology Mandi – 175005, India*

**corresponding author Email: prateek@dtrl.drdo.in*

Abstract

Changes in the Earth's climate are likely to increase natural hazards like landslides in the hilly regions of north India. Thus, forecasting of these events at local-scale will help improve the preparedness of society in facing landslide disasters. There has been prior machine-learning research to predict landslide occurrence based on the statistical analysis of historical data and different triggering factors. While these attempts have shown promising results, these approaches have been limited to predicting landslides at a daily-scale. In this paper, we overcome the daily-scale limitation and focus on a minute-scale prediction of landslides by monitoring several soil and weather properties from a landslide site at Kamand, Himachal Pradesh. Data about temperature, humidity, rain, atmospheric pressure, light intensity, soil moisture, soil pressure, and soil movement were collected every 11-minutes from a landslide location on the Indian Institute of Technology Mandi campus at Kamand, Himachal Pradesh over a 10-day period in August 2017. The data contained a total of 842 instances to train several supervised machine-learning (ML) techniques. These included logistic regression, C4.5 decision tree, Naive Bayes, random forest and support vector machine with a non-linear polynomial kernel function. These models predicted soil movements as a binary class-problem, where the positive-class corresponded to soil movement, and the negative-class referred to no-movement. As the movement data had several instances of no-movement (732 instances) and a few cases of movement (110 instances; i.e., class-imbalance), accuracy was not a good measure of classification (classification accuracy is likely to be high due to the majority no-movement class). Thus, we assessed different ML techniques using metrics like the True Positive (TP) rate and False Positive (FP) rate. Results revealed that the C4.5 decision tree had the highest TP rate (= 61%) and a low FP rate (= 2%) among all algorithms. Thus, C4.5 decision tree algorithm performed best among the different classifiers. As part of our future research, we plan to explore some techniques to correct the class-imbalance in data and improve our current predictions. Additionally, since our data is a time series, we also plan to investigate time-series forecasting using traditional and deep-learning models in future.

Keywords: *Minute-Scale Prediction, Soil Movement, Machine-Learning, Landslides, True-positives, False-positives*

Introduction

Landslides cause a lot of damages to life and property, block roads, and disrupt the transportation of goods and services especially in the Himalayan Region of India (Chaturvedi, Shrivastava, & Kaur, 2017). For places at very high altitudes, where everything from food to clothing is imported from cities, blocking of roads due to landslides is a critical problem. Some of these reasons, including others, make landslide prediction a problem that needs to be addressed at the earliest.

Machine-learning (ML) techniques, i.e., techniques that enable computers to learn patterns in data have been gaining a lot of popularity across several real-world domains (Brenning, 2015). In fact, ML algorithms have recently been used in predicting landslides (Agrawal, Baweja, Dwivedi, Saha, Prasad, Agrawal, Kapoor, Chaturvedi, Mali, Kala, Dutt, 2017; Catani, Lagomarsino, Segoni, & Tofani, 2013). These attempts have not only been able to enhance the accuracy of prediction, but they have also made the interpretability of different factors involved in triggering a landslide much clearer (Catani *et al*. 2013). With the widespread use of ML algorithms and the advent of very high computational power, machine-learning techniques have become a more analytics-friendly tool compared to the physics- and geology- based traditional mathematical tools for predicting landmass movement (Agrawal *et al.* 2017).

Recent ML research (Pham, Bui, Pourghasemi, Indra, & Dholakia, 2017; Goetz, Brenning, Petschko, & Leopold, 2015; Bui, Pradhan, Lofman, & Revhaug, 2012) emphasized on predicting landslides at a daily-scale; however, little research has been done on predicting landslides at a minute-scale. Predicting landslides at a minute-scale is important as the minute-scale predictions help to warn people about impending landslides promptly. This real-time tracking can also be very helpful in knowing how active a site could be regarding its susceptibility to landslides. Furthermore, machine-learning algorithms could also help us understand the rate of change in site-specific soil and weather properties, which contribute to triggering of soil movement.

The primary goal of this paper is to predict site-specific soil-movement at the minute-scale by using traditional ML techniques. We use several ML algorithms like logistic regression (Brenning *et al.* 2005), C4.5 decision trees (Quinlan, 1986), Naïve Bayes (Pham *et al*. 2017), random forests (Breiman, 2001), and Support Vector Machines (Vapnik, 1998) for predicting soil-movement at minute-level. The data used in this study was collected using sensors deployed at one of the landslide-prone sites in Kamand, Himachal Pradesh. Since landslides are a rare phenomenon, the instances where soil-movements are recorded (positive class) are relatively smaller compared to

instances where soil-movements are not recorded (negative class). In such class-imbalanced datasets, accuracy maybe a misleading performance measure for evaluation (accuracy is likely to be high due to many instances of the negative class). Thus, we use more specific performance measures like the true-positive (TP) rate and the false-positive (FP) rate for evaluating the performance of different ML techniques.

