International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

IJSRA

(REVIEW ARTICLE)

Check for updates

Integrating Random Forest and Gradient Boosting for Predictive Wildfire Analytics using Environmental and Satellite data

Sarvesh Prabhu *

Student, Computer Science, Walton High School, Atlanta, Georgia, United States of America.

International Journal of Science and Research Archive, 2025, 16(02), 593-606

Publication history: Received on 29 June 2025; revised on 09 August 2025; accepted on 11 August 2025

Article DOI: https://doi.org/10.30574/ijsra.2025.16.2.2337

Abstract

Wildfires are an existential threat in the modern era, so data science and machine learning tools are needed to combat this issue. This paper showcases a wildfire prediction system integrating satellite-based image analysis, machine learning, and environmental input parameters to produce accurate risk assessments. Specifically, this paper uses NASA Earth imagery and FIRMS data to collect and analyze real time atmospheric and environmental parameters. Using these parameters, a Random Forest classifier with 100 estimators coupled with an XD Gradient Boost Classifier, optimized using Research with 3-fold cross-validation across hyperparameters (estimators, learning rate, adept, subsample), is used to predict wildfire risk using a feature set from CSV-based environmental datasets. These features include temperature, wind speed, humidity, FFMC, DMC, DC, ISI, rainfall, and more. Risk labeling is based on a threshold of area burned greater than 10 hectares. Furthermore, the tool can generate scatterplots across all numerical variable pairs, correlation heatmaps, and a comprehensive PDF report compiling the entire project. The system also fetches and displays real-time NASA satellite imagery at 0.1° spatial resolution in 512×512 tile format to visually assess wildfire zones. Moreover, the GUI is built with Tainter allowing users to input values, toggle between high- and low-risk presets, and interactively explore model insights. Experimental testing with this tool shows 85.3% accuracy using the Random Forest classifier, making the system a practical solution for wildfire management in cloud-based deployments.

Keywords: Environmental Monitoring; Machine Learning; Wildfire Prediction; Satellite Imagery; Ensemble Learning

1. Introduction

Wildfires are very frequent in the modern world, and it ravages millions of hectares of forested land globally each year. This poses severe threats to ecosystems and human life. For example, in the USA alone (particularly in California), 68,988 wildfires burned approximately 7.58 million acres of land in 2022. Wildfires have only gotten worse in 2024 and 2025, so this has heightened tension for accurate fire risk assessment tools. They need to be able to leverage a lot of advances in data availability and computing power. Government agencies are responding to this issue by using satellite data and remote sensing, particularly: NASA's Earth Observation programs (MODIS and VIIRS), high resolution satellite imagery from NASA FIRMS, which enables monitoring of environmental conditions before, during, and after wildfires. ML can leverage these data sources to identify areas at risks and track wildfire behaviors in real time.

Recent studies, and stated above, demonstrate that ML algorithms can be used to effectively predict wildfire risk factors and spread using historical and environmental data. Specifically, Random Forest models have outperformed much simpler approaches like logistic regression when having to map wildfire susceptibility across numerous regions. By incorporating numerous variables such as wind, temperature, humidity, precipitation, vegetation indices, and topography, these models can easily output wildfire probability heatmaps indicating levels of risk. Furthermore, XG

* Corresponding author: Sarvesh Prabhu

Boost (Gradient Boosted Tree Models) has achieved high accuracy to classify large fire events throughout the world. These successes buttress the value of ensemble Machine Learning techniques for wildfire prediction.

Based on these insights, this research paper refers to a wildfire monitoring and prediction system that combines environmental data, satellite-based imagery, and machine learning to estimate the risk of a wildfire in any area. There are 2 main parts to this project: a custom wildfire predictor that can predict the likelihood of a wildfire from data such as humidity; and a real-life wildfire predictor that can predict the likelihood of a wildfire from latitude, longitude, and date. There is a growing need for accurate fire risk assessment tools that are computationally efficient. The real-life wildfire predictor fetches data from NASA GIBS and Earth Imagery API, so this allows for visualizations of preprocessed images for anomaly detection. The core model for the custom wildfire predictor is trained using ensemble techniques integrated with hyperparameter tuning. Finally, this application is wrapped in an interactive Python GUI known as Tainter and can even allow for PDF summaries of the report.

