Supporting Information

Below we expand on critical analysis components from the main text. In addition to the descriptions below, the code used to analyze all data is publicly available at https://github.com/pgbrodrick/termiteResilience/.

Appendix S1: Termite mound identification

Identifying termite mounds over large landscapes in LiDAR data is a laborious task, and in this work we sought to automate the process. This automation provided benefits by both reducing expert time required (for manual mound identification) and increasing the consistency of mound classification. Our algorithmic approach, which admittedly may contain similar biases to manually identified methods, will consistently identify termite mounds within and between landscapes in much less time than a manual procedure.

We automated the task of termite mound identification using a convolutional neural network (CNN) to semantically segment the landscape (as in Brodrick et al. 2019). In total, 16,025 manually identified mounds were used for training the CNN, which were (spatially) segregated into training and validation datasets. Manually identified mounds throughout the landscape were initially designated as single points, but we chose to utilize semantic segmentation rather than object detection to identify mounds in order to have a better understanding of characteristics that might contribute to errors. As such, each manually identified mound was given a 5 x 5 meter buffer around the identified point. While true termite mounds are not square and vary in size, this provided enough guidance to train the CNN to characterize a mound.

Our CNN is a commonly used encoder/decoder structure (as in Ronneberger et al. (2016)), though each convolution layer uses a fixed number of 16 filters, rather than growing the number in interior layers, and batch normalization was not used. Raw landscape elevation data and manually identified termite mounds were converted to training data by segmenting data into sample squares using a fixed window size, first of 256 and second of 512 pixels (see below). Equal number of samples were taken from areas with and without mounds. For samples with mounds, a random offset was applied so that each mound was not in the center of the image, enabling the CNN to identify mounds at different locations within the object window. In contrast to the approach by Ronneberger et al. (2016), we did not mirror the interior of each

image sample, but instead limited the CNN loss function to the inner half of the image, enabling context from the exterior of each image to be propagated to the interior without the need to hypothesize about the surrounding layer. To standardize inputs prior to training, each sample was standardized by centering based on the mean elevation of that sample (also repeated during model application).

We trained two separate CNNs with different window sizes - one at 256 x 256 pixels and one at 512 x 512 pixels - and then ensembled the resulting termite-mound likelihoods together, which showed improved predictive performance compared to each model individually. Each model was trained until the validation performance failed to improve for five consecutive epochs, using a categorical cross-entropy loss function. We refer to this ensemble below, and in the main text, simply as the CNN model for convenience (and this framing could be constructed as a single model if desired). More complex methods for managing multiple window size perspectives in CNNs do exist, but this simple method worked well for this problem. After ensembling, a cutoff threshold of 0.99 was applied to the termite likelihood maps, and any block of contiguous pixels with more than 7 pixels greater than that threshold was classified as a mound. Both the threshold and the number of pixels were selected by examining the training prediction maps. This minimum pixel number does not mean that mounds smaller than 7 square meters (pixels) cannot be identified, but instead is the result of the training data comprising 5 x 5 pixel squares rather than specific mounds. As such, areas immediately surrounding mounds may also have been identified as a mound with high likelihood.

To assess the error rates of our model, we selected six landscape subsets of equal size: 3 from the training regions and 3 from independent test regions, which also encompassed the different land-use treatments. Our precision and recall rates were 0.87 and 0.91 for the training set and 0.84 and 0.91 for the test set, respectively. Precision and recall rates give a better sense of the model performance of true and false positive rates due to the strong negative class imbalance and reveal that the model both identifies mounds well and avoids a large number of false positives. While precision rates were lower than recall rates, we consider this slightly favorable because Type II errors were more likely than Type I errors in the manual identification. In total, 526 mounds were present in the training assessment landscapes, and 797 mounds were present in the test assessment landscapes.

Appendix S2: Bootstrap Size Sensitivity

In order to assess the influence of bootstrap size on our mound density measurements, we compared results from 8 different bootstrap subsample sizes, ranging from 400 x 400 pixels (16 ha) to 1800 x 1800 pixels (324 ha). While particularly small bootstrap sizes naturally increase the zero-tendency of the mound density distribution, due to the scarce and clustered distribution of mounds, distributions stabilize when subsample size becomes reasonably large (we selected 1000 x 1000 pixels or a 100 ha subsample for final analysis) (Figure S1).

