# Editor

**Where possible authors should design all graphics with greyscale reproduction in mind, being aware that greyscale reproduction of colour images can lead to ambiguous use of greys.**

*We feel that the use of colour figures is justified, and that it makes a lot of concepts easier to understand, instead of using scales of gray or dotted/dashed lines etc. If deemed necessary, we can update the figures into a grayscale form.*

**PLEASE check your English spelling and grammar. The paper will be returned if this is not done.**

*Answer*

**Reviewer 2 also notes that RF is not an AI method. This is correct - it is a machine learning method.**

*We now only say that random forest is a machine learning method.*

**Please try to avoid making the paper longer (keep in mind that the average length of papers in this journal is 18-20 published pages).**

*Answer*

# Reviewer 1

**I suggest not to declare the unknown location of ditches a 'research gap', maybe this rather can be phrased as a research objective**

*Good point, this has now been amended by rephrasing the paragraph 2 of the Introduction section.*

**p2/17ff .. the performance assessment of methods should be based on references or own empirical evidence**

*We have rewritten this section to make it more clear that these performance assessments are based on observations, and that no empirical evidence exists for them. We have also clarified that the reason that the difference in false positives and negatives is relevant is that we hypothesise that combining them would prune their weaknesses and amplify their strengths.*

*This is clarified in Introduction, paragraph 2:*

*“Different false positives and negatives can be observed when examining each terrain index manually. Sky View Factor tends to give a large amount of false positives in steep terrain, whereas Impoundment Index and High Pass Median Filter are more difficult to analyse. Due to the difference in false positives and negatives, we want to combine the indices to help prune the weaknesses and amplify the strengths of each index.”*

**ground truth: more information on length/depth/width of ditches would be desirables, beyond 'on average wider than 0.5m'**

*We have expanded on this in 2.2 Digitising the ground truth:*

*“From the ditch mapping, the vector layer was rasterised so that it could be compared to the automatically derived ditches from the 0.5 \* 0.5 m DEM. Although we have no data about the width of the ditches in the catchment, field observations have shown that the vast majority of ditches are between 0.5 and 3.5 metres wide. Because we wanted to ensure that the model received all ditch pixels correctly in the training phase, we widened the ditches so that all pixels within a radius of three pixels (1.5 metres) of the vectors were labelled as ditch pixels. This ensured that the edges of ditches were included, as well as that the raster converted from vector format actually covered the ditches.”*

**aren't road ditches typically on both sides of a road?**

*We need some clarification on what needs to be corrected with this comment (what paragraph etc.). It is true that road ditches are typically on both sides of the road (sometimes road ditches are only dug on one side of the road if the other side slopes steeply away from the road). However, we can’t find any inconsistencies with this in our article.*

**any depression is considered a ditch, no distinction from the concept of a sink is made, which might be helpful with expressing the linear character of a ditch**

*Several approaches were taken to try to prune sinks incorrectly predicted as ditches. One input variable (which has subsequently been removed in the feature space reduction experiment) attempted to fill small gaps in ditches by looking in opposing directions of a pixel, and updating their values if a suspected ditch was found in both directions.*

*We also use gabor filters for several input variables. Gabor filters are specifically designed to detect linear features in images, and feature using this technique proved to be very effective, as can be seen in the feature importance table (LÄGG TILL KORREKT TABLE). The gabor filter explanation has been clarified in section 2.4.1. Processing the digital terrain indices:*

*“30 Gabor filters, which were rotated in different angles and with different frequencies, were combined to detect lines, amplifying ditches by utilising the fact that ditches have a linear elongated shape (Figure 3: d and g).”*

*Finally, our post processing method makes substantial efforts to remove sinks and small “clusters” of incorrectly predicted ditch pixels. We both look at the size of connected clusters of ditch predictions, as well as the shape of the clusters. We have clarified this explanation somewhat, to make it more clear that the purpose is to retain linear directional characteristics. This post processing step is also illustrated graphically in Figure 8.*

*This is the clarified explanation in 2.6.3. Cluster removal:*

*“A distance calculation was also performed in tandem with this method to find the largest distance of pixels inside each given cluster. This helped to remove sinks and hollows that were not removed by the initial small cluster removal, but that did not have a linear directional characteristic, indicating that they did not represent a ditch (Figure 8 b).”*

**would sink filling assist with developing a suitable index?**

*Sink filling could probably work as a post/pre processing method. However, because of our own cluster removal algorithm (see previous answer), we do not deem the inclusion of a sink filling algorithm necessary. Our cluster removal algorithm is tailored to the specific problem of correctly detecting ditches, whereas a sink filling function is more general, and could possibly fill ditches erroneously.*

**p6/15 typo argGIS**

*This mistake has been fixed!*

**why is not curvature index used?**

*We have made other approaches at removing streams from the predictions. This is described in 2.4.1. Processing the digital terrain indices paragraph 7 (now 6). As can be seen in the feature importance table (LÄGG TILL KORREKT TABLE), these input variables work quite well. They do, however, have the downside of sometimes removing deep ditches as well, which can be seen in Figure 9 (f).*

*Comparing curvature index with our own, and other, stream removal methods would be an interesting addition in a future study, but was out of scope for this study. We talk about this potential future improvement (shape index) in section 4. Discussion, last paragraph.*