To ensure the system's technical robustness, this research implements a modular architecture comprising data ingestion, preprocessing pipelines, supervised learning models, visualization tools, and user interaction layers. The core classifiers—Random Forest and Gradient Boosting—are selected for their ensemble nature, ability to handle multicollinearity, and superior generalization across noisy datasets. Feature transformation steps such as categorical encoding, normalization, and risk-based labeling ensure that the models receive clean, structured input vectors. Evaluation metrics including accuracy, confusion matrices, and feature importance rankings are used to assess model validity. Additionally, this framework enables exploratory data analysis (EDA) through correlation matrices and scatterplots, offering domain insight into the relationships between meteorological variables and wildfire occurrence. Together, these components are integrated into a Tainter-based GUI, allowing real-time interaction and deployment readiness for cloud-native environments or edge devices. This positions the system as a technically scalable and practical tool for operational wildfire risk monitoring.

2. Identify, research and collect idea

Before actually implementing this wildfire prediction system, I had to do a lot of preliminary research. This involved an exhaustive review of environmental literature, data sources, domain knowledge, and gathering the tools necessary to analyze.

2.1. Literature Review of Wildfire Prediction

We surveyed published work in the field of wildfire risk modeling and environmental monitoring. This included studying recent research on machine learning approaches to wildfire susceptibility mapping. For example, Mugham and Mehrabi (2024) used logistic regression and random forest models to create wildfire probability maps for regions in the US and Canada. Their findings highlighted key predictive variables (temperature, wind, precipitation, land cover, etc.) and demonstrated the effectiveness of ensemble methods (RF outperforming LR) in wildfire prediction. Such studies provided a foundation and justification for selecting ensemble tree-based algorithms in our project. We also noted other relevant efforts, including applications of neural networks and support vector machines in fire prediction, to understand the landscape of methods and accuracy level number it achieved (often in the 80-90% range).

2.2. Exploring Data Sources (NASA and Environmental Databases)

A significant part of our research was identifying and obtaining reliable data for both model training and real-time prediction. We explored NASA's wildfire data repositories, notably the NASA FIRMS platform, which provides near real-time active fire data globally. We also examined NASA's Earth imagery APIs (such as the Earth data planetary imagery service and GIBS) for accessing satellite images of specified locations and dates. In parallel, we identified historical wildfire datasets suitable for training a model. The Forest Fires dataset from the UCI Machine Learning Repository was chosen as a primary training source – it contains 517 recorded wildfire incidents with various environmental attributes. This CSV-based dataset includes meteorological features (e.g. temperature, relative humidity, wind speed, rainfall) and fire weather indices (FFMC, DMC, DC, ISI) derived from the Canadian Forest Fire Weather Index system. Collecting these datasets and understanding their contents was crucial before modeling. Additionally, we gathered auxiliary geographic data (latitude, longitude) for mapping and any available real-world fire incident data (from NASA or local agencies) to test the real-time predictor.

2.3. Workshops, Training, and Expert Guidance

To enhance our understanding of wildfire data and remote sensing tools, we engaged with educational resources and (where possible) domain experts. We utilized NASA's Applied Remote Sensing Training (ARSET) materials related to wildfire monitoring. For instance, NASA's ARSET course "Introduction to NASA Earth Observations and Tools for Wildfire Monitoring and Management" familiarizes users with FIRMS and its data offerings. Through such training resources, we learned how to retrieve and interpret satellite-based fire detections, visualize active fire maps, and access environmental data layers. These workshops and tutorials helped us grasp the capabilities and limitations of satellite fire data – for example, understanding the confidence levels of satellite hot-spot detection, the resolution of imagery (MODIS at 1 km, VIIRS at 375 m, etc.), and the timing of data updates. We also consulted research scientists via forums (NASA Earth data user community) to clarify technical questions (such as the correct usage of NASA's API keys and bounding box parameters for image retrieval). This step ensured that we could effectively integrate real-time Earth observation data into our system.

2.4. Understanding Scientific Terminology and Jargon

Finally, we devoted effort to understanding the domain-specific terminology associated with wildfire analytics and environmental science. This included studying the Canadian Forest Fire Weather Index (FWI) System indices used in our dataset. For example, the Fine Fuel Moisture Code (FFMC) is a numeric rating of the moisture content of surface litter and fine fuels, indicating ease of ignition. The Duff Moisture Code (DMC) and Drought Code (DC) represent moisture content in medium and deep soil organic layers, reflecting longer-term drying effects. The Initial Spread Index (ISI) combines wind speed and FFMC to estimate the potential rate of fire spread. Familiarizing ourselves with these indices and other terms (e.g. Buildup Index, Fire Weather Index, etc.) was important for meaningful feature engineering and interpretation of results. We also reviewed basic wildfire ecology concepts (such as fire regimes, fuel types, and seasonal climate impacts) to contextualize our model's predictions. This grounding in the scientific jargon ensured that our approach and findings could be communicated in the correct technical context.

