

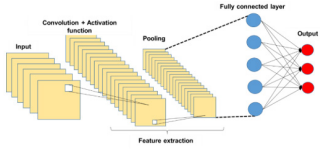
Summary of the Changes to Reviewer 2's Recommendations and Comments

Journal: Natural Hazards and Earth System Sciences

Ref: NHESS-2024-86

Title: Predicting Deep-Seated Landslide Displacements in Lushan Mountains through the Integration of Convolutional Neural Networks and an Age of Exploration-Inspired Optimizer

The authors appreciate the reviewer's valuable feedback. The summary of the changes based on the reviewer's recommendations & comments is listed below. All the revisions are TRACKED in the re-submitted WORD file along with marked RED COLOR for the ease of the reviewer's perusal. Our colleague, a native English speaker of BLUE COLOR, has corrected grammatical and writing style errors in the original version.

Recommendations and Comments of Reviewer	Authors' Summary of the Changes
<p>The manuscript can be an interesting contribution for the methodology of use and interpretation of data for the prediction of deep landslide movements. However, it requires a substantial review in the text, in the figures and in the production of additional figures to show the final results. The list presented below are the specific comments:</p>	<p>We are pleased to receive positive feedback from the reviewer on this study. We also sincerely appreciate the reviewer's detailed comments, which have identified the limitations of our research. We have endeavored to revise the manuscript in response to each of the reviewer's comments. The details of these revisions are outlined below.</p>
<p>1) Sections 3.1 and 3.2 should be in the text in more synthetic form, placing much of the content in an appendix</p>	<p>We completely agree with the reviewer's suggestion. Excessive focus on the operational mechanisms of the AI models could distract readers from the primary objective of the study. Therefore, we have moved this content to the appendix.</p> <p>260 3.1 Convolutional Neural Networks 261 In 1998, LeCun introduced a novel type of DNN known as the CNN, specifically designed for 262 processing data with a grid-like structure, such as images. The complex, layered system of CNN facilitates 263 the automated extraction of features without extensive preprocessing, making it ideal for object 264 recognition, image classification, and segmentation tasks. The detailed mechanism of the CNN model can 265 be found in appendix A. The architecture of a typical CNN, as illustrated in Figure 2, comprises an input 266 layer (to receive image data), followed by hidden layers (including convolutional, pooling, and fully 267 connected layers), and concludes with the output layers. As depicted in Figure 2, the complexity of CNN 329 3.2 Deep Learning Models for Time Series 330 RNN was introduced by Elman in 1990 (Elman, 1990). This model makes predictions based on 331 sequential data, crucial for language modeling, document classification, and time series analysis. The 332 architecture of an RNN model can be found in appendix B.</p> <p style="text-align: center;">APPENDIX</p> <p>1 Appendix A. Convolutional Neural Networks</p> <p>2 The architecture of a typical CNN, as illustrated in Figure A-1, comprises an input layer (to receive 3 image data), followed by hidden layers (including convolutional, pooling, and fully connected layers), and 4 concludes with the output layers. As depicted in Figure A-1, the complexity of CNN progressively 5 increases from the convolutional layer to the fully connected (FC) layer. This design enables CNN to 6 recognize relatively simple patterns (lines, curves, etc.) before progressing to capture more intricate 7 features (faces, objects, etc.), with the ultimate aim of extracting relevant information for accurate pattern 8 identification. 9</p> <div style="text-align: center;">  <p>The diagram illustrates the structure of a basic CNN. It starts with an 'Input' layer represented by a stack of yellow rectangles. This is followed by a 'Convolution + Activation function' layer, also shown as a stack of yellow rectangles. The next stage is 'Pooling', represented by a smaller stack of yellow rectangles. Finally, the data passes through a 'Fully connected layer', depicted as a network of blue and red nodes, leading to an 'Output' layer with red nodes. A dashed line labeled 'Feature extraction' spans from the input to the fully connected layer.</p> </div> <p style="text-align: center;">Figure A-1. Structure of basic CNN.</p>