**would the concept of negative/positive terrain be helpful?**

*We use a high pass median filter (HPMF) to capture the curvature in the terrain. This is similar to negative/positive terrain. We feel that we cover this adequately in 2.3 Extracting ditches with digital terrain indices:*

*“The algorithm operates essentially by subtracting the value at the grid cell at the centre of the window from the median value in the window. Negative values indicate depressions in the DEM, such as ditches.”*

**p11/13 Python in upper case**

*This mistake has been fixed!*

**table 1: was there any sensitivity analysis done re inclusion/exclusion of variables? (ok, I see this partially answered in table 4)**

*We conducted an experiment where we used two subsections (that are not in the final experiment) and a cross validation method to evaluate each algorithm with varying feature spaces (testing several feature spaces from 81 down to 10 features). This helped in determining the correct machine learning algorithm. The algorithms that were examined are: Extreme Gradient Boosting, Naive Bayes, Support Vector Machines, and Random Forests. Random Forests was shown to be the most suitable algorithm.*

*To determine what input variables to use in the final experiment, we performed a more detailed experiment with Random Forests, in which we also included the post processing steps. The input variables were reduced by 2 at a time, from 81 down to 2. Random Forest’s feature importance was used to determine which variables to exclude from the previous iteration. This approach showed us that using 40 input variables yielded comparable results to using 60 or 70 variables. With this approach, we could remove unnecessary features that dragged the performance and speed down.*

*With these 40 variables, we also performed a detailed parameter tuning which included the post processing steps.*

*The final experiment was then rerun using only these 40 input variables (down from 81), producing completely new results for the study. Adjustments have been made in several parts of the article due to this change:*

*2.4.1. Processing the digital terrain indices, paragraph 5:*

*This paragraph was removed, due to the removal of this input variable. Consequently, a reference to this section in 2.6.1. Noise reduction and gap filling, paragraph 2 was updated to function without referencing the removed paragraph.*

*We added a paragraph to explain the sub-experiment for variable selection:*

*“Overall, 81 input variables were extracted from the terrain indices. We then conducted a sub-experiment to find the best input variables, as well as the optimal number of variables to use. Random Forest's feature importance was used to select the most valuable input variables for different variable amounts. It was found that using 40 input variables produced the best results.”*

*Of course, this also means new result metrics for the end experiment, as well as new feature importances. These results are better for some metrics and worse for other metrics. This is in part due to using fewer input variables, but also using a different random seed for the Random Forest algorithm. In the previous experiment, no random seed was set, meaning that the results were not completely reproducible. We now use a random seed to ensure the validity of the results.*

**overall, ditches typically have rather stable directional characteristics. This seems not to be considered in the chosen parametrisation (higher probability for a pixel representing a ditch on one direction over others)**

*See answer to the comment about any depression being considered a ditch, and no distinction being made from the concept of a sink (Reviewer 1).*

**similarly, is (vector) topology a potential useful approach to fill gaps and connect ditches?**

*This would be interesting in a future study, but we feel that it is beyond the scope of this paper. Because the article is already lengthy, we can’t introduce several new terrain indices for this study.*

**p15/15 replace chance with random?**

*Chance has been replaced with random when referring to the Cohen’s Kappa metric.*

**in several places, replace 'cavity' with 'sink' or 'concavity'?**

*Good point, the word cavities has been replaced with the word sinks throughout the text.*

**it would be interesting to compare the ML/RF approach with using simple morphographic indicators, whether there is any significant improvement?**

*This was done in the study, we compare our method to the Sky View Factor, Impoundment Index, High Pass Median Filter, as well as Slope indices. Both Slope and HPMF (positive/negative terrain) are morphographic indicators. see Result section: paragraph 1, and Table 2.*

*We talk about this in the introduction, last paragraph. We have clarified this slightly by adding “compared to if a single index would be used”:*

*“We hypothesise that by combining the information from all the digital terrain indices using a machine learner (Random Forests), we can improve the detection of the ditches, compared to if a single index would be used.”*

*We also talk about this in the discussion, first paragraph. We have also here clarified slightly by adding “than if indices are used separately”:*

*“In our study we have shown that it is possible to locate ditches automatically in high-resolution DEMs (0.5∗0.5 metres, in our case), and that more of the ditches can be detected if the information from several terrain indices (Table 2) is combined through machine learning than if indices are used separately (Table 3).”*

# Reviewer 2

**The authors combine the information from all the digital terrain indices using Random Forests to improve the detection of the ditches. Why we should combine terrain indices and RF? Authors need to emphasize the significance and originality of this work in the introduction because terrain indices, and RF for ditch detection have done by other researchers.**

*We agree that the approach of the study needed clarification in the Introduction, we have added two sentences to the Introduction, paragraph 4 to highlight that the originality of our work is the development of custom input variables for the model, as well as custom post processing steps:*

*“Several other input variables for the model are calculated from the indices to place emphasis on the clear linear directional characteristics of ditches. These characteristics are also used when processing the model's output, in order to remove noise and other inconsistencies in the predictions.”*

*We feel that we have already motivated why we should combine indices and use Random Forests in Introduction, last paragraph:*

*“We hypothesise that by combining the information from all the digital terrain indices using a machine learner (Random Forests), we can improve the detection of the ditches, compared to if a single index would be used.”*