In what follows, first, we provide a brief overview of the research that has been conducted on sensors for real-time monitoring of on-site soil and weather properties. This overview is followed by a description of traditional ML techniques that have been popularly used in literature. Next, we detail the study area and data collected from the study area using different sensors. Then, we provide a comparison of different ML techniques in accounting for soil-movement at a minute-scale at the study area. Finally, we close the paper by highlighting the implications of our results for predicting soil-movement at a minute-scale.

Previous Work

Prior research has used different methods for site-specific real-time monitoring of soil properties, soil movement, and weather (Ramesh, 2014). Some of these methods include visual interpretation of stereoscopic aerial photographs (Podolszki, 2014), satellite technology (Pham *et al.* 2017), unmanned aerial vehicles (UAVs) – based remote sensing (Neithammer, James, & Rothmund, 2012), digital-elevation models (DEMs) from airborne laser altimetry data (Mckean & Roering, 2004), and Brillouin optical time-domain reflectometry (BOTDR) (Zhang, Bin, & Hong-Zhoung, 2004). In India, several research organizations like Geological Survey of India, Central Building Research Institute, Defence Terrain Research Laboratory, and Amrita University have worked in the field of landslide monitoring and warning using sensors and systems for monitoring various soil and weather parameters (Kanungo, Maletha, Singh, & Sharma, 2017). However, the cost of these sensors and systems is presently very high, and the accuracy of these systems are unknown for minute-scale landslide predictions. These limitations restrict the large-scale deployment of current landslide monitoring sensors and systems in the real-world (Chaturvedi *et al.* 2017; McKean *et al.* 2004).

Furthermore, there have been several studies that have used certain state-of-the-art machine learning technique for predicting soil movements (Catani *et al.* 2013; Goetz *et al.* 2015; Mathew, Babu, Kundu, Kumar, & Pant, 2014; Pham *et al.* 2017). For example, one of the attempts has used machine-learning algorithms like Multilayer Perceptron, Functional Trees, and Naïve Bayes models for mapping susceptibility of 430 landslides locations using attributes like slope angle, slope aspect, elevation, and rainfall (Pham *et al.* 2017). Another attempt has shown that Random Forests, an

ensemble technique, performs better than other machine-learning techniques for landslide susceptibility mapping (Catani *et al.* 2013; Goetz *et al.* 2015). Furthermore, some researchers have used a logistic regression model for predicting the slope-failure initiation using the antecedent 30-day and 15-day rainfall (Mathew *et al.* 2014). This logistic-regression model is further validated through the Receiver Operating Characteristic (ROC) curve analyses using a set of samples which had not been used for training the classifier. The model showed an accuracy of 95.1% (Mathew *et al.* 2014). Subsequently, Agrawal *et al.* (2017) have predicted landslides on a daily-scale and have used several machine-learning algorithms along with class-imbalance correction techniques to improve the efficacy of their classifiers. While these studies prove that machine-learning techniques have shown promising results in predicting soil movements on a daily-scale, little research has taken place that investigates the problem of predicting landslides at a minute-scale. Minute-scale predictions are important to timely warn people about landslides.

In this paper, we use low-cost sensor technology for sensing different weather and soil parameters in real-time at a minute-scale. Furthermore, we investigate different ML techniques for predicting landslides in a minute-scale. As part of this study, we compare five different machine-learning algorithms that include logistic-regression (Brenning *et al.* 2005), C4.5 decision trees (Quinlan *et al.* 1986), Naïve Bayes (Tien Bui *et al.* 2012), random forests (Breiman *et al.* 2001), and support vector machines (Vapnik *et al.* 1998). We evaluate the performance of these algorithms using the standard 10-fold cross-validation technique, where data is randomly and repeatedly divided into non-overlapping training and test sets (Duda, 2014). The choice of these machine-learning algorithms is based upon their prior use for landslide predictions (Catani *et al.* 2013; Goetz *et al.* 2015; Mathew *et al.* 2014; Pham *et al.* 2017). As decision-tree algorithms have performed well at predicting landslides at a daily-scale (Catani *et al.* 2013; Goetz *et al.* 2015), we expected that the Decision Tree and Random Forest algorithms would perform well in predicting soil movement at a minute-scale. Also, decision-tree algorithms are much easier to interpret compared to other machine-learning techniques, and this feature makes them apt for understanding factors that contribute to triggering of soil movements.