	<p>12 As illustrated in Figure A-2, the convolutional layer is responsible for most computations in the 13 network. This involves extracting local features from an image using a set of learnable filters known as 14 kernels. The behavior of the filter in the convolutional layer is influenced by two main factors: stride and 15 padding. Stride refers to the pixel shift of the filter across the image, while padding aims to preserve 16 information at the corners. In each iteration, a portion of the image is convolved with a filter to generate 17 a dot product of pixels within its receptive field. This process is replicated across the entire image to 18 produce a feature map. The convolution operation is defined as follows:</p>
<p>2) In section 3.4.2 the equation of the MAPE, MAE and RSME objective function is not presented</p>	<p>We have revised Section 3.4.2, adding detailed explanations of the calculations and the significance of each evaluation metric. These explanations enable readers to better understand the objective function when these evaluation metrics are applied.</p> <p>375 3.4.2 Performance Metrics</p> <p>376 This study utilized four widely recognized performance measures to assess the model's model's 377 effectiveness in prediction accuracy (Chou and Nguyen, 2023). The measures included mean absolute 378 error (MAE), mean absolute percentage error (MAPE), and root mean square error (RMSE).</p> <p>379 MAE represents the mean of absolute errors, calculated as the average of the absolute 380 differences between actual and predicted values. Its advantage lies in its simplicity, which 381 provides a straightforward measure of average prediction error. However, a drawback of MAE is 382 its insensitivity to more significant errors, so it may not effectively highlight differences between 383 models when significant errors are present. It is defined as:</p> $384 MAE = \frac{1}{n} \sum_{i=1}^n y_i - \hat{y}_i \tag{5}$ <p>385 where n is the number of predictions, y_i is the i^{th} forecasted value, and \hat{y}_i is the corresponding i^{th} 386 actual value.</p> <p>387 MAPE quantifies the average absolute error ratio—derived from the differences between 388 actual and forecasted values—to the actual value. It provides a clear metric in percentage terms, 389 facilitating straightforward interpretation across various datasets. However, MAPE's limitation 390 arises from its sensitivity to zero values in the actual data, which can become undefined or 391 impractical to compute, limiting its utility in scenarios involving zero or near-zero actual values. 392 The expression for MAPE is as follows:</p> $393 MAPE = \frac{1}{n} \sum_{i=1}^n \left \frac{y_i - \hat{y}_i}{y_i} \right \tag{6}$ <p>394 where n is the number of predictions, y_i is the i^{th} forecasted value, and \hat{y}_i is the corresponding i^{th} 395 actual value.</p> <p>396 RMSE represents the square root of the average squared error between actual and forecasted 397 values and is widely used for its ability to indicate the dispersion of errors. This method captures 398 the magnitude and direction of errors, making it practical for assessing overall prediction 399 accuracy. However, RMSE tends to be more sensitive to outliers and significant errors than MAE 400 due to its squaring of errors during computation. This sensitivity can disproportionately affect its 401 evaluation in datasets with extreme values. The expression for RMSE is as follows:</p> $402 RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \tag{7}$ <p>403 where n is the number of predictions, y_i is the i^{th} forecasted value, and \hat{y}_i is the corresponding i^{th} 404 actual value.</p>
<p>3) Section 3.5 - Chou and Nguyen in 2024 article not present in the bibliography or not mentioned in the correct form</p>	<p>The AEIO algorithm employed in this study was developed in 2024. It has successfully undergone testing on small, average, and large-scale benchmark functions, as well as in optimizing the hyperparameters of AI models. However, since the algorithm is currently under review for publication in a separate journal, we are unable to include it as a reference in this manuscript. We kindly ask for the reviewers' understanding regarding this limitation. Although we have not added a citation for the AEIO algorithm, we have provided a highly detailed explanation of its usage to ensure that readers can easily understand and apply it, as outlined below.</p>

