Appendix

Documentation of the data used in this project

Specification Table

Subject	Geography, Environmental Science, Remote Sensing, Computer Science, Climate studies
Specific subject area	Tree Carbon in boreal forests, Remote Sensing modelling, Environmental Science climate processing
Type of data	Raster data (Tif file type) Excel data (CSV) Code files (html)
Method of data acquisition	Predictor and Explana tory data were acquired online Field combustion data was acquired: The ABoVE Fire Emissions Database USGS Earth Explorer: Landsat 8 band 1-9 GEE Climate variables: Precipitation, humidity, minimum and maximum temperature. ASTER DEM: Digital elevation
Data format	Processed
Description of data acquisition	Field combustion dataset were acquired from the above fire emissions derived from Potter et al. 2023 [^1^].
For prediction of Carbon emissions, we used different covariate layers:	• Time-series 2014 and 2015 : USGS Landsat bands [^7^], Optical vegetation indices, GEE Climate variables (Humidity, precipitation, mean, min. and max. air temperature, ASTER DEM
Location of data source	Landsat bands: [^2^] GEE climate dataset: [^3^] ASTER DEM: [^4^]
Accessibility of data	The predicted carbon emission dataset will be found at 30 m resolution in the following repository [^5^] : Repository name: Chinyereruth Data identification number: Still in progress Direct URL to data: Still in progress Detailed code associated with the data analysis is available from the Github repository [^5^]

VALUE OF DATA

- The map provides the amount of carbon emission from trees from Canada's boreal forests, using statistical methods such as regression and machine learning.
- The map will support research on Carbon emission models, and climate change assessment and can be used to inform climate and fire scientists on carbon accounting and reporting, and also recommend measures to reduce carbon emission footprints. It can also be used to compare carbon emissions from other boreal forest zones.
- The methods that will be outlined and code that will be provided can also be replicated in other locations and zones to derive carbon emission maps.

Data Description

Predictors will be provided within the domain of the boreal forest in a folder called the "carbon-emission-modelling-2023.zip". Carbon emitted from each province will be named according to its geographic location (i.e. 059E_14N corresponds to 59E to 60E, 14N to 15N). The "pronvince_carbon_emission_2023.Tif"file will contain carbon emissions for each province within Canada and the "Canada_Carbon_emission_2023.tif"file contains the final carbon emission map for the entire Canada . The detailed code associated with this project will be found in the Github repository (https: //github.com/Chinyereruth/carbon-emission), allowing for the prediction applied here to be replicated. In addition, the code used for this folder will be found in "carbon-emission-Canada-2023- code. Rmd" which will be located in the main GitHub repository folder.

Table 1 Files used in the study for wildfire carbon emission modelling in Canada for the year 2023

File description	File name
Predicted Carbon emitted for Canada 2014	Carbon-emiss-Can_ JanDec_2014-30m.tif
Minimum temperature for Canada 2014	Min-temp-Can_ JanDec_2014-30m.tif
Maximum temperature for Canada 2014	Max-temp-Can_ JanDec_2014-30m.tif
Soil data for Canada 2014	Soil-Can_ JanDec_2014-30m.tif
NDVI data for Canada 2014	NDVI-Can_ JanDec_2014-30m.tif
Tree cover for Canada 2014	Treecov-Can_ JanDec_2014-30m.tif
DEM for Canada 2014	ASTER-dem-Can_JanDec-2014-30m.tif

Experimental Design, Materials and Methods

Training data I used a compilation of field carbon combustion data obtained from the ABoVE Fire Emissions Database (n=467). This field combustion data that were collected across Alberta, Saskatchewan, Manitoba and North Western Territories.

Covariate layers Combustion measurements that were obtained from ABoVE fire database were related to covariates of remotely sensed variables for fire severity, elevation, soil and climate.

Climate variables Climate data for minimum and maximum temperature were obtained from Climate ClimateNA using the GEE climate hub portal for the year 2023. The climate data used here provided the climate point estimate in degrees Celsius which was imported in the spatial environment. The data upscaled to a 30 m grid. This helped to capture the climate impact on vegetation, fuel loads, and fuel moisture, which affect combustion losses.