Landslide site and data collection

The dataset used for this research has been collected from a landslide-prone hill located on the Indian Institute of Technology Mandi campus at Kamand, Himachal Pradesh (see Figure 1A). An initial site inspection revealed that a crack had started developing on the top of the hill at the selected site and a small section of the soil mass had started separating from the hill mass (see Figure 1B).

A B

Figure 1. Pictures of the selected landslide site at Kamand, Himachal Pradesh. A. Elevation view of the landslide site. B. Cracking of the soil at the top of the selected landslide. Different wired low-cost sensors can be seen deployed on the sliding soil mass beyond the crack.

To study the patterns of soil-movement at the site, we buried different sensors on top of the hill and data was collected by these sensors every 11-minutes. The system consisted of two types of sensors: surface sensors and buried sensors. The surface sensors included the following: temperature and humidity sensor, barometric-pressure sensor, light-intensity sensor, and a rain gauge. The buried sensors included the following: soil-moisture sensor, a force sensor, and an accelerometer. The soil-moisture sensor used the resistance property to measure water content in the soil surrounding its electrodes. Resistance is inversely proportional to soil moisture and output voltage. When the sensor was dry, a high value of resistance is recorded. Force sensor measured the pressure (in Newton) due to the internal pressure caused by soil and moisture. Temperature and humidity sensors measured temperature in °C and humidity in the percentage of water vapor in the air. The soil-moisture sensor measured the volume of water in the soil in a thin cylindrical volume surrounding the sensor probes. Similarly, light and pressure sensors sensed the induced light in lux and atmospheric pressure in kilo-Pascal (kPa). Rain gauge measured rain (in inches) on the site every 11-minutes. One of the sensors used in this study, i.e., the accelerometer was programmed differently. The accelerometer was programmed in such a way that whenever it recorded movement in the soil, the rate of change of the angular position (i.e., angular velocity, $\boldsymbol{\Omega}$) was measured. The accelerometer sensor reported values as a vector where the first three tuples corresponded to the three x-, y-, and z-axes acceleration components. Also, the next three tuples sensed non-zero angular velocity ($\boldsymbol{\Omega}$) along three axes ($\boldsymbol{\Omega}_x$, $\boldsymbol{\Omega}_y$, $\boldsymbol{\Omega}_z$). Whenever any soil-movement was observed, the angular rotations were summed in these three tuples ($\boldsymbol{\Omega}_x$, $\boldsymbol{\Omega}_y$, $\boldsymbol{\Omega}_z$). Every 11-minutes these tuples

were reset to record fresh accelerations and angular movements for a new 11-minute cycle. Every sensor used in the study has been calibrated and validated as per the field conditions (Mali, Chaturvedi, Dutt, Kala, 2017). The data collection at the site was done over a 10-day monsoon period between 11th August 2017 and 21st August 2017. The dataset contained 842 data points, where each point recorded different sensors values every 11-minute. We discuss the data-cleaning and preprocessing techniques in the next section followed by a brief description of machine-learning classifiers that we have used in this study.

Methodology

In this section, we describe the techniques that we used for cleaning the data before feeding it to the machine-learning classifiers. Next, we discuss the machine-learning algorithms used in this study. Finally, we mention the performance metrics used for evaluating the performance of different ML algorithms.

1. Data Cleaning

A validation process was run on the collected sensor data to validate the recorded rain accumulation, temperature, and humidity values. This validation was done from multiple weather websites as well as another local weather station installed at Kamand, Himachal Pradesh. These additional data sources helped us validate weather data collected from our sensors was accurate. Also, we performed proper calibration of buried sensors before installing them on site. The calibration ensured that the data reported by these sensors were accurate.

A machine-learning problem can typically be defined as a mathematical function which takes in input variables (independent variables) and outputs a decision variable (in our case, the decision is a value of "Yes (Y)" for soil-movement and a value of "No (N)" for no soil-movement). A data instance was labeled as 'Y' if any of the x-, y-, or z- angular velocities were non-zero. Thus, the decision variable that we used for classifying soil-movements can be mathematically expressed as:

$$\Omega_{tot} = |\Omega_x| + |\Omega_y| + |\Omega_z|$$

Where Ω_{tot}, is the decision variable. If $\Omega_{tot} \neq 0$, then we classified an instance as Y or soil-movement else we classified it as N or no soil-movement. Thus, the Ω_{tot} was the decision variable and all other sensed data like temperature, humidity, light-intensity, soil-moisture, and force were the independent variables that contributed in the formation of the machine-learning problem. It is important to note that accelerations, pitch, and roll were not taken as input variables to predict soil-

movement because these values are directly correlated to Ω_{tot} and the presence of these attributes may make different classifiers biased.

