Real Forest Fires

Real Forest Fires

Scenario and Objectives

Forest fires pose serious ecological and economic risks, particularly in heavily forested regions of Eastern Europe, where a mix of dense cover creates conditions for ignition and rapid fire propagation. This project models forest fire spread across a random area of Belarus using satellite-derived vegetation data from the [Hansen Global Forest Change dataset] (2000 Tree Cover %), enabling a realistic simulation of burn dynamics across a 100×100 cell terrain patch (~3×3 km).

Using a Monte Carlo simulation framework, the model incorporates:

- *Real tree density data* to determine per-cell fire spread probability,
- *Wind direction and strength parameters* to simulate directional spread,
- *Stochastic fire ignition* to reflect real-world uncertainty in ignition points.

The simulation's core objectives are to:

- *Identify high-risk zones* where fire is most likely to spread or reach.
- *Compare outcomes under varying wind conditions* (e.g., East vs. North, weak vs. strong).
- *Assess the impact of firebreaks* as a mitigation strategy.
- *Quantify model uncertainty* using confidence intervals across multiple runs.

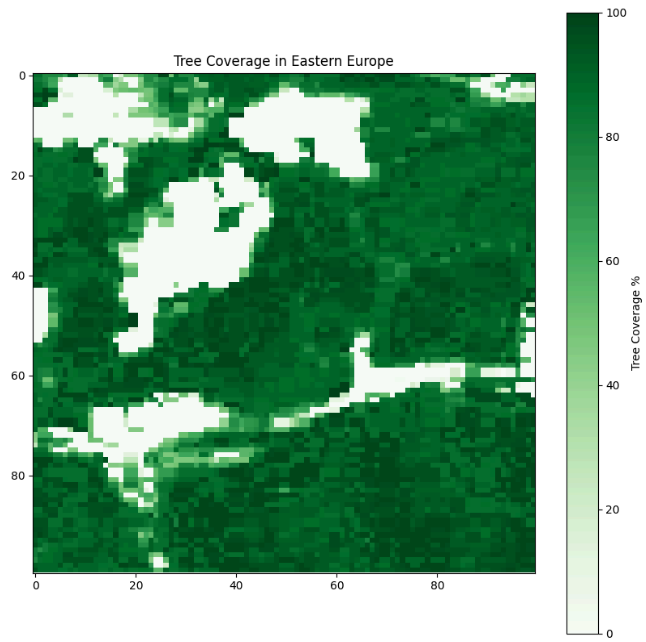

Figure 1. *Selected Region in Belarus.* It has a 100 grid size, and is very high in tree density.

Simulation Framework

- Fires start at random locations with probability proportional to tree density.

```
def start_fire(self, i, j):
    if np.random.random() < self.forest[i, j]:  # ← tree-density-proportional probability
def start_random_fire(self):
    potential_cells = np.where(self.forest > 0.3)
    ...
    i, j = potential_cells[0][idx], potential_cells[1][idx]:
```

- Each burning cell attempts to spread fire to its eight immediate neighbors (N, NE, E, SE, S, SW, W, NW).

```
self.directions = [(0, 1), (-1, 1), (-1, 0), (-1, -1), (0, -1), (1, -1), (1, 0), (1, 1)]
```

- ***Spread Probability*** = Tree Density * Wind Modifier * Randomness
  - Proportional to the tree density of the *target* cell.
  - Modified by *wind alignment* between the wind vector and the direction of spread.
  - Includes a *random component* to capture natural unpredictability.

```
probabilities = np.minimum(1.0, densities * wind_effect * 2) if wind_effect==1 else np.minimum(1.0,
densities * wind_effect)
random_values = np.random.random(len(ni))
catch_fire = (random_values < probabilities) & (~self.burned[ni, nj]) & (~self.burning[ni, nj])
```

- Wind boosts spread in aligned directions and reduces it against the wind: `angle_diff = np.abs(np.radians(self.wind_direction) - spread_angle_rad)`
- Unburned → Burning → Burned (1-step transitions)

```
self.burned[burning_i, burning_j] = True  # burning → burned
self.burning[ni[catch_fire], nj[catch_fire]] = True  # unburned → burning
```

- The simulation ends when no cells remain burning or a preset maximum number of time steps (100) is reached.

Key Assumptions

- Wind direction and strength remain spatially and temporally constant.
- Only immediate neighbors determine fire spread; no long-range spotting or ember transport is modeled.
- The model assumes no firefighting or mitigation during the burn.

- Tree density is fixed and does not change as fire progresses.