Environmental variables I used a variety of environmental variables. This includes soils, topography and vegetation category. The soil data was gotten at 250 m resolution but this was upscaled. Topographic variables that will be derived include aspect in degrees, elevation (m),

and slope at a 30 m pixel size.

Remotely sensed variables Here, I derived various remotely sensed vegetation indices from Landsat, and this includes NDVI, the normalized difference infrared index (NDII) and the dNBR. All images used were atmospherically and geometrically corrected using the correction and geometric tool in R.

Model training For the model training, remotely sensed variables, climate and environmental variables were used. Areas of clouds, cloud shadows, and snow were extracted out using an extraction function in R.

Spatial carbon modelling The results from this research are still in progress. Preliminary results are displayed here.

Ethic statement This study did not use human subjects, experiments carried out on animals, and did not also acquire any data from social media platforms.

Data availability

Carbon emission modelling from wildfire disturbances in Canada at 30 m resolution for the year 2023 will be available at Github/ChinyereRuth/carbon-emission.

Goggle Earth Engine (Google Earth Engine) used in deriving dNBR, minimum,mean and maximum tempertures.

GEE code for differenced normalised burn ratio (dNBR)

// Assuming you've uploaded the CSV to Google Drive and have the file ID var myPoint = ee.FeatureCollection('projects/ee-ottahchinyereglcf/assets/Burnedpoint'); // Add points to the map Map.addLayer(myPoint, { color: 'blue' }, 'CSV Points');

//Set up Region of Interest var albertaGeometry = ee.FeatureCollection('FAO/GAUL_SIMPLIFIED_500m/2015/level1').filt
var albertaBounds = geometry.bounds();

// Function to mask clouds using Landsat band. var fun = require('users/ottahchinyere2/biomass:PrepFunctions');

//The preprocess function takes an input start year, and end year, followed by a start month and end month var PreFireImg = fun.preprocessHLS([2013, 2013], [6,9], albertaBounds).median(); var DurFireImg = fun.preprocessHLS([2014, 2014], [6,9], albertaBounds).median(); var PostFireImg = fun.preprocessHLS([2015,2015], [6,9], albertaBounds).median();

//Calculate NBR before, during and after 2015 fire var preNBR = PreFireImg.select('nir').subtract(PreFireImg.select('swir2'))
.divide(PreFireImg.select('nir').add(PreFireImg.select('swir2'))); var durNBR = DurFireImg.select('nir').subtract(DurFireIm
.divide(DurFireImg.select('nir').add(DurFireImg.select('swir2'))).rename('NBR'); var postNBR = PostFireImg.select('nir').subtract(PostFireImg.select('swir2')) .divide(PostFireImg.select('nir').add(PostFireImg.select('swir2'))).

//Calculate dNBR between pre- and post- fire images var dNBR = preNBR.subtract(postNBR).rename('dNBR');

print(dNBR.projection().getInfo()) //Prints out the projection of the dataset to the console

var sample = dNBR.reduceRegions({ collection: myPoint, //The point dataset scale: 30, //This forces it to sample at 30m resolution, This is important as the default is very very very large. crs: ee.Projection('EPSG:4326'), //This is the projection of the original landsat data, so we aren't reprojecting anything reducer: ee.Reducer.median().setOutputs(["dNBR"]), //to use a mean, change this to ee.Reducer.mean() (the setOuput part just renames the band to dNBR instead of 'median' or 'mean') tileScale: 5 //This prevents out of memory errors by splitting the analysis into tiles that can run in parallel (I think) });

print(sample.getInfo()) //Prints out 'sample' to the console. This should contain the same points as the imported geometry, but with the dNBR data appended.