3.5 Age of Exploration-Inspired Optimizer

This study employs a range of AI models to forecast deep-seated displacement in mountainous regions. To enhance the prediction accuracy of these AI models, the study incorporates a novel metaheuristic optimization algorithm known as the Age of Exploration-Inspired Optimizer (AEIO). Developed by Chou and Nguyen in 2024, this algorithm has demonstrated high effectiveness in fine-tuning the hyperparameters of AI models. This algorithm treats each particle in the search domain as an explorer. The movement of particles toward regions with higher fitness values parallels the exploratory activities of the Age of Exploration, where explorers sought ideal locations for establishing colonies. In this study, each particle represents a set of hyperparameters, with the ultimate goal of the search process being to identify the optimal particle or hyperparameter set that minimizes prediction error for AI models. Figure 8 illustrates the AEIO algorithm.

The strength of the AEIO algorithm lies in its ability to develop specific strategies for particles based on their positions, enabling faster convergence to the optimal point. Using density-based spatial clustering of applications with noise (DBSCAN) for particle clustering, the AEIO determines whether particles are in favorable or unfavorable positions, reminiscent of explorers during the Age of Exploration. The proximity (within clusters) allows explorers to gather information and move toward optimal locations, thereby enhancing their ability to establish new colonies. In contrast, explorers far apart (outside clusters) adopt different strategies, relying on limited peer guidance or general trends in their quest for new territories.

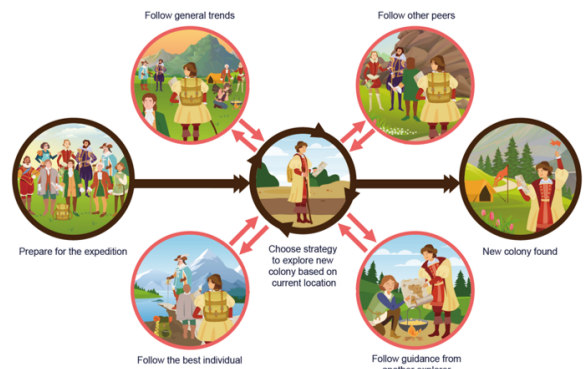


Figure 8. Illustration of Age of Exploration-Inspired Optimizer

In each iteration, explorers forecast their next move. If it promises a better position, they relocate. Otherwise, if the new spot is less favorable for colony establishment, they stay put and await the next iteration. The algorithm employs specific mathematical formulas to calculate the movement step of explorers or particles in the AEIO. The exploratory steps of explorer in the AEIO algorithm will continuously iterate until the stop condition is satisfied.

• Explorers follow general trends

The explorer choosing this movement type will calculate the distance from their location $x_{i,d}(t)$ to the center of all other explorers ($Meanvl_d(t)$), then attempt to move towards that central point in the hope of finding a better location with the potential to establish a new colony. The following formula determines the explorer's position after the movement:

$$x_{i,d}(t+1) = x_{i,d}(t) + \alpha * (Meanvl_d(t) - x_{i,d}(t)) * rand(0,1) * R \quad (8)$$

$$Meanvl_d(t) = \frac{x_{1,d}(t) + x_{2,d}(t) + \dots + x_{n_{Pop},d}(t)}{n_{Pop}} \quad (9)$$

where $d = 1, 2, \dots, D$; D is the number of dimensions; $i = 1, 2, \dots, n_{Pop}$; n_{Pop} is the total number of explorers; $t = 1, 2, \dots, MaxIt$ is the number of iterations; $MaxIt$ is the maximum value of iteration; α is a parameter for adjusting the particle's movement toward the centroid position (usually equals 3). $Meanvl_d(t)$ is the centroid of all particles in dimension d . $rand(0,1)$ is the random number in the range $[0,1]$. R : a number that equals 1 or 2 depending on the value of $rand(0, 1)$ per the equation. $R = round(1 + rand(0,1) * 1)$, $x_{i,d}(t)$ is the location of particle i in iteration t , $x_{i,d}(t+1)$ is the location of particle i in iteration $(t+1)$.