These assumptions are valid for short-duration simulations over compact regions. Still, they may limit applicability in complex terrain-induced wind variation, vegetation regrowth, or diversity dynamics during the fire.

Parameters and Implementation

1. Tree Density: floating point value between 0 and 1; The raw 0–100 tree cover percentages are linearly scaled to a 0–1 density scale via `self.forest = forest_data / 100.0.`
2. Wind Effects: 0° = East, 90° = North, 180° = West, 270° = South; and has strength from 0 to 1. Wind influences fire spread by modifying the base probability (tree density) according to the angle difference between wind direction and spread direction, captured by an exponential decay function: `wind_effect = np.exp(-angle_diff / (np.pi * self.wind_strength));` Here, angle difference is the different between wind direction and spread direction; π normalizes the range of angles to [0, π]; and `wind_strength` (from 0 to 1) controls how sharply wind influences fire behaviour. In practice, this captures real-world fire dynamics, where wind accelerates fire along its path and suppresses it in opposing directions.

The simulation is developed in Python using object-oriented design and vectorized operations for performance. It has the defined class `ForestFireSimulation`, where `step()` handles a single timestep of fire spread using NumPy vectorization. There are also three state arrays: `self.burning` (tracks currently burning cells); `self.burned` (tracks burned cells); `self.affected` (tracks all burned cells and their neighbors).

The method `def` **`run_monte_carlo`** repeats independent runs and produces a risk map showing how likely each cell will be affected by fire. Firebreaks are modeled by setting tree density to 0 in selected rows/columns of the forest array.

Results and Analsysis

The selected region (Figure 1) in central Belarus demonstrates heterogeneous vegetation structure, with patches of high tree density (≥0.8) interspersed with low-density or barren cells (<0.2). This spatial variation is critical in shaping fire spread dynamics, creating natural barriers and channels through which fire can propagate more efficiently.

Controlled Fire Ignition Test

To isolate fire behavior under neutral environmental conditions, I performed a controlled ignition test with no wind (0° wind, strength = 0.0). The fire was manually ignited at a fixed point, and its spread was monitored across several steps. The results served to validate the base model performance.

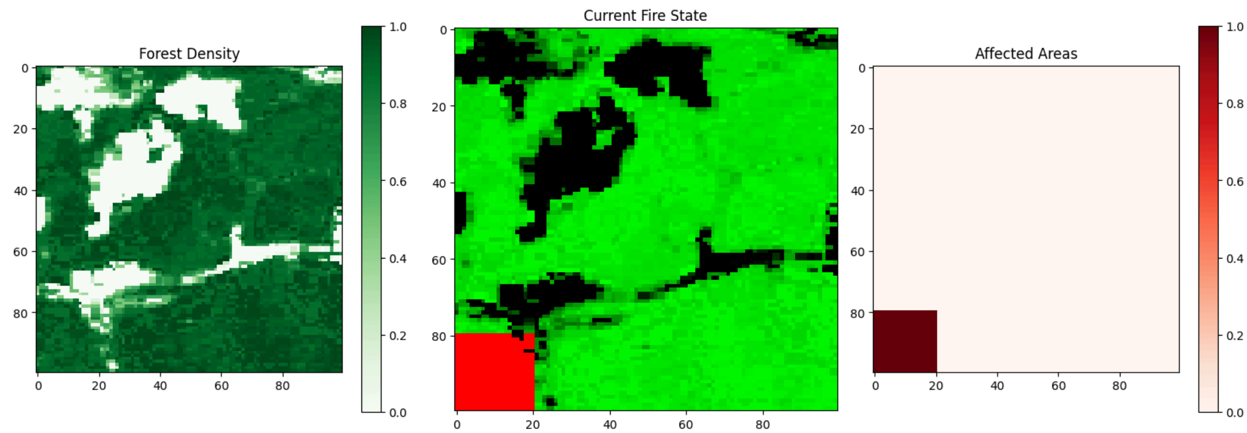

Figure 2. *Controlled Fire Test*

Wind Scenario Comparison

To evaluate the effect of wind on fire dynamics, I tested two wind configurations:

1. East Wind: Direction = 0°, Strength = 0.5
2. North Wind: Direction = 90°, Strength = 0.7 (stronger)

Each configuration was run for 50 Monte Carlo simulations. The resulting risk maps display the probability that each cell is affected (burned or adjacent to a burn) across all runs.