Export.table.toDrive({ //This sends the 'collection' to your own Google Drive. Set the fileNamePrefix and folder options to save to a specific place in your drive. collection: sample, //The 'feature collection' to export description: 'sampleTable', //just a description for the task manager in the GEE console (not important) folder:'GEE4', //folder on your google drive fileNamePrefix: 'burnedarea', //filename essentially fileFormat: 'csv' });

//Add layers to the map: var visualization = { bands: ['red', 'green', 'blue'], min: 0.0, max: 0.3, };

Map.addLayer(PreFireImg, visualization, 'Pre Fire - True Color (321)'); //Map.addLayer(DurFireImg, visualization, 'During 2015 Fire - True Color (321)'); Map.addLayer(PostFireImg, visualization, 'Post Fire - True Color (321)'); Map.addLayer(dNBR,{palette : ['green','white','Red'], min: -0.3, max: 0.3},'dNBR');

//dNBR

2. Prefire Tree cover // Assuming you've uploaded the CSV to Google Drive and have the file ID var myPoint = ee.FeatureCollection('projects/ee-ottahchinyereglcf/assets/AGC-rangerforests_Nov'); // Add points to the map Map.addLayer(myPoint, { color: 'blue' }, 'CSV Points');

//Set up Region of Interest var albertaGeometry = ee.FeatureCollection('FAO/GAUL_SIMPLIFIED_500m/2015/level1').filt
var albertaBounds = geometry.bounds();

// Load NASA MEaSUREs GFCC Tree Canopy Cover dataset for 2010 var tcDataset = ee.ImageCollection('NASA/MEASUR .filter(ee.Filter.date('2010-01-01', '2010-12-31')) .filter(ee.Filter.bounds(albertaBounds)); var treeCanopyCover = tcDataset.select('tree_canopy_cover'); // Print the number of images in the ImageCollection var count = tcDataset.size(); print("Number of images in the ImageCollection:", count); // Load ASTER Global Digital Elevation Model (GDEM) var asterGDEM = ee.Image("projects/sat-io/open-datasets/ASTER/GDEM").clip(albertaBounds); print(asterGDEM) // Visualization parameters for tree canopy cover var treeCanopyCoverVis = { min: 0.0, max: 100.0, palette: ['ffffff', '00ff00'] // Adjust the palette as needed }; // // AGCpoints2 code // var AGCpoints2 = asterGDEM.addBands(treeCanopyCover.median()) // .reduceRegions({ // collection: myPoint, // scale: 30, // crs: ee.Projection('EPSG:4326'), // reducer: ee.Reducer.median().setOutputs(["tree_canopy_cover"]), // tileScale: 5 // }); var AGCpoints2 = asterGDEM .select(['b1']) .rename(['GDEM']) .addBands(treeCanopyCover.median()) .reduceRegions({ collection: myPoint, scale: 30, crs: ee.Projection('EPSG:4326'), reducer: ee.Reducer.median(), tileScale: 5 }) // .select(['GDEM', 'tree_canopy_cover']); // Export the 'sample' feature collection to Google Drive Export.table.toDrive({ collection: AGCpoints2, description: 'AGCpoints2', folder: 'outputFolder', fileNamePrefix: 'GDEM_TreeCover', fileFormat: 'CSV' }); // Add the DEM layer to the map for Alberta only Map.addLayer(asterGDEM, { min: 0, max: 3000, palette: ['0000ff', 'ffffff'] // Adjust the palette as needed }, 'ASTER GDEM - Alberta'); // Add the tree cover layer to the map for Alberta only Map.addLayer(treeCanopyCover.median(), treeCanopyCoverVis, 'Tree Canopy Cover - Alberta'); // Add the Region of Interest (Alberta) to the map Map.addLayer(albertaBounds, {color: 'FF0000'}, 'Alberta'); // Center the map on the Region of Interest Map.centerObject(albertaBounds, 3);

// Calculate slope var slope = ee.Terrain.slope(asterGDEM);

// Calculate Topographic Position Index (TPI) var tpi = asterGDEM.subtract(ee.Image(asterGDEM.reduceNeighborhood({ reducer: ee.Reducer.mean(), kernel: ee.Kernel.square(5, 'pixels'), }))).rename('TPI');

// Add layers to the map for slope, TPI, and TWI Map.addLayer(slope, {min: 0, max: 45, palette: ['blue', 'yellow', 'red']}, 'Slope'); Map.addLayer(tpi, {min: -100, max: 100, palette: ['blue', 'white', 'red']}, 'TPI');