• Explorers follow three other peers

Explorers employing this movement method will calculate the average position of three randomly selected other explorers ($\frac{x_{1,d}(t) + x_{2,d}(t) + x_{3,d}(t)}{3}$) and then move toward this newly calculated average position. The explorer's new position is computed using the following formula:

$$x_{i,d}(t+1) = x_{i,d}(t) + \left(\frac{x_{1,d}(t) + x_{2,d}(t) + x_{3,d}(t)}{3} - x_{i,d}(t) \right) * rand(0,1) * R \quad (10)$$

where: $x_{1,d}(t)$, $x_{2,d}(t)$ and $x_{3,d}(t)$ are three random explorers in dimension d at iteration t , $d = 1, 2, \dots, D$; D is the number of dimensions; $i = 1, 2, \dots, n_{Pop}$; n_{Pop} is the total number of explorers; $t = 1, 2, \dots, MaxIt$ is the number of iterations; $MaxIt$ is the maximum value of iteration.

	<p>453 • Explorers follow the best one</p> <p>454 According to this strategy, the explorer ($x_{i,d}(t)$) will move closer to the position of another explorer</p> <p>455 currently holding the best position ($Best_d(t)$), as determined by the following formula:</p> <p>456 $x_{i,d}(t+1) = x_{i,d}(t) + (Best_d(t) - x_{i,d}(t)) \times rand(0,1) \times R$ (11)</p> <p>457 where: $Best_d(t)$ represents the position of the particle with the best fitness in dimension d at iteration t,</p> <p>458 the parameters d and t hold the same significance as defined in Equation 10.</p> <p>459 • Explorers follow guidance from another one</p> <p>460 Explorers in favorable positions with access to information can execute this movement strategy. In</p> <p>461 this scenario, explorers ($x_{i,d}(t)$) will consult with each other another explorer. The consulted explorer will</p> <p>462 compare their direction and distance to the best individual, who holds the most favorable position</p> <p>463 ($Best_d(t)$) and guide the inquirer. This algorithm assumes that the inquirer can be any explorer, i.e., a</p> <p>464 random explorer ($x_{1,d}(t)$). The following formula describes how to calculate the new position of the</p> <p>465 explorer following this strategy:</p> <p>466 $x_{i,d}(t+1) = x_{i,d}(t) + (Best_d(t) - x_{1,d}(t)) \times rand(0,1) \times R$ (12)</p> <p>467 where: $x_{1,d}(t)$ is a random explorer in dimension d at iteration t. the parameters d and t hold the same</p> <p>468 significance as defined in Equation 10.</p> <p>469 • Crowd control mechanism</p> <p>470 To enhance the efficiency of AEIO in transitioning between exploration and exploitation, a</p> <p>471 mechanism is employed to adjust the parameters of DBSCAN throughout each cycle, according to the</p> <p>472 following formula:</p> <p>473 $\epsilon_d = \left(0.1 + \frac{t}{MaxIt}\right) \times (MeanVl_d(t) - Best_d(t))$ (13)</p> <p>474 $MinPts = round\left(1 + \frac{t}{MaxIt} \times 10\right)$ (14)</p> <p>475 The exploratory steps in the AEIO algorithm begin by classifying positions using the DBSCAN</p> <p>476 algorithm. Subsequently, the explorers update the crowd control mechanism according to equations (13)</p> <p>477 and (14), and move according to various strategies defined by equations (8), (10), (11), and (12). This</p> <p>478 process is conducted iteratively, continuing until the maximum number of iterations is reached.</p> <p>479 To fine-tune the hyperparameters of AI models, the AEIO algorithm treats each hyperparameter as</p> <p>480 a variable. Furthermore, the objective function of the AEIO algorithm seeks to minimize the prediction</p> <p>481 error of AI models, which is quantified by an evaluation metric (MAPE). Figure 4 presents a flowchart</p> <p>482 illustrating the process by which the AEIO algorithm aids in fine-tuning hyperparameters for AI models.</p> <p>Additionally, the AEIO algorithm demonstrated strong optimization capabilities for the hyperparameters of AI models in this study, highlighting its effectiveness.</p>