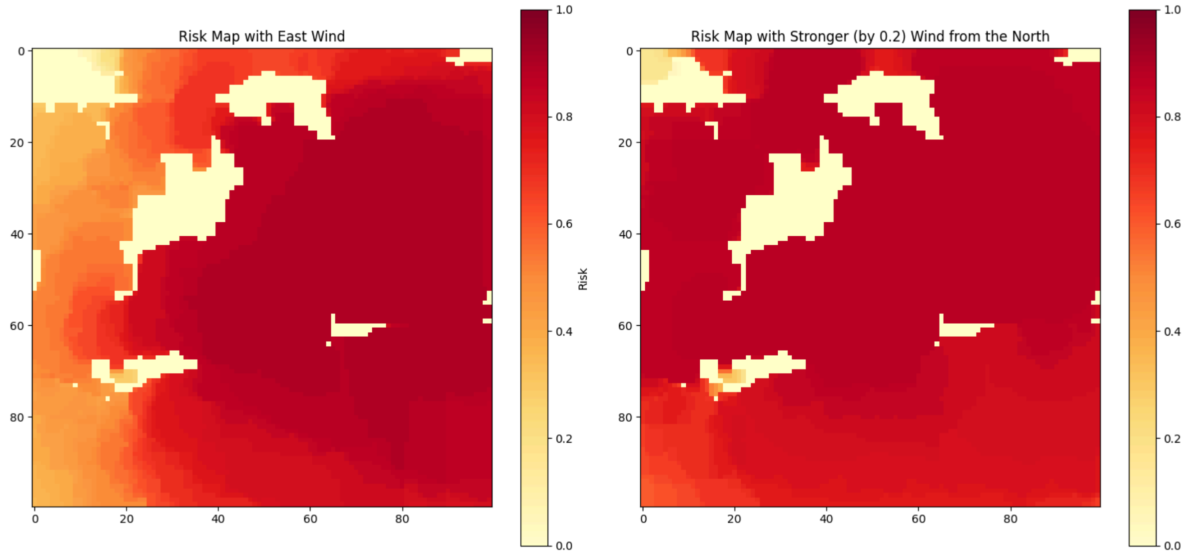

Figure 3. *Risk Map Comparison.* I see that the stronger North Wind has slightly darker red areas, meaning it spreads faster

The North Wind scenario showed a more aggressive northward spread, impacting upper forest areas with greater severity due to the stronger wind.

Scenario	Avg. Risk	95% CI Width
East Wind (0.5)	0.6919	±0.0988
North Wind (0.7)	0.7615	±0.0943

The stronger North Wind produces a higher average risk and a shift in the spatial distribution of affected areas. Confidence intervals overlap, indicating a statistically insignificant outcome difference between wind scenarios.

Firebreak Effectiveness

A horizontal firebreak was created to evaluate a mitigation strategy by setting tree density to 0 in rows 50–55. This artificial barrier mimics controlled clearings used in real-world fire management. In this experiment, wind conditions matched the East Wind scenario (0°, 0.5 strength) and were run for 50 Monte Carlo simulations.

- Average risk before firebreak: 0.6919
- Average risk after firebreak: 0.3868
- Risk reduction: 44.09%

The firebreak dramatically curtailed northward spread, effectively compartmentalizing the forest and reducing risk to the northern sector. This validates the practical utility of firebreaks as a policy-relevant intervention.