2. Machine-learning Algorithms

Here, we discuss different machine-learning approaches that have been successful in the past to predict landslides with higher accuracies. In this paper, we have compared several popular machine-learning techniques like logistic-regression (Brenning *et al.* 2005), C4.5 decision tree (Quinlan *et al.* 1986), Naive Bayes (Tien Bui *et al.* 2012), random forest (Breiman *et al.* 2001), and support vector machine with a non-linear polynomial kernel function (Vapnik *et al.* 1998).

Logistic regression has been particularly used in modeling landslides as it provides a probability of landslide occurrence against every data point using the logit model (Brenning *et al.* 2005). This algorithm has been widely used in landslide susceptibility mapping (Mathew *et al.* 2014). A decision tree is a hierarchical model composed of decision rules that recursively split independent variables into zones such that each maximum time balance in each split is achieved (Quinlan *et al.* 1986). The advantage of decision trees is that they can handle categorical as well as numeric variables and can incorporate them without strict assumptions on data (Tien Bui *et al.* 2012). In this study, we have used the J48 algorithm which is a Java implementation of the C4.5 algorithm (E. Frank, Hall, & Witten, 2016). The C4.5 uses an entropy-based measure as the attribute selection criteria on the tree nodes, and it is the same as the ID3 algorithm (Quinlan *et al.* 1986). Given a training dataset T with subsets T_i, i = 1,2,...,s, the C4.5 algorithm constructs a decision tree using the top-down and recursive-splitting technique starting with attributes with the maximum gain (Quinlan *et al.* 1986).

A Naïve Bayes (NB) classifier is a classification system based on Bayes' theorem that assumes that all the attributes are fully independent and give the output class, called the conditional independence assumption (Tien Bui *et al.* 2012). The main advantage of the NB classifier is that it is very easy to construct without needing any complicated iterative parameter estimation schemes (Tien Bui *et al.* 2012). In the case of NB classifier, the probability is first calculated for each output class (Y, N), and the classification is then made for the class with the largest posterior probability.

Random forest (RF) is an ensemble technique that utilizes many classification trees (a ‘forest’) to stabilize the model predictions (Breiman *et al.* 2001). The RF algorithm exploits random binary trees which use a subset of the attributes through bootstrapping techniques: From the original dataset a random selection of the attributes is performed and used to build the model, the data not included is referred to as “out-of-bag” (OOB) (Breiman *et al.* 2001). Each tree is developed to minimize classification errors; but, the random selection influences the results, making a single-tree

classification very unstable. For this reason, the RF method makes use of an ensemble of trees (the so-called "forest") thereby ensuring model stability (Breiman *et al.* 2001). The RF algorithm has been used in landslide predictions domain and susceptibility modeling by several studies (Goetz *et al.* 2015; Catani *et al.* 2013).

Support Vector Machine is a supervised learning method based on statistical learning theory and the structural risk minimization principle (Vapnik, 1998). Using the training data, SVM implicitly maps the original input space into a high-dimensional feature space. Subsequently, in the feature space, the optimal hyperplane is determined by maximizing the margins of class boundaries. We chose a non-linear polynomial kernel function in this paper since it has outperformed other kernels in prior research (Vapnik, 1998).

While each of these machine-learning algorithms could be used with a variety of settings and procedures for model selection, we chose configurations that we have considered typical based upon prior applications. All techniques mentioned above were run in the Java-written Weka package with default parameter settings and using a 10-fold cross-validation approach (Frank *et al.* 2016; Duda, 2004).

3. Analysis Methodology

Accuracy is the most straight-forward way to describe the performance of classifiers. It is defined as the ratio of instances (both positive and negative) correctly classified by the total number of instances present in the dataset. However, accuracy can be misleading in predicting natural hazards like landslides (Batista, Prati, & Monard, 2004). That is because soil-movement (landslide) occurrence is a rare phenomenon. This property makes landslide-prediction a class-imbalanced problem. In our study, the distribution of the two classes, i.e., Y and N, are 13% and 87%, respectively. If a trained classifier is biased towards the N class and labels each instance as belonging to the N class, then the classifier's accuracy would be 87%. As a classifier may not accurately predict the Y class and still may have a high accuracy, we used more specific performance measures like true-positive (TP) rate and false-positive (FP) rate to compare different classifiers (Batista *et al.* 2004). The TP rate is the percentage of landslide instances correctly classified by the classifier as landslides, and the FP rate is the percentage of no-landslide instances that are classified as landslides by the classifier. Thus, it is desirable for a classifier to possess a high TP rate and a low FP rate. In the next section, we present the results from different classifiers using a ten-fold cross-validation approach.