**Random forest is a commonly used machine learning algorithm and in most cases are worked. So that be ok to use RF to detect ditches. But my concern is that I notice the author used 81 features to train the RF model and my question is that do all these features worked or not? More specifically, in table 4, the author list top 20 important input variables. And we can easily find out from the table that, the importance score of the top one feature (Impoundment mean 3) is much higher than the 20th feature (Slope non-ditch amplification) in the list. So, I guess the score of the least important feature could be extremely low (maybe 0.1 or 0.05 and something like that). Therefore, in that case, I doubt that it’s unnecessary to use all 81 features. Authors need to make more analysis about this problem. You perhaps can reference the following paper:**

**Georganos S , Grippa T , Vanhuysse S , et al. Less is more: Optimizing classification performance through feature selection in a very-high-resolution remote sensing object-based urban application[J]. GIScience & Remote Sensing, 2017.**

*We agree. See the answer to Reviewer 1’s comment regarding inclusion/exclusion of variables.*

**Section 2.4.2, you manually divided the dataset into 21 subsections. All the upper subsection was used to develop RF model (figure 1), and all the lower was used to validate. I doubt that using all the upper may cause systematic bias for the model. In general, the subsection must be divided and selected randomly for RF model, not manually. Please revise your model or explain why.**

*We probably have not explained our methodology clearly enough. The upper 10 subsections were not used to build a model that was evaluated on the 11 lower subsections. Rather, we used a cross validation approach, where each of the lower 11 subsections were used to evaluate on once with a model trained from the remaining subsections (i.e. 11 different models were trained for the experiment). The cross validation process has been clarified in subsection “2.4.2 Training and validation datasets - paragraph 2” by adding the sentence: “* A new model is trained using the training folds for each iteration (Figure 4).*”. A figure was also added to graphically explain the cross validation approach with the 11 hold-out subsections.*

*We feel that subsection “2.4.2 Training and validation datasets - paragraph 2” explains clearly that the upper 10 zones are not used in the final experiment: “The remaining 10 subsections were used solely before the experiment to develop the model to test different ways of preparing the data. This allowed the model to be evaluated on unseen data to strengthen the validity of the experiment. Figure 1 shows which subsections were used for development and evaluation respectively.”.*

*However, we changed the use of the word “model” to “models” when referencing the models used in the final experiment in a lot of places throughout the text to make it clear that multiple models are built for the final experiment.*

**the title can not represent this work well. A lot of methods for detection ditch from DEM have proposed. Perhaps you need to emphasis your feature.**

*We have now changed the title to include the word machine learning, to more accurately represent the approach of our study.*

**Page 2, line 29. You said you used artificial intelligence. Which AI are you used? RF is not AI.**

*We now only say that random forest is a machine learning method.*

**In section 2.3 (page 5). do you have any basis for the window selection of the terrain indices? Such as radius of 10 m for Sky View Factor, dam length of 3 m for Impoundment Index.**

*We based the window selection on what produced the best results with observation, and trial and error. Most ditches are not wider than 3 metres, and the impoundment index radius needed to cover the ditch sufficiently, without looking too far. The sky view factor radius was not as sensitive, but it also needed to cover the ditch. We set the limit to 10 metres to avoid the index being affected by steep hills in close proximity to ditches.*

*We have updated the motivations slightly in 2.3 Extracting ditches with digital terrain indices:*

*“The Sky View Factor was calculated in SAGA GIS, using a search radius of 10 metres to ensure that ditches were covered, and to not let steep hills close to ditches affect the results.”*

*and*

*“Here, we used dam height calculated with a dam length of 3 metres, to cover the entire ditch which is typically not wider than 3 metres.”*

**Section 2.4.1, page6, line 44. “This facilitated finding obscurities in the neighbouring areas around pixels”. The ‘obscurities’ is hard to understand. What is mean?**

*We agree. This sentence has now been rewritten to:*

*“Using these statistical aggregations with the help of neighbouring areas around pixels aided in pruning pixels with outlier values, often smoothing out the data to represent ditches more accurately on a per-pixel basis.”*

**Page 12, line 52. What is the custom function? Please specify.**

*The custom function is described in the same paragraph. We have clarified that the sentences following the first sentence are describing the custom function by adding “In this function,” to the start of the sentence:*

*“The second step for removing noise was to use a custom function to remove pixels that had a semi-high ditch probability, but that lay far away from any other high probability pixels. In this function, a threshold value was used to avoid removing pixels with a high enough ditch probability, helping to retain pixels that lay in or close to a ditch. The max ditch probability value in a circular radius of 10 pixels was then calculated. If this max value was too low, the probability value of the examined pixel was lowered (Figure 7 a).”*

**Page 13, line 44-45. Why 6 \* 6 grid zone, and probability of 35 % are used?**

*We evaluated several different zone sizes (from 2\*2 to 12\*12 pixels), but 6\*6 showed the best results overall. We also noticed a numerical error in this paragraph: a probability threshold of 40 % (not 35 %) was used in the experiment. The probability threshold, similarly to the grid zone size, was evaluated in a pilot study, and 40 % showed the best results.*