<p>4) Section 3.5 - EQ. 10 and 11 - The meaning of the Maxit and Mind parameters are not indicated</p>	<p>We acknowledge the error in our initial manuscript, as pointed out by the reviewer's suggestion. We have now added annotations for the parameters d, D, n_{pop}, t, and $MaxIt$ in Equation (10). Additionally, we have clarified that these values hold the same meaning in Equations (11) and (12).</p> <p>445 • Explorers follow three other peers</p> <p>446 Explorers employing this movement method will calculate the average position of three randomly</p> <p>447 selected other explorers $\left(\frac{x_{1,d}(t)+x_{2,d}(t)+x_{3,d}(t)}{3}\right)$ and then move toward this newly calculated average</p> <p>448 position. The explorer's new position is computed using the following formula:</p> <p>449 $x_{i,d}(t+1) = x_{i,d}(t) + \left(\frac{x_{1,d}(t)+x_{2,d}(t)+x_{3,d}(t)}{3} - x_{i,d}(t)\right) \times rand(0,1) \times R$ (10)</p> <p>450 where: $x_{1,d}(t)$, $x_{2,d}(t)$ and $x_{3,d}(t)$ are three random explorers in dimension d at iteration t, $d = 1, 2, \dots, D$;</p> <p>451 D is the number of dimensions; $i = 1, 2, \dots, n_{pop}$; n_{pop} is the total number of explorers; $t = 1, 2, \dots, MaxIt$</p> <p>452 is the number of iterations; $MaxIt$ is the maximum value of iteration.</p> <p>453 • Explorers follow the best one</p> <p>454 According to this strategy, the explorer ($x_{i,d}(t)$) will move closer to the position of another explorer</p> <p>455 currently holding the best position ($Best_d(t)$), as determined by the following formula:</p> <p>456 $x_{i,d}(t+1) = x_{i,d}(t) + (Best_d(t) - x_{i,d}(t)) \times rand(0,1) \times R$ (11)</p> <p>457 where: $Best_d(t)$ represents the position of the particle with the best fitness in dimension d at iteration t,</p> <p>458 the parameters d and t hold the same significance as defined in Equation 10.</p> <p>459 • Explorers follow guidance from another one</p> <p>460 Explorers in favorable positions with access to information can execute this movement strategy. In</p> <p>461 this scenario, explorers ($x_{i,d}(t)$) will consult with each other another explorer. The consulted explorer will</p>

462 compare their direction and distance to the best individual, who holds the most favorable position
 463 ($Best_d(t)$) and guide the inquirer. This algorithm assumes that the inquirer can be any explorer, i.e., a
 464 random explorer ($x_{1,d}(t)$). The following formula describes how to calculate the new position of the
 465 explorer following this strategy:
 466
$$x_{1,d}(t+1) = x_{1,d}(t) + (Best_d(t) - x_{1,d}(t)) \times rand(0,1) \times R \quad (12)$$

 467 where: $x_{1,d}(t)$ is a random explorer in dimension d at iteration t , the parameters d and t hold the same
 468 significance as defined in Equation 10.

5) Section 3.6.0-In Figure 9, references are indicated to the 18-19-20-21 and 22 equations. But these equations do not exist and the text

We have revised the equation numbering in this flowchart to ensure consistency with the sequence of equations presented earlier.