Results

In this section, we report the results of each classifier and their comparison incorrectly predicting the soil-movements in the dataset.

Table 1 shows the ten-fold cross-validated results from different classifiers. We can observe that the highest accuracy was obtained for C4.5 decision tree followed by logistic regression. Regarding interpretability, C4.5 decision tree is a very user-friendly technique as we can print the decision tree to see the attributes (in levels) that were picked up by the algorithm for classifying the data. Other classifiers also produced a high accuracy; however, accuracy is likely a biased measured due to the class-imbalance present in the dataset. Thus, next, we evaluated the TP and FP rates to compare different classifiers.

Table 1**:** Ten-fold cross-validation comparisons of different classifiers on the landslide dataset

Classifier	Accuracy	True Positive Rate	False Positive Rate
Logistic Regression	92.16	0.48	0.01
C4.5 Decision Tree	92.87	0.61	0.02
Naïve Bayes	91.45	0.53	0.28
Random Forests	89.07	0.17	0.01
Support Vector Machines	90.26	0.27	0.001

Table 1 shows that the C4.5 decision tree had the highest TP rate of 0.63 followed and a moderately low FP rate 0.02. This result indicated that C4.5 decision tree is 63% of the times correct in predicting soil-movement and at the same time has a relatively low a false-alarm rate. Such a combination is desirable in the landslides prediction domain as we want both the true-positive rate to be high and the false-positive rate to below. The Naïve Bayes algorithm had the second highest TP rate of 0.53; however, with a relatively high FP rate of 0.28. These results suggest that while the Naïve Bayes algorithm accurately predicted the soil-movement more than 50% times, it misclassified no-landslides as landslides in 28% of the instances. Lastly, both Support Vector Machines and Random Forests had very low FP and TP rates, which indicated that almost all the instances in the dataset were classified as belonging to the N class by these classifiers.

C4.5 Decision Tree

Figure 2 shows the resulting C4.5 decision tree for predicting soil-movement in the data set. As can be seen in Figure 2, the decision tree suggested that the primary attribute for distinguishing movement class from the no-movement class was the rain recorded in the last 11-minutes. This result shows that rain was one of the primary reasons for triggering soil movements at the chosen location. On descending further one can observe that force (pressure due to moist soil), humidity,

temperature, and time were other important attributes in the decreasing order of importance. Interestingly, other variables like light and soil-moisture did not enter the decision tree. Thus, their variables did not influence the soil movement as much compared to other attributes.

To understand the structure of the tree, we used a depth-first search approach wherein we descended along one path of the tree and inferred each node as we proceed. Thus, if no rain has been recorded and if no force has been recorded, then the movement of soil is unlikely. In contrast, if the force is non-zero, but rain is zero, then humidity is used as a splitting attribute. One can observe in the structure that force occurs three times along the route with different critical thresholds and each successive threshold is always greater than the preceding one. This result is notable as it may suggest that three different forces can attribute to three different magnitudes of soil-movement, i.e., no-movement, moderate movement, and severe movement extending the scope for future research, i.e., a multi-class problem. Furthermore, higher thresholds of humidity and temperature along this path suggest that soil-movements were more likely when the temperature and humidity were higher than 26ºC and 69%, respectively. This result indicated that soil-movements were more abundant in higher levels of relative humidity and temperatures. Finally, time was accounted as a factor for splitting the dataset along this route, where the threshold was 3:15 am. This result suggested that movements were more likely to occur after 3:15 am than earlier on the site.

On traversing the tree from the top towards the right branch, one can observe that force (pressure exerted by moist soil) was an important attribute in the tree. A non-zero force coupled with non-zero rain was indicative of soil-movement as per this route. Finally, if the force was non-zero, then time was used as the final attribute. The critical time calculated by the C4.5 algorithm along this route was 2 pm, i.e., soil-movements were likely to occur if the time of the day was greater than 2 pm than otherwise.