*We have updated the article to clarify this:*

*“A mean probability rating was calculated for each 6∗6 grid zone, classifying the entire zone as part of a ditch if the mean probability exceeded 40 %. Different zone sizes and probability levels were evaluated in a pilot study, and these values were found to produce the best results.”*

**Page 16, line 31-38 and figure 8. You modified the evaluation labels, which will affect the accuracy assessment. I disagree to modify original label data. you used the data to train the RF model and then you modify it when you assess accuracy. This contradicts itself. For your concerns, perhaps you can convert the raster results to vector form. Then, comparing it with your digitized label ditch. When the detection result falls in a suitable buffer zone of the label ditch, you can regard it as correct detection.**

*We understand this concern. However, we feel that the modified evaluation labels are still justified. As proposed, a buffer zone is a good approach to avoid punishing the model’s prediction incorrectly. We feel that our evaluation method solves the incorrect punishment of ditch pixels adjacent to ditches with a “custom buffer”. This is an overall problem with classifying pixels, when the labelled data is derived from a vector format without knowledge of the width of individual ditches.*

*The Random Forests model, with our method, will receive the correct label for the majority of the pixels during the training phase, and therefore should learn to correctly detect ditches fairly well, despite some pixel’s labels being incorrectly around the edges of ditches. When predicting, however, using these incorrectly labelled pixels as a ground truth, would do the model an injustice, producing metrics that do not accurately describe the performance of the model.*

*If we had converted our predictions to vector format, many more assessment difficulties would have been created: We would not be able to use evaluation metrics such as Cohen’s Kappa or Area under the precision/recall curve. It would also be difficult to find a fair way to assess partial ditch detection, or incorrect ditch detection if vectors were used. For instance, if a ditch of incorrect shape and location was predicted, should it be punished more than other incorrect predictions? How is it turned into a vector if it does not have a linear shape?*

**Figure 8. why do you use 3 m2 zone?.**

*This is another numerical error, we have corrected this. The zones are 9 m2 (3 \* 3 metres). This is the same zone size that has been used when producing the final prediction, as described in section “2.6.2. Binarisation with grid zones”.*

**Some important figures, such as Figure 5 to 8, should be added legend instead of making too many descriptions in figure title.**

*We have revised the figure descriptions somewhat. We have tried to follow the style guides in the LaTeX template: “Template for International Journal of Geographical Information Science”, as well as used other published articles as a reference point.*

*Instead of adding legends to the figures, we have shortened the description of each sub-figure to only act as a headline for the sub-figure, and moved up some of the descriptions to the overall figure description, instead of as a sub-figure description.*

*For example in Figure 9 (previously Figure 8):*

*“Illustration of the modified evaluation labels where* ***a*** *shows the original result, and* ***b*** *shows the modified result. Green marks true positives, red marks false negatives, and blue marks false positives. False positives and false negatives that lay within one grid zone (9 m2) of a ditch label were evaluated as true positives and true negatives.”*

*was changed to:*

*“Illustration of the modified evaluation labels. Green marks true positives, red marks false negatives, and blue marks false positives. False positives and false negatives that lay within one grid zone (9 m2) of a ditch label were evaluated as true positives and true negatives.* ***a:*** *original results,* ***b:*** *modified results.”*

*Figures 6-8 (previously 5-7) had the headline for the sub-figures shortened.*

**In the discussion part, I suggest the author make some comparative analysis between RF and other machine learning methods such as SVM, ANN, etc., which are also very commonly used.**

*We conducted an experiment to ensure that the best algorithm for the task was selected. We compared four algorithms (as described in the answer to Reviewer 1’s comments regarding inclusion/exclusion of variables): Extreme Gradient Boosting, Random Forests, Naive Bayes, and Support Vector Machines. We used two zones (that are not in the final experiment) and a cross validation method to evaluate each algorithm with varying feature spaces (testing several feature spaces from 81 down to 10 features). Random Forests was shown to be the best performing algorithm with respects to the Cohen’s Kappa metric.*

*We feel that a more proper place in the report to talk about this is in the Introduction section as a reason to select Random Forests as the algorithm, rather than in the Discussion section. The following sentence was added to Introduction, paragraph 4: “In a pilot study, we compared several different algorithms (Random Forests, Extreme Gradient Boosting, Naive Bayes, and Support Vector Machines), and it was found that Random Forests produced the most accurate results.”*

**The conclusion needs to be more accurate of a description of the work.**

*We have rewritten large parts of the conclusion to more accurately reflect the queries raised in the Abstract and Introduction. We removed some parts that felt shoehorned in, and added more suggestions for future work that have been hinted at throughout the article.*

## Change log

## **2.3. Extracting ditches with digital terrain indices - paragraph Slope**

ArgGIS ⇒ ArcGIS

## **1. Introduction - paragraph 2**

A major research gap is that the locations of many ditches are unknown. A comparison between a field inventory of ditches (within the National Inventory of Landscapes in Sweden (NILS)) and current maps, shows that approximately 90 \% of the ditch networks are missing on current maps (Ågren, unpublished). To better prioritise the restoration of ditched wetlands, it is necessary to know where the ditches are located, but field inventories are often too costly. A solution to this can be to automatically detect ditches from high-resolution digital elevation models (DEMs) using digital terrain indices.

