Referee #1 (Samuele Segoni)

Dear Authors, thank you very much for your careful answers and revisions. I think you did an excellent job in addressing my concerns and the manuscript is almost ready for publication. There is just one small issue pending, probably because I didn't explain myself clearly in my previous comment.

This is the original comment:

"- I understand that you train the ANN with 70% of 144 triggering events and 70% of 47398 events. Doesn't it lead to an unbalanced prediction? ANN will be trained to detect non triggering conditions more effectively than triggering conditions."

What I meant is that when you train an ANN (or other machine learning algorithms) usually you input the same number of positive cases (triggering rainfall events, in your work) and negative cases (non triggering events). This is a fundamental assumption to have a balanced prediction. If I understood correctly, you trained your ANN with 101 triggering events and 33179 non-triggering events. First of all, I was asking you to confrm that my understanding is correct. Second, If my understanding of the procedure was right, I was asking to discuss the problem possibly induced in the ANN: given the bias in the training dataset (negative events several orer of magnitude more numerous than positive events), maybe the prediction is biased accordingly? I agree with you that the tuning of the false alarms(missed alarms or the definition of weights are out of the scopes of the work, but this is one more reason to get a prediciton that is perfectly balanced between false alarms and missed alarms.

After this issue is clarified, the manuscript could move to the typesetting and publication stages.

Best regards

Reply: We thank Dr. Samuele Segoni for reassessing our manuscript. We thank him also for mentioning the issue of imbalanced training data. Landslides are natural hazards, and as such, they are rare in comparison to rainfall events. In our case, we have around the 3/1000 triggering events/total number of events, which makes our data set very imbalanced.

However, this issue can be solved by using the appropriate ROC performance metric (e.g., the True skill statistic), as we did in our case.

Indeed, common ANN training software often use the ROC accuracy index by default

$$ACC = \frac{TP+TN}{TP+TN+FP+FN},$$

which is the ratio of correct predictions respect to the total number of data. With this metric imbalanced training data is an issue, as ACC is near to 1 (= 0.9970), for a triggering threshold higher than all the rainfall event data, i.e. with the following confusion matrix (where P = number of positives, i.e. triggering events, and N = number of negatives, i.e. non-triggering events):

Tab.R1

TP = 0	FP = 0
FN = P	TN = N

If we would have trained the ANN maximizing the ACC it would have been dominated by the non-triggering events. However, it is easy to see that with the same confusion matrix, the true skill statistic is instead:

TSS = TP/P 0/P – 0/N = 0

which has the meaning of a “random guess”. This occurs because the TSS weights separately triggering and non-triggering events, and thus it is not subject to the issue of imbalanced training data.

We can make another example with the data of the paper. If one takes, for instance, the results of Tab. 1, first row (ANN model based only on Duration), one can compute, based on TPR = 0.74, FPR = 0.44, an accuracy ACC = (107 + 26543)/(144 + 47398) = 0.5606, which is way lower than the ACC = 47398/(144 + 47398) = 0.9970, corresponding to the dreaded situation of Tab.R1. This confirms that ACC is not the appropriate performance metric for the aim of developing landslide triggering thresholds.

Thanks to the referee we had also the chance to improve the terminology, which was a bit loose (we used “importance”, now we use “utility” and “loss”, which are more correct terms), and to add some text on the issue of imbalanced training datasets (Line numbers are those of the tracked-changes manuscript):

LL 126-129: We then identify the mentioned threshold value by maximizing the TSS, which has the advantage to do not be affected by unbalanced training dataset issues respect to other indices, such as ROC accuracy ACC = (TP + TN)/(TP + TN + FP + FN) – a performance metric used as default by many ANN training software tools. Maximization of TSS implicitly assumes that all entries of the confusion matrix have the same utility. Quantifying the loss of a false negative respect to a false positive, is a complex task that goes beyond the aim of the present analysis, and that has been faced only in very recent studies.

Referee #3

Dear authors, I carefully read your revised manuscript, and I think that you fully considered all the points highlighted by reviewers. I think also that the research is ready for publication.

Anyway, some details remain to be improved in order to propose a good-quality work:

Reply: We thank the referee for the quick and careful review, which helped us to further improve our manuscript. We thank the referee also for the corrections provided in the annotated PDF, which were all accepted. We took the chance to make a general improvement of the manuscript (see track-changes file).

Please, consider the following suggestions:

Figure 1: improve again the quality of the image, better showing landslides location using an adequate symbology (improve dots dimension for example). The legend seems also to be slightly out of focus.

Reply: Thank you. We have improved Figure 1, starting from your suggestions.

Figure 2: I strongly recommend to add figure reference in the text, and to better explain figure 2b. Otherwise, you propose a detailed image avoiding a corresponding exhaustive explanation for the reader. For example, what does "Hidden layer" stands for? Similarly, what does the "Dichotomization" term stand for? Actually, it is not fully clear reading the text. Please improve Figure 2 explanation.

Reply: Thank you, we have added some text explaining better the content of Fig. 2b (Line numbers are those of the tracked-changes manuscript):

L 105-107: The neural network, characterized by a feed-forward structure, is composed of three layers: input, hidden and output (Fig 2b). The input layer takes the series of predictors and sends them to the hidden layer, where the series are combined and transformed though a specific activation function.

L 125 -126: The output of the ANNs is transformed into a binary code (dichotomization), by assuming a value of 1 (ANN predicts a landslide) when the output is greater that a threshold value, and of 0 otherwise.

Thank you