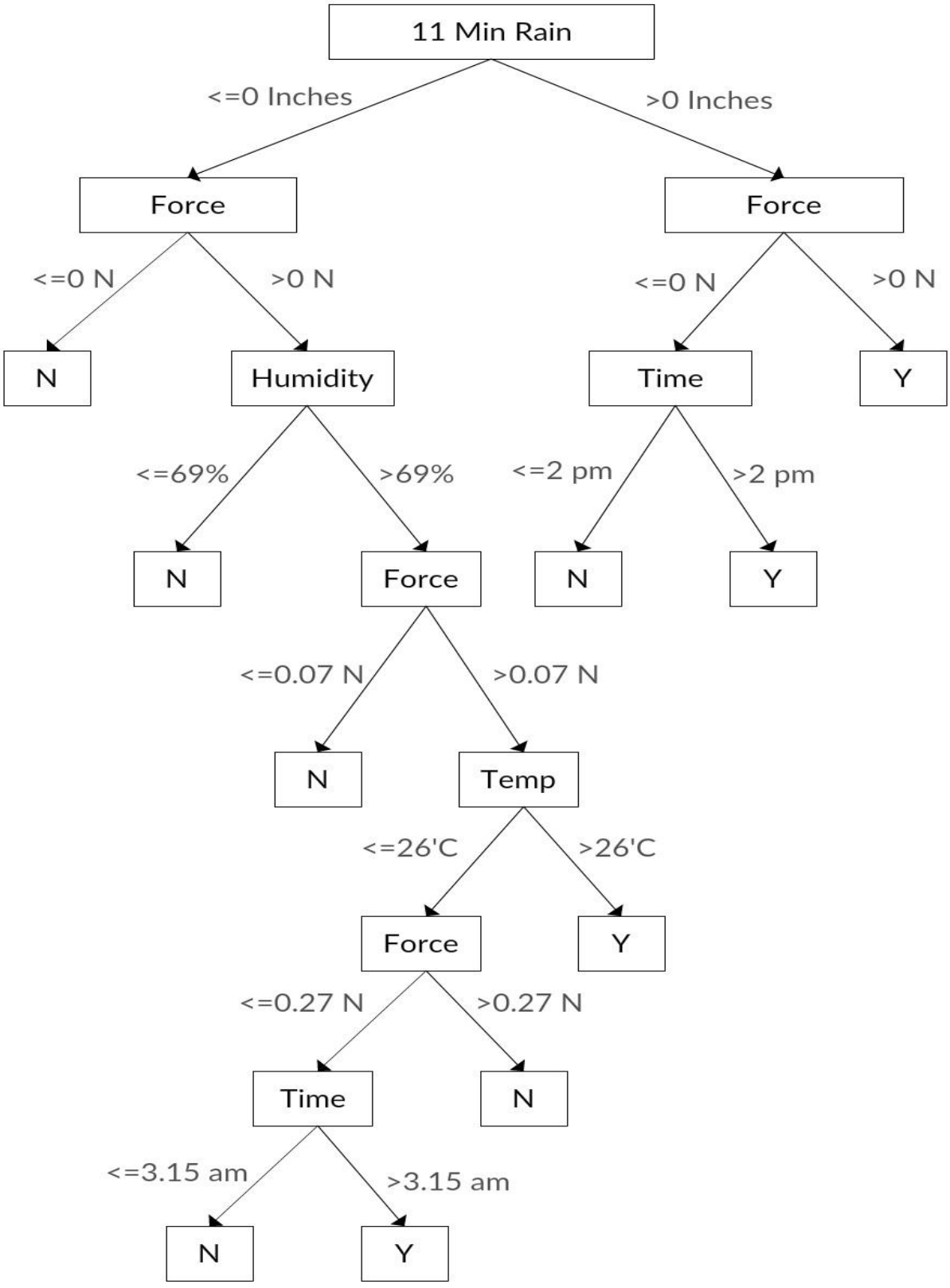

Fig 2: Decision Tree produced by C4.5 algorithm with Ten-Fold Cross-Validation

Discussion and Conclusions

Till recently, machine-learning (ML) techniques had been used to predict landslides on a daily scale (Goetz *et al.* 2015; Catani *et al.* 2013**).** In this paper, our primary goal was to try different ML algorithms to make minute-scale predictions for soil-movements at a landslide site at Kamand, Himachal Pradesh. We compared and evaluated the performances of five different ML algorithms that have proven to work well in prior research predictions (Pham *et al.* 2017; Goetz *et al.* 2015; Mathew *et al.* 2014; Catani *et al.,* 2013). We observed that C4.5 decision tree algorithm outperformed other machine-learning techniques in predicting soil-movements at the minute-scale. This result agrees with prior literature, where non-parametric algorithms like Random Forest and C4.5 decision tree had performed accurately for daily-scale prediction of landslides (Pham *et al.* 2017; Goetz *et al.* 2015).