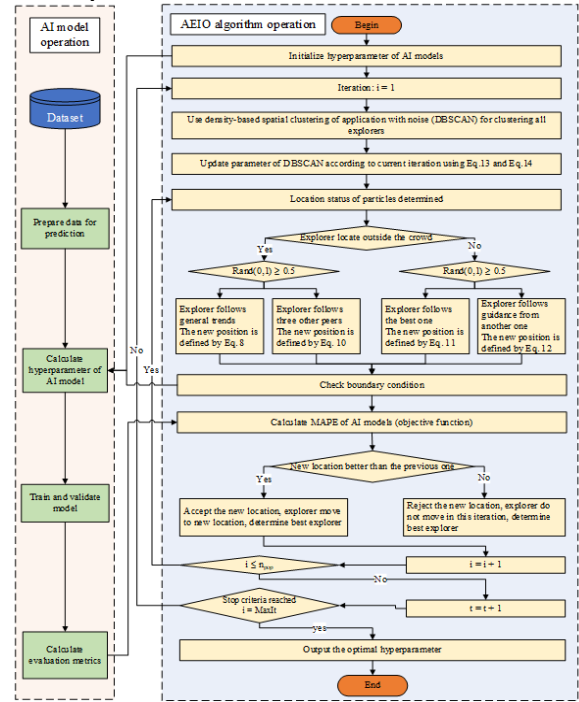


Figure 9. Flowchart of the fine-tuning process of AI models by the AEIO algorithm

6) section 3.6.0 in Figure 9 and in the text the optimization stop criterion should be indicated.

We fully agree with the reviewer's suggestion and have added content to the manuscript to emphasize the stop criterion of the AEIO algorithm.

426 In each iteration, explorers forecast their next move. If it promises a better position, they relocate.
 427 Otherwise, if the new spot is less favorable for colony establishment, they stay put and await the next
 428 iteration. The algorithm employs specific mathematical formulas to calculate the movement step of
 429 explorers or particles in the AEIO. The exploratory steps of explorer in the AEIO algorithm will
 430 continuously iterate until the stop condition is satisfied.

475 The exploratory steps in the AEIO algorithm begin by classifying positions using the DBSCAN
 476 algorithm. Subsequently, the explorers update the crowd control mechanism according to equations (13)
 477 and (14), and move according to various strategies defined by equations (8), (10), (11), and (12). This
 478 process is conducted iteratively, continuing until the maximum number of iterations is reached.

We have also incorporated the stop criterion into the flowchart of the AEIO algorithm during the fine-tuning of the AI model's hyperparameters.

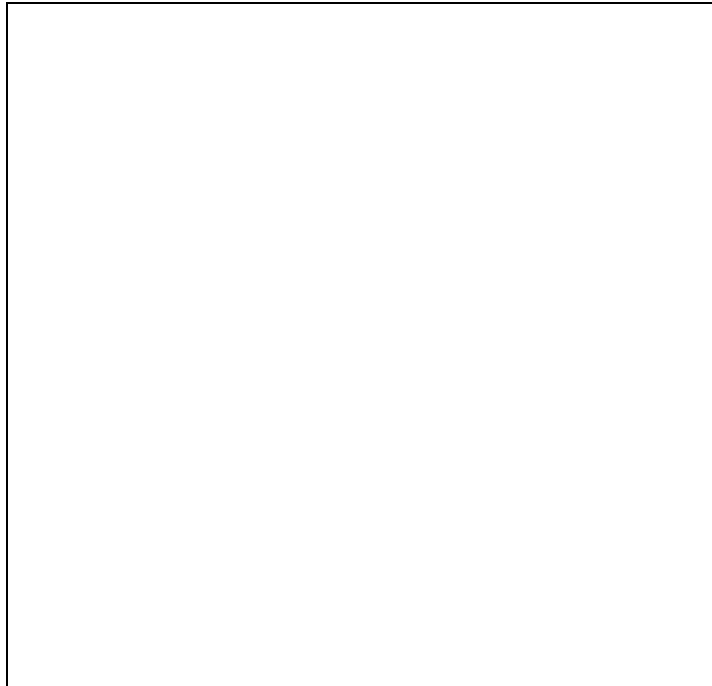


Figure 9. Flowchart of the fine-tuning process of AI models by the AEIO algorithm