⇒

A comparison between a field inventory of ditches (within the National Inventory of Landscapes in Sweden (NILS)) and current maps, shows that approximately 90 \% of the ditch networks are missing on current maps (Ågren, unpublished). To better prioritise the restoration of ditched wetlands, it is necessary to know where the ditches are located, but field inventories are often too costly. The research objective of this study is therefore to automatically detect ditches from high-resolution digital elevation models (DEMs) using digital terrain indices.

## **1. Introduction - paragraph 3**

From the high resolution DEM, we use artificial intelligence in the form of machine learning to locate the ditch networks.

⇒

From the high resolution DEM, we use machine learning to locate the ditch networks.

## **2.5. Building the random forests model - paragraph 5**

## python ⇒ Python

## **2.7. Evaluation - paragraph 1**

Cohen's κ index measures how much better a prediction is compared to a prediction based purely on chance, where chance would yield a value of zero (Sim and Wright 2005). With our data, a κ value close to zero would be attained by predicting 2 % of the occurrences as ditch pixels completely at random. Values above zero are better than chance and values below zero are worse than chance.

⇒

Cohen's κ index measures how much better a prediction is compared to a completely random prediction, where random would yield a value of zero (Sim and Wright 2005). With our data, a κ value close to zero would be attained by predicting 2 % of the occurrences as ditch pixels completely at random. Values above zero are better than random, and values below zero are worse than random.

**2.7. Evaluation - paragraph 2**

chance ⇒ random

**4. Discussion - paragraph 3**

This means that our model performed substantially better than one based purely on chance.

⇒

This means that our model performed substantially better than a completely random model.

## **1. Introduction - paragraph 2** & **2.4.1 Processing the digital terrain indices - paragraph 5** & **2.6.3 Cluster removal - paragraph 1** & **4. Discussion - paragraph 4**

cavities ⇒ sinks

**2.4. Processing the digital terrain indices - paragraph 2**

This facilitated finding obscurities in the neighbouring areas around pixels.

⇒

Using these statistical aggregations with the help of neighbouring areas around pixels aided in pruning pixels with outlier values, often smoothing out the data to represent ditches more accurately on a per-pixel basis.

**1. Introduction - paragraph 4**

We reproduce these neighbouring area variables when building the model in our study. Random Forests also computes Mean Decrease in Impurity, or Gini importance, highlighting what input variables are the most important for a given prediction (Menze *et*  *al.* 2009). Because of this, and the fact that it proved to be suitable for Roelens *et al.* (2018), a Random Forests model is used to locate the ditches.

⇒

We reproduce these neighbouring area variables when building the model in our study. In a pilot study, we compared several different algorithms (Random Forests, Extreme Gradient Boosting, Naive Bayes, and Support Vector Machines), and it was found that Random Forests produced the most accurate results. Random Forests also computes Mean Decrease in Impurity, or Gini importance, highlighting what input variables are the most important for a given prediction (Menze *et al.* 2009). Because of these factors, and the fact that it proved to be suitable for Roelens *et al*.(2018), Random Forests models is used to locate the ditches.

**2.4.1 Processing the digital terrain indices - paragraph 6 (now 5)**

Both HPMF and Sky View Factor were used with the image processing gabor filter,which can be used to detect lines of a certain orientation in an image (Hong *et al.* 1998). To detect lines in all directions, 30 Gabor filters, which were rotated in different angles and with different frequencies, were combined to amplify ditches, see Figure 3:d and g.

⇒

Both HPMF and Sky View Factor were used with the image processing gabor filter,which can be used to detect lines of a certain orientation in an image (Honget *al.*1998). 30 Gabor filters, which were rotated in different angles and with different frequencies, were combined to detect lines, amplifying ditches by utilising the fact that ditches have a linear elongated shape (Figure 3: d and g).

**2.4.1 Processing the digital terrain indices - paragraph 5**

The *Sky View Factor Conic filter* was developed to attempt to detect and fill gaps in ditches. This was done by calculating the mean of all the pixels covered by a cone-shaped mask with a radius of 5 metres, which expanded outwards from the examined pixel point in eight directions. If two opposing mean values were both below a threshold, the pixel was given a lower value. This meant that only pixels with strong ditch indicative values in two opposing directions were updated, allowing the filter to avoid updating pixels in cavities or hollows, focusing only on linear geographical properties.

⇒

*removed*

**2.6.1 Noise reduction and gap filling - paragraph 2**

The third step involved taking measures to try to fill gaps in ditches that the model failed to correctly predict. A similar method to the one described in skyviewconic was employed to calculate the mean of cone masks expanding outwards in different directions from the examined pixel. This step also amplified some of the noise that was left, but filling the gaps in the ditches was judged to be more important to help make the next step more effective (Figure 5 b).

The third step involved taking measures to try to fill gaps in ditches that the model failed to correctly predict. This was done by aggregating pixels covered by cone masks expanding outwards in different directions from the examined pixel. This step also amplified some of the noise that was left, but filling the gaps in the ditches was judged to be more important to help make the next step more effective (Figure 5 b).

**2.4.2 Training and validation datasets - paragraph 2**

Using the 11 subsections in the hold-out data for the final experiment, a process called k-fold cross validation was employed (11-fold cross validation in our case). K-fold cross validation is a method where you divide your dataset into folds of similar size and traina model on all but one of your folds (subsections). You then use that subsection to evaluate the results (Wong 2015). Using this technique, shifting which subsection to leave out from the training, allowed us to train 11 different Random Forests classifying models with a large amount of data from the remaining 10 subsections in the hold-out data, producing 11 sub-experiments to evaluate the method on.