In the C4.5 decision tree algorithm, we found that rain and force (pressure due to moist soil) were listed as the top decision attributes in splitting the movement and no-movement classes. A non-zero force and non-zero rain were predicted movement class; whereas, both zero rain and zero force were predicted as a no-movement class. This result may seem primitive; however, it is important as it confirms our key hypothesis that rainfall and soil-pressure were relevant indicators of soil-movements. Furthermore, force occurred at three different levels of the tree and, on each succeeding level, the critical threshold of force was greater than the preceding one. Although we can only speculate currently, this result perhaps indicates that the three different levels of soil-pressure thresholds likely correspond to the three different magnitudes of soil-movements occurring at the site. This speculation needs to be tested as part of future research. Lastly, two attributes namely light-intensity and soil-moisture did not enter the decision tree. This result indicates that these sensed values were not important for evaluating soil-movements on a minute-scale at the selected site compared to the other attributes. One explanation for this discrepancy could be the presence of consistent seepage of water from other internal sources along the hill at the chosen site (this water seepage was revealed upon a preliminary inspection of the site). This internal seepage of water could result in consistently high moisture-values irrespective of the moisture added by rainfall at the site. Another explanation could be that the rainfall attribute, which measured the rainfall falling at the site, accounted for soil-moisture indirectly as well. While light-intensity may not be a relevant factor for evaluating soil-movements, however; soil-moisture, as per our expectation and prior research, could become a key factor in determining soil-movements at other locations. This factor needs to be investigated more thoroughly as part of future research.

Furthermore, this study also shed light on the methodology to follow while evaluating the performance of different machine-learning classifiers in real-world data sets involving class

imbalance. In cases of class imbalance, the accuracy is likely to be high, and one needs to rely upon specific performance measures like true-positive rates and false-positive rates for evaluating machine-learning algorithm's performance. These measures enable us to check the performance of classifiers across both negative and positive instances of a binary classification problem.

There are several ideas as part of future research that could help us improve our current results. First, we plan to deploy our sensors on several other landslide-prone sites beyond the one selected in this paper. We then plan to use the C4.5 trained decision tree on different datasets to measure the robustness of the algorithm. Subsequently, we plan to integrate site-specific different data sets collected from different hills and combine them into a unified dataset. This unified dataset could have several soil properties like texture, structure, pore space, and consistency. Also, these properties could be supported by other properties like local weather, lighting, soil-moisture, and force.

Second, as part of our future research, we would like to emphasize on evaluating time-series forecasting techniques like the Auto-Regressive Moving Average (ARIMA) model (Khashei & Bijari, 2011) and recurrent neural networks like Long-Short Term Memory (LSTM) models (Mikolov, Karafiát, Burget, Cernocký, & Khudanpur, 2010) for soil-movement predictions. ARIMA models have been widely used in financial forecasting where the data is a time-series, like our landslide dataset (Khashei & Bijari, 2011). Similarly, LSTM models, which keep a record of the memory of events and how this memory affects the current predictions, have been used in predicting health outcomes (Mikolov *et al.* 2010; Kaushik, Choudhury, Dasgupta, Natarajan, Pickett, & Dutt, 2017). Furthermore, we would also like to use more sophisticated performance measures for model comparison as part of our future research. Measures like Area under the Receiver Operator Characteristics (ROC) curve (Japkowicz & Stephen, 2002) and sensitivity-index (d') (Macmillan & Creelman, 2010) may provide alternate performance measures for comparing the performance of different ML techniques.

References

[1] S. L. Gariano, F. Guzzetti, 2016, Landslides in a changing climate, Earth-Science Reviews, 162, pp 227-252.

[2] B. T. Pham, D. T. Bui, H. R. Pourghasemi, P. Indra, M. B. Dholakia, 2017, Landslide susceptibility assessment in the Uttarakhand area (India) using GIS: a comparison study of

prediction capability of naïve Bayes, multilayer perceptron neural networks, and functional trees methods, Theoretical and Applied Climatology, 128(1-2), pp 255-273.

[3] J. N. Goetz, A. Brenning, H. Petschko, P. Leopold, 2015, Evaluating machine learning and statistical prediction techniques for landslide susceptibility modeling, Computers & Geosciences, 81, pp 1-11.

[4] J. Mathew, D. G. Babu, S. Kundu, K.V. Kumar, and C. C. Pant, 2014, Integrating intensity–duration-based rainfall threshold and antecedent rainfall-based probability estimate towards generating an early warning for rainfall-induced landslides in parts of the Garhwal Himalaya, India, Landslides, 11(4), pp 575-588.