7) Section 3.6.2. Figures 12, 13, and 14 should be presented together in the same group with the same temporal axis. And an additional figure should be added to the group, with the temporal sequence of the rains

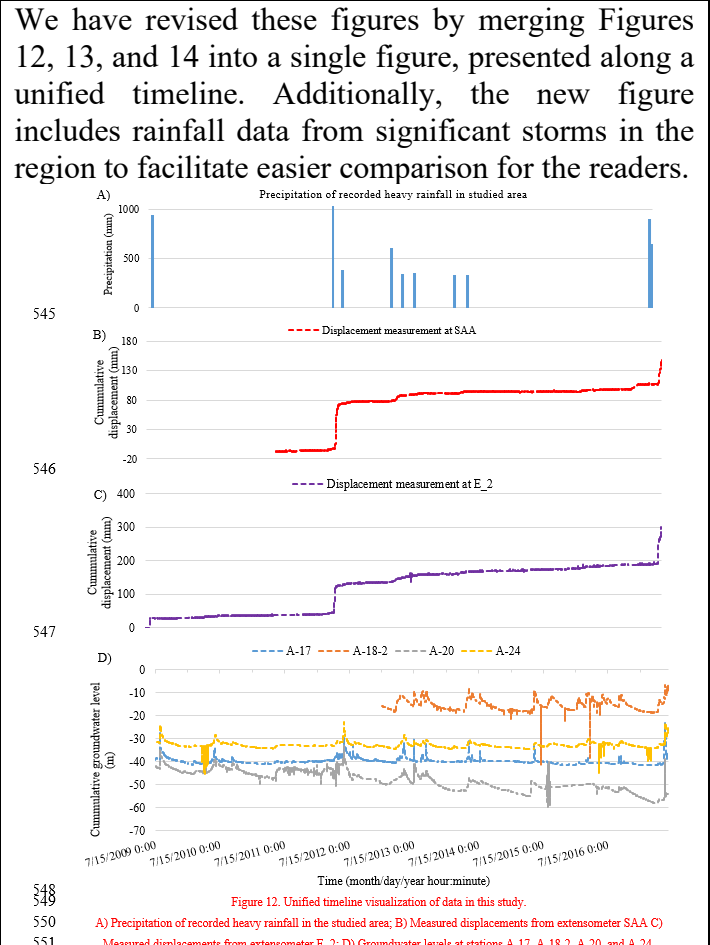


Figure 12. Unified timeline visualization of data in this study. A) Precipitation of recorded heavy rainfall in the studied area; B) Measured displacements from extensometer SAA C) Measured displacements from extensometer E_2; D) Groundwater levels at stations A-17, A-18-2, A-20, and A-24.

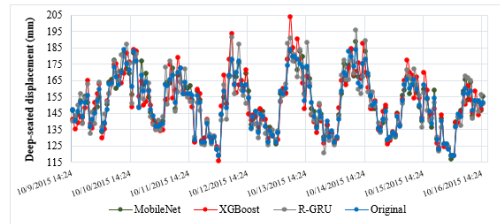
8) in Section 4, the comparative result of the deformations observations (shown in figure 14) with

In response to the reviewer's suggestion, we have added a figure that displays the predicted deep-

the comparative predictions of the best model should be graphically presented.

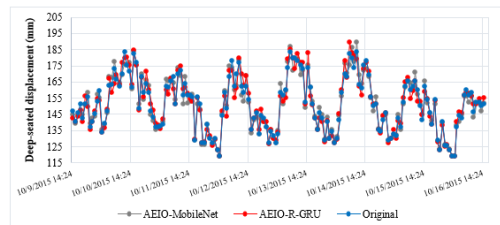
seated displacement of the best machine learning model, the best time-series deep learning model, the best CNN model, and the best hybrid models. This allows readers to compare and assess the predictive capabilities of these models.