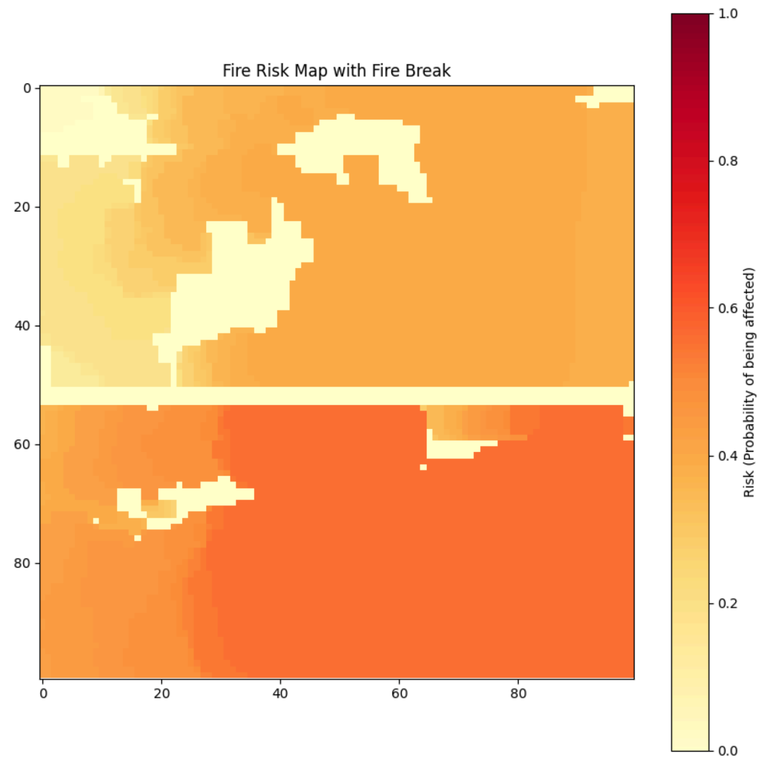

Figure 4. *Horizontal Fire Break.* The number of fires is significantly lower than in the East Wind Scenario.

Burn Extent Distribution

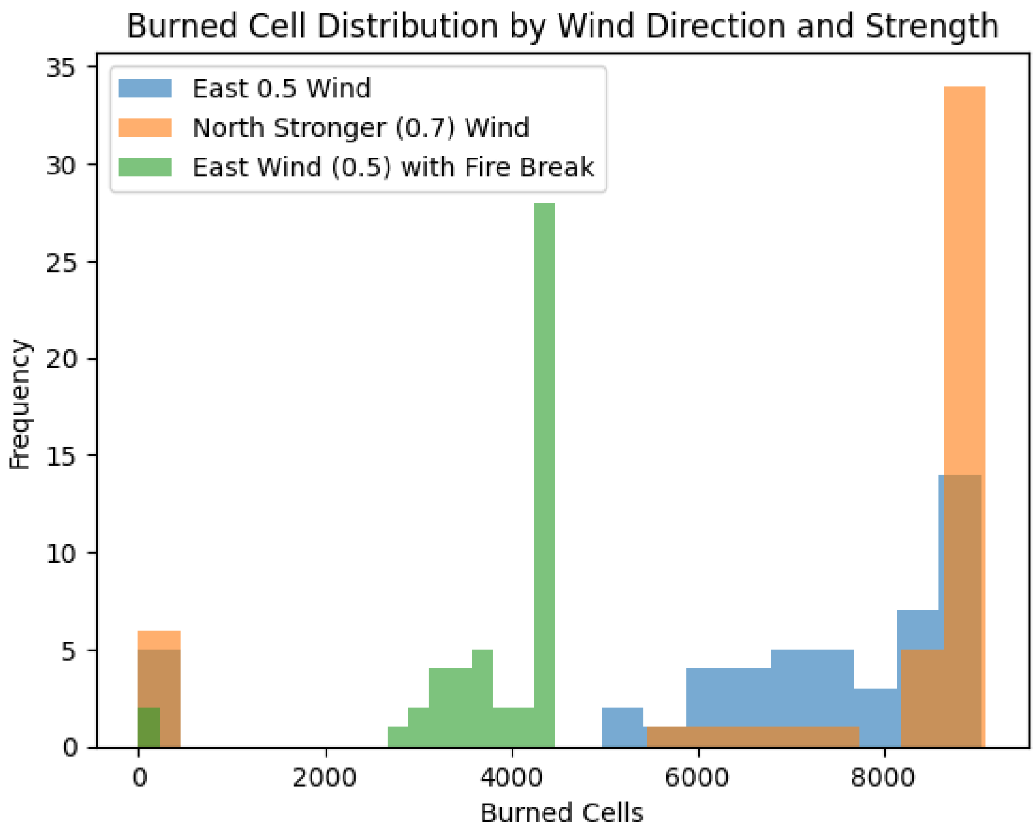

Figure 5. *Histogram of Burned Cells*. Blue is East 0.5 Wind, orange is North 0.7 Wind, and green is the fire break strategy.

1. East Wind (blue) exhibited a bimodal distribution, indicating two common outcomes: limited burns (early containment) and large-scale burns (uncontained spread).
2. North Wind (orange) showed a right-shifted, skewed distribution, suggesting more simulations resulted in extensive fires.
3. Firebreak scenario (green) displayed a tight, left-skewed distribution, confirming consistently low burn extent post-intervention.

Percolation Threshold Validation

Forest fire dynamics exhibit a well-known connection to percolation theory. In a 2D grid, percolation refers to the probability that a connected path spans the system.

According to classical percolation theory, the critical probability (threshold) for large-scale connectivity on a square lattice is approximately 0.5927[1]. This implies that, for a uniform grid, when the tree density exceeds ~59%, there is a high probability that a fire can propagate through the forest.