[5] A. Brenning, 2005, Spatial prediction models for landslide hazards: review, comparison, and evaluation, Natural Hazards and Earth System Science,9 vol. 5, no. 6, pp. 853-862.

[6] J. Quinlan, 1986, Induction of decision trees, Machine Learning, vol. 1, no. 1, pp. 81-106.

[7] L. Breiman, 2001, Random forests, Machine Learning, vol. 45, no. 1, pp. 5-32.

[8] Tien Bui, B. Pradhan, O. Lofman, I. Revhaug, 2012, Landslide Susceptibility Assessment in Vietnam Using Support Vector Machines, Decision Tree, and Naïve Bayes Models, Mathematical Problems in Engineering, vol. 2012, pp. 1-26.

[9] V. Vapnik, 1998, Statistical learning theory. New York: J. Wiley.

[10] G. E. Batista, R. Prati, M. Monard, 2004, A study of the behavior of several methods for balancing machine learning training data, ACM SIGKDD Explorations Newsletter, vol. 6, no. 1, p. 20.

[11] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, S. Khudanpur, 2010, Recurrent neural network based language model, In Interspeech, 2, pp 3.

[12] M. Khashei, M. Bijari, 2011, A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. Applied Soft Computing, 11(2), pp 2664-2675.

[13] F. Catani, D. Lagomarsino, S. Segoni, V. Tofani, 2013, Landslide susceptibility estimation by random forests technique: sensitivity and scaling issues, Natural Hazards and Earth System Science, vol. 13, no. 11, pp. 2815-2831.

[14] L. Podolski, 2014, Stereoscopic analysis of landslides and landslide susceptibility on the southern slopes of the Medvednica Mt, Retrieved from http://hrcak.srce.hr/file/219540

[15] U. Niethammer, M. R. James, S. Rothmund, 2012, UAV-based remote sensing of the Super-Sauze landslide: Evaluation and results, Engineering Geology, 128, 2-11.

[16] D. Zhang, S. H. I. Bin, X. Hong-Zhong, 2004, Experimental study on the deformation monitoring of reinforced concrete T-beam using BOTDR, Journal of Southeast University (Natural Science Edition), 4, 12.

[17] J. McKean, J. Roering, 2004, Objective landslide detection and surface morphology mapping using high-resolution airborne laser altimetry, Geomorphology, 57(3), 331-351.

[18] P. Chaturvedi, S. Shrivastava, P. Kaur, 2017, Landslide Early Warning System Development using Statistical Analysis of Sensors' Data at Tangni Landslide, Uttarakhand, India, Advances in Intelligent Systems and Computing, Springer International Publishing, 547

[19] D. P. Kanungo, A. K. Maletha, M. Singh, N. Sharma, 2017, Ground-Based Wireless Instrumentation and Real-Time Monitoring of Pakhi Landslide, Garhwal Himalayas, Uttarakhand (India), In *Workshop on World Landslide Forum*, pp. 293-300

[20] M. V. Ramesh, 2014, Design, development, and deployment of a wireless sensor network for detection of landslides. *Ad Hoc Networks, 13*, 2-18.

[21] E. Frank, M. Hall, and I. Witten, 2016, Data Mining: Practical Machine Learning Tools and Techniques, 4th ed. Morgan Kaufmann.

[22] R. Duda, 2004, Pattern Classification 2nd Edition with Computer Manual 2nd Edition Set. John Wiley & Sons.

[23] N. Macmillan, C. Creelman, 2010, Detection theory. New York, NJ [u.a.]: Psychology Press.

[24] N. Mali, P. Chaturvedi, V. Dutt, V. U. Kala, 2017, Training of Sensors for Early-Warning System of Rainfall-Induced Landslides, Proceedings of 19th International Conference on Soil-Mechanics and Geotechnical Engineering, Sydney, Dec. 2017 (In Press).

[25] S. Kaushik, A. Choudhury, N. Dasgupta, L. Pickett, V. Dutt, 2017, A study of Statistical and Predictive Analysis of US Pain Medications. In International Conference on Machine-Learning and Data Science, IEEE Conference (In Press).

[26] K. Agrawal, Y. Baweja, D. Dwivedi, R. Saha, P. Prasad, S. Agrawal, S. Kapoor, P. Chaturvedi, N. Mali, V.U. Kala, V. Dutt, 2017, A Comparison of Class Imbalance Techniques for Real-World Landslide Predictions. In International Conference on Machine-Learning and Data Science, IEEE Conference (In Press).