771 Figure 10 illustrates the differences between typical AI models' actual and predicted deep-seated
772 displacement. Specifically, Figure 10a compares the performance of single models against the predicted
773 values, while Figure 10b does the same for hybrid models. The chart shows that, hybrid models
774 demonstrate superior predictive capability for deep-seated landslides compared to single models. This is
775 evident from the displacement line of the hybrid models in Figure 10b which closely aligns with the actual
776 deep-seated displacement and significantly outperforms the single models depicted in Figure 10a.



777
778

a) Prediction results of deep-seated displacement by single AI models.



779
780
781

b) Prediction results of deep-seated displacement by AI models optimized using the AEIO algorithm.
Figure 10. Graph comparing the real and predicted deep-seated displacement.

9) section 4.2 is too long and should be simplified and synthesized

We fully understand the reviewer's concern regarding the length of Section 4.2. However, it is important to note that much of the length is due to the inclusion of performance result tables for the models, which are essential and cannot be condensed.

Additionally, we believe that the explanations and commentary on the models' performance are equally essential. These details not only enhance the manuscript's relevance to readers interested in landslide research but also appeal to those focused on the use of AI models for regression studies.

Last but not least, while this section is lengthy, it is organized in a logical structure. As a result, readers will not be distracted by its length; instead, they can easily find information on the specific models they are interested in, corresponding to each subsection within Section 4.2.

However, in response to the reviewer's valuable suggestion, we have revisited Section 4.2 and removed redundant content, retaining only the information that is most valuable to the readers.

660 most suitable machine learning model for predicting deep-seated landslides, exhibiting both high
661 prediction accuracy and a short running time. ~~The following section will compare this model with the best~~
662 ~~time series deep learning model to select the optimal numerical model for fine tuning.~~

664 Similar to the machine learning models, in this section, the time series deep learning models will
665 also be trained with default hyperparameters, as found in ~~research of Chou and Nguyen's research in 2023~~
666 Chou and Nguyen (2023). The performance results of these models are shown in Table 7. Overall, akin to
679 demonstrates higher prediction accuracy. Therefore, the R-GRU model will be chosen as the best
680 numerical AI model. ~~R-GRU will undergo fine tuning in the following section using the AEIO algorithm,~~
681 ~~further enhancing this model's accuracy.~~

692 displacement of R-GRU before fine-tuning was 7.9%, but this number decreased to only 3.03% after fine-
693 tuning. ~~All other predictions similarly show a decreasing trend.~~

703 Additionally, Table 9 indicates that predictions from the dataset of the E-2 station consistently
704 outperform those of the SAA station. ~~Specifically, the displacement prediction at the E-2 station is 3.03%~~
705 ~~and 6.38%, better than the corresponding numbers for the SAA station, which are 3.94% and 7.96%,~~
706 ~~respectively. This is attributed to the dataset collected by the E-2 station being more comprehensive and~~
707 ~~gathered over a more extended period than the SAA station (as shown in Table 4).~~

763 the AEIO algorithm, are presented in Table 14. ~~Compared to models in previous sections,~~ CNN models
764 with optimal hyperparameters ~~obtained in this section exhibit the most minor errors,~~ indicating that these
765 are the most effective models in this study for predicting deep-seated displacement landslide occurrences.

771 Figure 10 illustrates the differences between typical AI models' actual and predicted deep-seated
772 displacement. Specifically, Figure 10a compares the performance of single models against the predicted
773 values, while Figure 10b does the same for hybrid models. The chart shows that, hybrid models
774 demonstrate superior predictive capability for deep-seated landslides compared to single models. This is
775 evident from the displacement line of the hybrid models in Figure 10b which closely aligns with the actual
776 deep-seated displacement and significantly outperforms the single models depicted in Figure 10a.