3. Minimum, maximum and mean temperature code

Assuming you've uploaded the CSV to Google Drive and have the file ID var myPoint = ee.FeatureCollection('projects/ee-ottahchinyereglcf/assets/Burnedpoint'); // Add points to the map Map.addLayer(myPoint, { color: 'blue' }, 'CSV Points');

//Set up Region of Interest var albertaGeometry = ee.FeatureCollection('FAO/GAUL_SIMPLIFIED_500m/2015/level1').filt var albertaBounds = geometry.bounds();

// Read in Image Collection and get first image var agera5_ic = ee.ImageCollection('projects/climate-engine-pro/assets/ce-ag-era5/daily'); var agera5_i = agera5_ic.first(); var Proj = ee.Projection('EPSG:6931'); // Print first image to see bands print(agera5_i);

// Filter Image Collection by date range (May 2014) var startDate = ee.Date('2014-05-01'); var endDate = ee.Date('2014-05-31');

var agera5_ic_may = agera5_ic .filterDate(startDate, endDate);

// Calculate monthly maximum, minimum, and mean temperature var agera5_monthly_max = agera5_ic_may.max(); var agera5_monthly_min = agera5_ic_may.min(); var agera5_monthly_mean = agera5_ic_may.mean();

// Define a color palette for temperature visualization var temp_palette = ['blue', 'purple', 'cyan', 'green', 'yellow', 'red'];

// Visualize monthly maximum, minimum, and mean temperature for May 2014 Map.addLayer(agera5_monthly_max.select(' {min: -10, max: 50, palette: temp_palette}, 'Monthly Max Temperature (May 2014)');

Map.addLayer(agera5_monthly_min.select('Temperature_Air_2m_Min_24h').selfMask().subtract(273.15), {min: -10, max: 50, palette: temp_palette}, 'Monthly Min Temperature (May 2014)');

Map.addLayer(agera5_monthly_mean.select('Temperature_Air_2m_Mean_24h').selfMask().subtract(273.15), {min: -10, max: 50, palette: temp_palette}, 'Monthly Mean Temperature (May 2014)');

// Mypoint code var myPoint_max = agera5_monthly_max .select(['Temperature_Air_2m_Max_24h']) .rename(['Max_Temperature']) .reduceRegions({ collection: myPoint, scale: 30, crs: Proj, reducer: ee.Reducer.median(), tileScale: 5 });

var myPoint_min = agera5_monthly_min .select(['Temperature_Air_2m_Min_24h']) .rename(['Min_Temperature']) .reduceRegions({ collection: myPoint, scale: 30, crs: Proj, reducer: ee.Reducer.median(), tileScale: 5 });

var myPoint_mean = agera5_monthly_mean .select(['Temperature_Air_2m_Mean_24h']) .rename(['Mean_Temperature']) .reduceRegions({ collection: myPoint, scale: 30, crs: Proj, reducer: ee.Reducer.median(), tileScale: 5 });

// Print the resulting FeatureCollections print('Max Temperature:', myPoint_max); print('Min Temperature:', myPoint_min); print('Mean Temperature:', myPoint_mean);

// Add points with temperature information to the map Map.addLayer(myPoint_max, { color: 'red' }, 'Max Temperature Information'); Map.addLayer(myPoint_min, { color: 'green' }, 'Min Temperature Information'); Map.addLayer(myPoint_mean, { color: 'blue' }, 'Mean Temperature Information');

print(myPoint_mean);

// Export the feature collections to Google Drive Export.table.toDrive({ collection: myPoint_max, description: 'MaxTemperature', // Description for the task manager folder: 'outputFolder2', // Specify the folder on your Google Drive fileNamePrefix: 'MaxTemperature', // Set the filename fileFormat: 'CSV' // Choose the file format });

Export.table.toDrive({ collection: myPoint_min, description: 'MinTemperature', // Description for the task manager folder: 'outputFolder2', // Specify the folder on your Google Drive fileNamePrefix: 'MinTemperature', // Set the filename fileFormat: 'CSV' // Choose the file format });

Export.table.toDrive({ collection: myPoint_mean, description: 'MeanTemperature', // Description for the task manager folder: 'outputFolder2', // Specify the folder on your Google Drive fileNamePrefix: 'MeanTemperature', // Set the filename fileFormat: 'CSV' // Choose the file format });