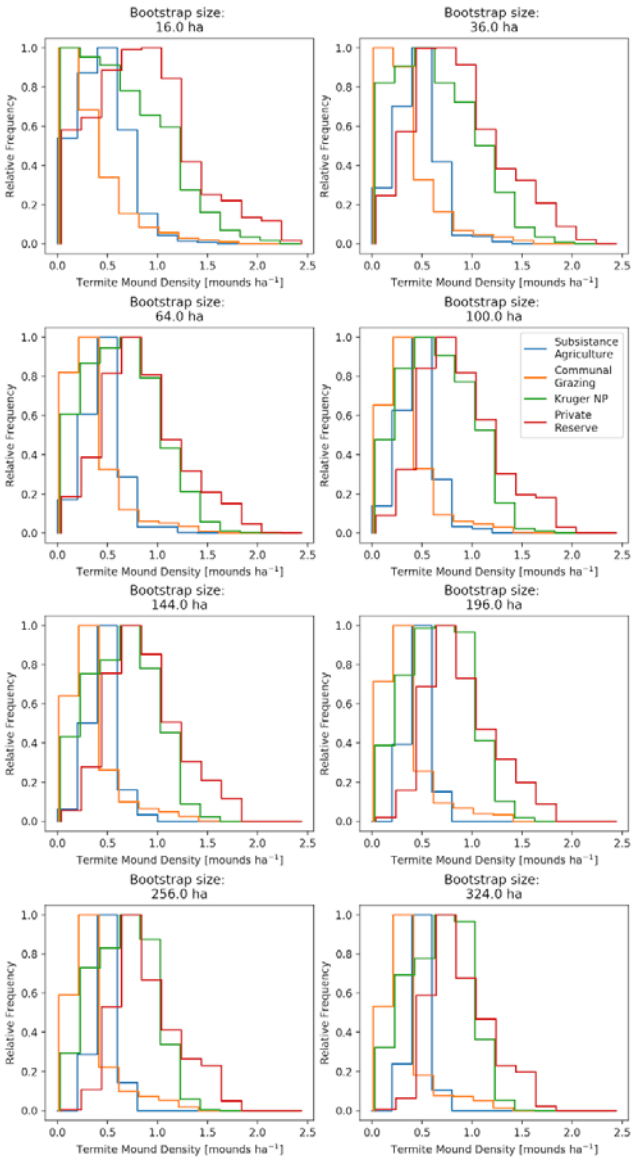

Figure S1. Distribution of termite mound densities using different bootstrap sizes.

Appendix S3: Ripley's K curve minimum polygon size

As discussed in the main text, Ripley's K curves were generated for each polygon. However, several polygons were non-uniform in shape (see Figure 1), and some were also relatively small. This combination can lead to strong boundary condition effects. Figures S2-3 shows the Ripley's K curves (L transformation) for all polygons individually. Polygons that were less than 300 ha in total area (with their numbers highlighted in red) show the non-smooth lines characteristic of insufficient points at all distances to calculate interpretable Ripley's K curves. Polygons under 300 ha in size were therefore omitted from the analysis.