**⇒**

Using the 11 subsections in the hold-out data for the final experiment, a process called k-fold cross validation was employed (11-fold cross validation in our case). K-fold cross validation is a method where you divide your dataset into folds of similar size and traina model on all but one of your folds (subsections). You then use that subsection to evaluate the results (Wong 2015) (Figure 4). A new model is trained using the training folds for each iteration (Figure 4). Using this technique, shifting which subsection to leave out from the training, allowed us to train 11 different Random Forests classifying models with a large amount of data from the remaining 10 subsections in the hold-out data, producing 11 independent sub-experiments to evaluate the method on.

Figure 4: new figure to make the cross validation approach more understandable from the perspective of our 11 subsections.

**Multiple places throughout the text**

model ⇒ models and classifier ⇒ classifiers (when referring to the final experiment)

**Multiple places throughout the text**

model ⇒ ditch detector (when referring to the entire ditch detection process (i.e. input variables, model, and post-processing))

**1. Introduction - paragraph 5**

This mapping was used to both train the Random Forests model and to ground-truth the models.

⇒

This mapping was used to both train the Random Forests models and to produce the ground-truth.

**2.2 Digitising the ground truth - figure 1**

The 10 subsections with a white border were used for developing the model before the experiment.

⇒

The 10 subsections with a white border were used for developing input variables and optimising parameters before the experiment.

**2.4.1. Processing the digital terrain indices - paragraph 1**

Developing the Random Forests model involved examining how different kinds of input variables affected the prediction.

⇒

To improve the performance of the Random Forests models, we examined how different kinds of input variables affected the prediction.

**2.5 Building the Random Forests model - Headline**

Building the Random Forests model

⇒

Developing the Random Forests model

**4. Discussion - paragraph 5**

However, small sinks or hilly areas incorrectly classified as ditches by the model were generally removed in the post-processing.

⇒

However, small sinks or hilly areas incorrectly classified as ditches by the Random Forests models were generally removed in the post-processing.

**2.4.1 Processing the digital terrain indices - last paragraph**

*added:*

Overall, 81 input variables were extracted from the terrain indices. We then conducted a sub-experiment to find the best input variables, as well as the optimal number of variables to use. Random Forest's feature importance was used to select the most valuable input variables for different variable amounts. It was found that using 40 input variables produced the best results.

**3. Results and analysis - Table 2**

New metrics after rerun experiment with fewer input variables.

**3. Results and analysis - Table 4**

New feature importances after rerun experiment with fewer input variables.

**Multiple places throughout the text**

81 input variables ⇒ 40 input variables

**2.6.3 Cluster removal - paragraph 1**

A distance calculation was also performed in tandem with this method to find the largest distance of pixels inside each given cluster. This helped to remove sinks and hollows that were not removed by the initial small cluster removal, but that had a shape that indicated that they did not represent a ditch (Figure 8 b).

⇒

A distance calculation was also performed in tandem with this method to find the largest distance of pixels inside each given cluster. This helped to remove sinks and hollows that were not removed by the initial small cluster removal, but that did not have a linear directional characteristic, indicating that they did not represent a ditch (Figure 8 b).

**1 Introduction - paragraph 5**

We hypothesise that by combining the information from all the digital terrain indices using a machine learner (Random Forests), we can improve the detection of the ditches.

⇒

We hypothesise that by combining the information from all the digital terrain indices using a machine learner (Random Forests), we can improve the detection of the ditches, compared to if a single index would be used.

**4. Discussion - paragraph 1**

In our study we have shown that it is possible to locate ditches automatically in high-resolution DEMs (0.5∗0.5 metres, in our case), and that more of the ditches can be detected if the information from several terrain indices (Table 2) is combined through machine learning (Table 3).

⇒

In our study we have shown that it is possible to locate ditches automatically in high-resolution DEMs (0.5∗0.5 metres, in our case), and that more of the ditches can be detected if the information from several terrain indices (Table 2) is combined through machine learning than if indices are used separately (Table 3).

**1. Introduction - paragraph 4**

There are other studies where geographical data has been used with machine learning to map ditches. Roelenset al.(2018) used the Random Forests algorithm to detect ditches in Belgium, using LiDAR point clouds. They used several different input variables where neighbouring points of a specific LiDAR point were examined. Values of all LiDAR attributes were represented for the neighbouring area for both 0.5, 1, 1.5, 2,3 and 4 metre radii. We reproduce these neighbouring area variables when building the ditch locator in our study. In a pilot study, we compared several different algorithms(Random Forests, XGBoost, Naive Bayes, and Support Vector Machines), and it was found that Random Forests produced the most accurate results. Random Forests also computes Mean Decrease in Impurity, or Gini importance, highlighting what input variables are the most important for a given prediction (Menzeet al.2009). Because of these factors, and the fact that it proved to be suitable for Roelenset al.(2018),Random Forests models are used to locate the ditches.