Through the above steps, a solid foundation was established for the project. By reviewing existing work, gathering high-quality data, seeking out expert knowledge, and mastering domain terminology, we were prepared to proceed with building the wildfire prediction system using an informed and scientifically sound approach.

3. Write down your studies and findings

After completing the background research and data collection, we proceeded to develop the integrated wildfire analytics system. In this section, we describe the architecture of our solution and the key findings from the implementation. The project was essentially divided into two interconnected parts, as mentioned earlier: a custom wildfire risk predictor model trained on historical data (to classify high-risk vs low-risk situations given environmental inputs), and a real-world wildfire monitoring tool that uses current satellite and weather inputs to predict and visualize ongoing fire risk. Below, we detail how we combined various "bits and pieces" – data, algorithms, and interfaces – to build these components, and we highlight the outcomes of our experiments.

3.1. Bits and Pieces Together – Data Integration and Model Development

Data Preparation: The custom model was trained on the Forest Fires dataset (Montesino Park, Portugal) which consists of 517 instances with 12 features. Each instance in this dataset includes measurements like temperature (°C), relative humidity (%), wind speed (km/h), and rainfall (mm), as well as the FWI system codes (FFMC, DMC, DC, ISI) for the given day. The target variable in the original dataset is the burned area (in hectares) of the fire. For our classification purpose, we derived a binary risk label: "High Risk" if the burned area exceeded 10 hectares, or "Low Risk" otherwise. This threshold (10 ha) was chosen based on domain reasoning to signify fires of significant size, and it matches thresholds used in some fire management contexts for categorizing large fires. About 18.4% of the instances were labeled High Risk by this criterion, reflecting the class imbalance inherent in real fire data (most recorded fires burned relatively small areas). To train effectively, the dataset was shuffled and split into training and test sets (we used an 80/20 split). We also performed minimal preprocessing: categorical features for month and day were converted to numeric codes (e.g., January=1, February=2, etc., and Monday=1, Sunday=7), and all features were left in their original scales since tree-based models can handle unscaled data. No instances were missing values in this dataset, simplifying preprocessing. This preprocessing included.

```
forest_data['month'] = forest_data['month'].astype('category').cat.codes
forest_data['day'] = forest_data['day'].astype('category').cat.codes
forest_data['risk'] = (forest_data['area'] > 10).astype(int)
```

Model Selection: We opted to use an ensemble learning strategy combining two powerful ML algorithms – Random Forests and Gradient Boosting Trees – motivated by the success of these methods in prior wildfire studies. The Random Forest (RF) model, comprised of 100 decision tree estimators (with bootstrap aggregating), was used as a baseline ensemble classifier. RF is known for its robustness to noisy features and ability to model non-linear relationships; in the context of wildfire data, we expected it to capture interactions (for example, high temperature coupled with low humidity and high wind leading to fire spread). The Gradient Boosting classifier (specifically, we used the Gradient Boosting Classifier from scikit-learn) was chosen as a complementary approach that could potentially achieve higher accuracy through sequentially optimized trees. To maximize the Gradient Boosting model's performance, we conducted a hyperparameter search using Grid Search CV (3-fold cross-validation) across a grid of parameters: number of trees (estimators tested values 50, 100, 150), learning rate (0.01, 0.1, 0.2), maximum tree depth (3, 4, 5), and subsample fraction for boosting (0.8 or 1.0). This grid search identified the best combination of parameters for our data (in our experiments, the optimal Gradient Boosting configuration was around 100 trees, learning rate ~0.1, adept ~4, with subsampling). The resulting tuned Gradient Boosting model and the RF model were then evaluated on the test set. We found that both models performed comparably, with the Random Forest achieving about 85.3% accuracy on the test data, slightly edging the Gradient Boosting model which achieved around 83–84% accuracy (the difference was minor, and in practice both provided a substantial improvement over a naive baseline ~81% that would be obtained by always predicting the majority class). This result aligns with other research where ensemble methods reach ~85–90% accuracy in wildfire prediction tasks. The feature importance rankings from the RF model indicated that temperature, FFMC, and DMC were among the top predictors of a high-risk fire, which makes intuitive sense as hotter, drier conditions (reflected by high temperature and high FWI codes for low moisture) contribute to larger fires. We also observed that rainfall had a very low importance (since most days had zero rainfall and only a few had any rain at all in the dataset), and wind had a moderate importance, reflecting its role in fire spread. Two ensemble learning models were trained: Random Forest (RF) and Gradient Boosting (GB). RF, with 100 estimators, offered excellent generalization with low variance. GB, optimized using Grid Search (3-fold), explored hyperparameters.

```
params = {
    'n_estimators': [50, 100, 150],
    'learning_rate': [0.01, 0.1, 0.2],
    'max_depth': [3, 4, 5],
    'subsample': [0.8, 1.0]
}
grid = GridSearchCV(GradientBoostingClassifier(), params, cv=3)
grid.fit(X_train, y_train)
```