I ran a percolation threshold test using uniform-density forests ranging from 0.1 to 1.0 to validate this. In each case, the fire was initiated in the center cell and allowed to spread without wind or firebreaks.

- At low densities, fire rarely spreads far from the ignition point.
- No sharp increase was observed in the mean burn percentage at any density.
- My empirical threshold was approximately 1.0000, higher than the theoretical 0.59 due to finite grid size effects, lack of long-range spotting, conservative spread rules, and uniform windless conditions.

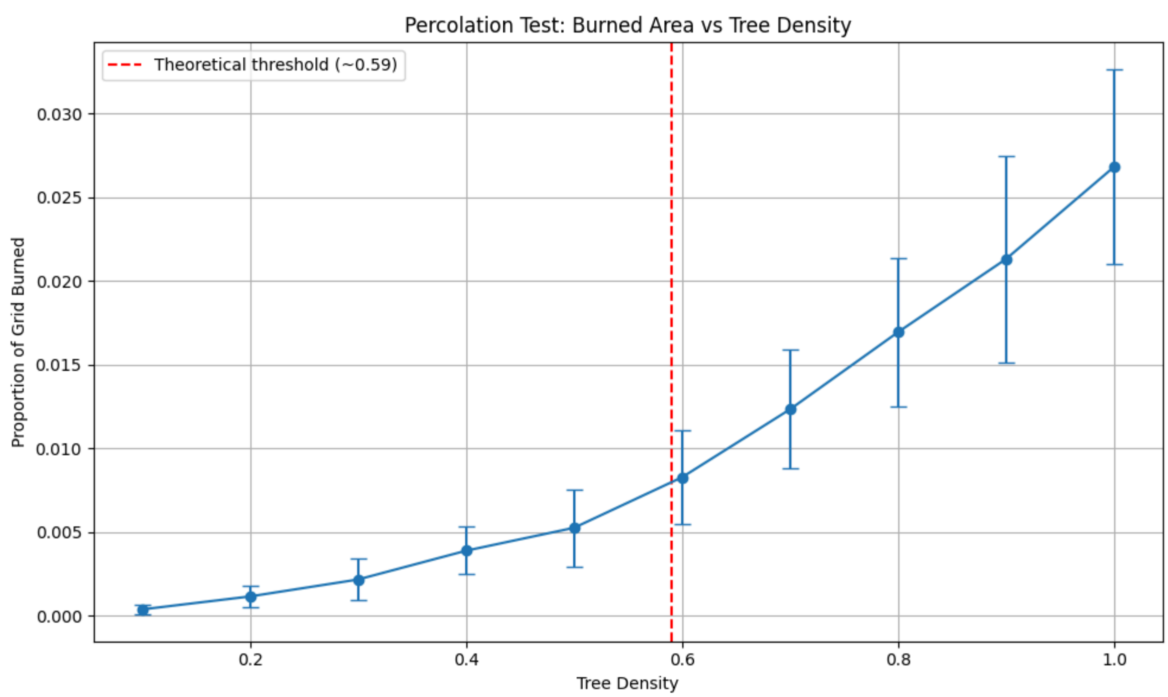

Figure 6. *Percolation Test Results*

[1]Go to the table in Percolation in 2D section, square.

Executive Summary

This project modeled fire spread using a high-resolution, data-driven Monte Carlo simulation in a 100×100 forested area of Eastern Europe. The simulation incorporated satellite-derived tree density data, wind parameters, and randomized ignition to generate actionable fire risk maps.

Based on the findings, I recommend

1. Construct Firebreaks Strategically: Implement a horizontal firebreak near the region's center (rows 50–55). The simulations show this could reduce fire risk by over 49%.
2. Adapt Resource Allocation to Wind Conditions: More resources should be deployed in the southern zones under predicted northerly winds.
3. Prioritize Protection for Persistent High-Risk Areas: Use the aggregated risk maps to identify vulnerable forest patches, corridors, and wind paths. These regions should be prioritized for evacuation drills, early detection sensors, and additional thinning.
4. Reduce Uncertainty with Further Simulations: Current confidence intervals average ±9%. Running 50–100 more simulations would reduce this uncertainty and improve precision in cell-level risk predictions.

AI Statement

I used Claude for coding help; for example, my logic in the step function was incorrect because I did not understand the transitions, so I used it to debug it. I used ChatGPT to help structure the report and check its grammar, such as technical summaries, documentation structure, and statistical analysis. I also used AI to help me identify the formula for wind spread (exponential) and teach me about percolation theory.