⇒

There are other studies where geographical data has been used with machine learning to map ditches. Roelenset al.(2018) used the Random Forests algorithm to detect ditches in Belgium, using LiDAR point clouds. They used several different input variables where neighbouring points of a specific LiDAR point were examined. Values of all LiDAR attributes were represented for the neighbouring area for both 0.5, 1, 1.5, 2,3 and 4 metre radii. Several other input variables for the model are calculated from the indices to place emphasis on the clear linear directional characteristics of ditches. These characteristics are also used when processing the model's output, in order to remove noise and other inconsistencies in the predictions. We reproduce these neighbouring area variables when building the ditch locator in our study. In a pilot study, we compared several different algorithms(Random Forests, XGBoost, Naive Bayes, and Support Vector Machines), and it was found that Random Forests produced the most accurate results. Random Forests also computes Mean Decrease in Impurity, or Gini importance, highlighting what input variables are the most important for a given prediction (Menzeet al.2009). Because of these factors, and the fact that it proved to be suitable for Roelenset al.(2018),Random Forests models are used to locate the ditches.

**Title**

Ditch detection in high-resolution digital elevation models

⇒

Detecting ditches using machine learning on high-resolution DEMs

**2.6.1 Noise reduction and gap filling - paragraph 2**

The second step for removing noise was to use a custom function to remove pixels that had a semi-high ditch probability, but that lay far away from any other high probability pixels. A threshold value was used to avoid removing pixels with a high enough ditch probability, helping to retain pixels that lay in or close to a ditch.

⇒

The second step for removing noise was to use a custom function to remove pixels that had a semi-high ditch probability, but that lay far away from any other high probability pixels. In this function, a threshold value was used to avoid removing pixels with a high enough ditch probability, helping to retain pixels that lay in or close to a ditch.

**2.6.2 Binarisation with grid zones - paragraph 1**

A mean probability rating was calculated for each 6∗6 grid zone, classifying the entire zone as part of a ditch if the mean probability exceeded 35 %.

⇒

A mean probability rating was calculated for each 6∗6 grid zone, classifying the entire zone as part of a ditch if the mean probability exceeded 40 %. Different zone sizes and probability thresholds were evaluated in a pilot study, and these values were found to produce the best results.

**2.7 Evaluation - Figure 9**

False positives and false negatives that lay within one grid zone (3 m2) of a ditch label were evaluated as true positives and true negatives.

⇒

False positives and false negatives that lay within one grid zone (9 m2) of a ditch label were evaluated as true positives and true negatives.

**2.7 Evaluation - Figure 9**

Illustration of the modified evaluation labels where **a** shows the original result, and **b** shows the modified result. Green marks true positives, red marks false negatives, and blue marks false positives. False positives and false negatives that lay within one grid zone (9 m2) of a ditch label were evaluated as true positives and true negatives.

⇒

Illustration of the modified evaluation labels. Green marks true positives, red marks false negatives, and blue marks false positives. False positives and false negatives that lay within one grid zone (9 m2) of a ditch label were evaluated as true positives and true negatives. **a:** Original results, **b:** Modified results.

**2.6.3 Cluster removal - Figure 8**

Step five and six of the post-process. Black pixels indicate a ditch prediction. **a:** Prediction after binarisation with grid zone probability. **b:** Final binary prediction after cluster removal.

⇒

Step five and six of the post-process. Black pixels indicate a ditch prediction. **a:** Grid zone binarisation **b:** Cluster removal

**2.6.1 Noise reduction and gap filling - Figure 7**

Step three and four of the post-process. Darker pixels indicate a higher ditch probability. **a:** Prediction after custom de-noising. **b:** Prediction after gap filling.

⇒

Step three and four of the post-process. Darker pixels indicate a higher ditch probability. **a:** Custom de-noising. **b:** Gap filling.

**2.6.1 Noise reduction and gap filling - Figure 6**

Step one and two of the post-process. Darker pixels indicate a higher ditch probability. **a:** Raw probability prediction produced by the Random Forests model. **b:** Prediction after bilateral de-noising.

⇒

Step one and two of the post-process. Darker pixels indicate a higher ditch probability. **a:** Random Forests probability prediction **b:** Bilateral de-noising.

**5. Conclusion**

This study investigates how to use digital elevation data together with manually labelled ditches to identify patterns and relationships that can be used for automatic ditch detection. The proposed method significantly outperforms classifying with the digital terrain indices separately. Although the method still has room for improvement, it performs well on the available data. In the future, a more extensive hyperparameter tuning could be performed, because, as Lavesson and Davidsson (2006) states; hyperparameter tuning is often more important than choice of algorithm. Some of the custom input variables could be reused in the future, with Random Forests or other machine learning algorithms. Several of the post-processing algorithms, such as the probability prediction noise removal or the cluster removal from the binary prediction could be generalised for use in any type of pixel classification of ditches.

The work in this study gives an insight into what input variables work well for ditch detection and what input variables do not, as well as an increased understanding of how machine learning can be applied to ditch detection. The proposed method may help in making ditch detection both faster and easier, reducing manual labour and cost. In conclusion, this study has shown that it is possible to use machine learning with digital elevation models to learn patterns that enable robust detection of ditches.