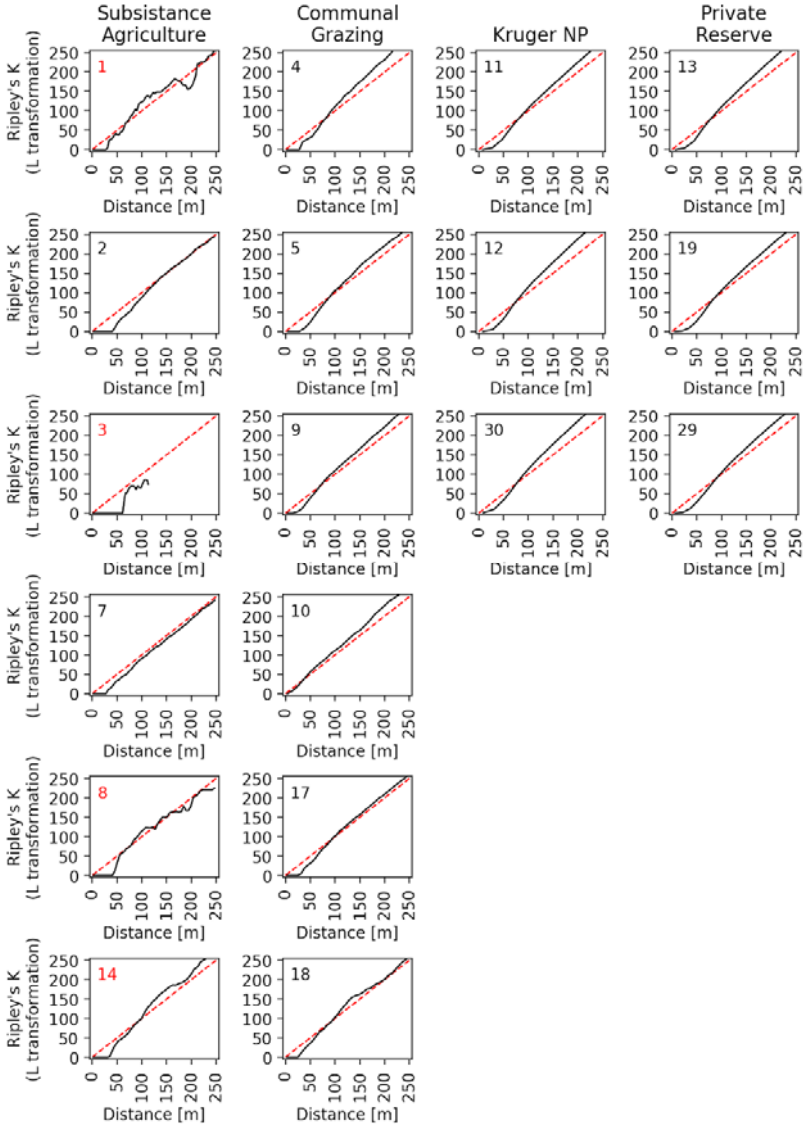

Figure S2. Ripley's K curves for all polygons (continued in Figure S3). Polygon numbers represent assigned polygon numbers, red numbers are polygons <300 ha in extent.

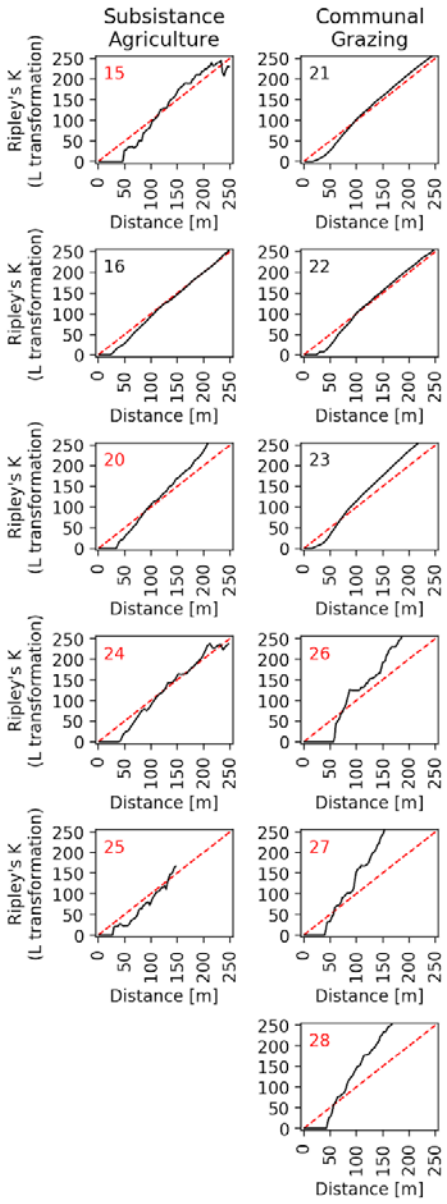

Figure S3. Ripley's K curves for all polygons (continued from Figure S2). Polygon numbers represent assigned polygon numbers, red numbers are polygons <300 ha in extent.