⇒

This study investigates how to use digital elevation data together with manually labelled ditches to identify patterns and relationships that can be used for automatic ditch detection. Although the method still has room for improvement, it performs well on the available data, and significantly outperforms classifying with the digital terrain indices separately. Future work could introduce more input variables and post-processing steps, such as using pathfinding algorithms to fill gaps in the ditch model, or shape indices to remove stream channels from the prediction. Moving towards image segmentation with grids of pixels instead of pixel classification of tabular pixel values could be a suitable future approach.

Many of the input variables could be reused in the future, with Random Forests or other machine learning algorithms. Several of the post-processing algorithms, such as the probability prediction noise removal, and the cluster removal from the binary prediction could be generalised for use in any type of segmentation of ditches. The proposed method may help in making ditch detection both faster and easier, reducing manual labour and cost. In conclusion, this study has shown that it is possible to use machine learning with digital elevation models to learn patterns that enable robust detection of ditches.

**2.4.1 Processing the digital terrain indices - Figure 3**

Removed subfigures d, f, and i and their descriptions. The other figures were renamed in alphabetic order to reflect the changes. References to the figure were updated as well.

**2.5 Developing the Random Forests model - Table 1**

Updated the table to reflect the reduced input variable space.

**3. Results and analysis - Table 3**

Updated with new experiment metrics.

**3. Results and analysis - Table 4**

Updated with new feature importances after new experiment.

**2.2 Digitising the ground truth - paragraph 1**

A high resolution DEM was generated from LiDAR data, see Appendix 5 for specifications of the process

⇒

A high resolution DEM was generated from LiDAR data, see ’Data and codes avail-ability statement’ for specifications of the process

**Multiple places throughout the text**

see Fgure xx ⇒ (Figure xx)

**2.5 Developing the Random Forests model - paragraph 2**

We performed a simple hyperparameter tuning to determine what parameter values for the Random Forests algorithm would yield the best prediction. Evaluating a maximum of 25 input variables for each node, and using 200 trees produced the best results. Setting the class weight to balanced also improved the performance of the classifiers.

⇒

We performed a hyperparameter tuning to determine what parameter values for the Random Forests algorithm would yield the best prediction. Using 300 trees and gini as the splitting criterion, with a minimum of 10 samples per node split, and no artificial class weight or max depth of trees produced the best results.

### 

### Grammatical changes

**2.2 Digitising the ground truth - paragraph 1**

The digitisation of the ditch networks showed that there were 107 km road ditches and 208 km forest/agricultural ditches (Figure 1).

⇒

The digitisation of the ditch networks showed that there are 107 km road ditches and 208 km forest/agricultural ditches (Figure 1).

**2.2 Digitising the ground truth - paragraph 2**

From the ditch mapping, the vector layer was rasterised so it could be compared to the automatically derived ditches from the 0.5∗0.5 m DEM.

⇒

From the ditch mapping, the vector layer was rasterised so that it could be compared to the automatically derived ditches from the 0.5∗0.5 m DEM.

**2.3 Extracting ditches with digital terrain indices - High Pass Median Filter**

A typical false positive with this index are small hollows that also show negative numbers.

⇒

Typical false positives with this index are small hollows that also show negative numbers.

**1 Introduction - paragraph 2**

However, all methods yield false negatives, and Sky View Factor tends to give a large amount of false positives in steep terrain, whereas Impoundment Index produces more false positives in flat terrain if volume is used as a measure, and will falsely detect natural stream channels if dam height is used as a measure. High Pass Median Filter tends to also detect small sinks as ditches.

⇒

Different false positives and negatives can be observed when examining each terrain index manually. Sky View Factor tends to give a large amount of false positives in steep terrain, whereas Impoundment Index and High Pass Median Filter are more difficult to analyse. Due to the difference in false positives and negatives, we want to combine the indices to help prune the weaknesses and amplify the strengths of each index.

**2.2 Digitising the ground truth - paragraph 2**

From the ditch mapping, the vector layer was rasterised so that it could be compared to the automatically derived ditches from the 0.5 ∗ 0.5 m DEM. Because the observed average width of ditches is larger than 0.5 metres, all pixels within a radius of three pixels (1.5 metres) of the vectors were labelled as ditch pixels.

⇒

From the ditch mapping, the vector layer was rasterised so that it could be compared to the automatically derived ditches from the 0.5 \* 0.5 m DEM. Although we have no data about the width of the ditches in the catchment, field observations have shown that the vast majority of ditches are between 0.5 and 3.5 metres wide. Because we wanted to ensure that the model received all ditch pixels correctly in the training phase, we widened the ditches so that all pixels within a radius of three pixels (1.5 metres) of the vectors were labelled as ditch pixels. This ensured that the edges of ditches were included, as well as that the raster converted from vector format actually covered the ditches.

**2.3 Extracting ditches with digital terrain indices - Sky View Factor**

The Sky View Factor was calculated in SAGA GIS, using a search radius of 10 m.

⇒

The Sky View Factor was calculated in SAGA GIS, using a search radius of 10 metres to ensure that ditches were covered, and to not let steep hills close to ditches affect the results.

**2.3 Extracting ditches with digital terrain indices - Impoundment index**

Here, we used dam height calculated with a dam length of 3 m.

⇒

Here, we used dam height calculated with a dam length of 3 metres, to cover the entire ditch which is typically not wider than 3 metres.

*Dear Editor*

*...*