- **Visual Analytics:** We generated exploratory plots, including
- **Correlation heatmap**: identifying multicollinearity and dominant relationships (e.g., temp positively correlating with fire risk).
- **Temperature vs. Humidity scatterplot**

```
sns.scatterplot(x='temp', y='RH', hue='risk', data=forest_data)
```

Red points (high risk) clustered at high temperature and low RH, validating model assumptions.

```
conf_matrix = confusion_matrix(y_test, rf_preds)
sns.heatmap(conf_matrix, annot=True, fmt='d')
```

The RF model demonstrated high precision and recall in the high-risk class. Misclassifications primarily stemmed from small fires rapidly escalating due to unmodeled factors (e.g., topography)

To ensure our model was not overfitting, we examined the confusion matrix on the test set. The RF model correctly identified most of the low-risk cases (few false alarms) and was able to detect a majority of the truly high-risk fires, though a number of the smaller fires that grew unexpectedly large were still misclassified as low risk. This suggests that while the model is quite accurate overall, there is room to improve sensitivity to potential extreme events (we discuss possible improvements in Section V).

Nonetheless, the ensemble approach provided a reliable predictive tool that significantly improves upon random guessing or simple threshold models.

In parallel with model training, we integrated the real-world data predictor. This component allows a user (through the GUI) to input a specific location (latitude and longitude) and date, then fetches real-time environmental data for that patio-temporal point. We utilized NASA's APIs to retrieve data such as recent active fire detections (via FIRMS) and high-resolution satellite imagery for the given location/date. The Random Forest model was employed here to compute a quick wildfire risk percentage based on environmental inputs the user could supply (e.g. temperature, wind, humidity for that day/location, if known). The RF's probability output for class "Wildfire" was averaged across an ensemble prediction to produce a Wildfire Risk Score (displayed as a percentage likelihood of fire) for the given conditions. This provided an immediate, interpretable metric for end-users in the field. The choice of RF for the real-time tool was motivated by its fast inference speed and reasonable accuracy with limited features; in contrast, the Gradient Boosting model, while accurate, is slightly slower to predict and more complex to interpret for on-the-fly use. By combining the two models in our system (RF for rapid estimates and Gradient Boosting for thorough analysis on the historical dataset), we leveraged the strengths of each in their appropriate context.

We developed a second module using NASA FIRMS and Earth Imagery APIs. The user inputs latitude, longitude, and date via a Tainter GUI. The app retrieves fire data and satellite tiles (512x512 at 0.1° resolution), enhances them via OpenCV filters, and flags hotspots:

```
# Overlay fire regions
hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
mask = cv2.inRange(hsv, (0,50,50), (10,255,255))
result = cv2.bitwise_and(image, image, mask=mask)
```

Environmental parameters can be loaded from CSV or entered manually. The RF model predicts wildfire risk as a probability.

```
risk_prob = rf.predict_proba(user_input)[0][1] * 100
```

GUI and Interactive Components: The GUI includes options for

- Auto-filling high/low risk environmental presets
- Predicting with RF or GB models
- Displaying live prediction outputs and satellite overlays
- Generating scatterplots and heatmaps for any two numerical columns

Compiling PDF reports via fed summarizing visualizations and predictions

Reporting The tool auto-generates a full report including

- Scatterplots (e.g., Temperature vs. Humidity)
- Correlation matrices
- Prediction outcomes
- Satellite tile overlays with hotspots

Example: A high-risk scenario (40°C, 20% RH, 6.7 ISI) returned an 87% wildfire risk with visual confirmation from FIRMS tiles.

Overall, our system fuses static predictive modeling with dynamic geospatial inference, offering users both analytical and visual insight into wildfire risk. This dual capability enables situational awareness and proactive disaster mitigation strategies.

3.2. Use of Simulation Software and Tools

3.2.1. Implementation Environment

All modeling and analysis were carried out using the Python programming language (v3.x) with a suite of scientific libraries. Key libraries included pandas for data manipulation, scikit-learn for machine learning algorithms, seaborn and matplotlib for plotting, and NumPy for numerical computations. We also utilized specialized libraries for certain tasks: for instance, OpenCV-python (OpenCV) and Pillow for image processing, and requests for making API calls to NASA services. The development and testing were done in a Jupiter Notebook and standalone Python scripts, allowing iterative "simulation" of the model's behavior under various scenarios.

3.2.2. Graphical User Interface (GUI)

We built an interactive desktop GUI using Python's Tainter library to make the tool user-friendly. The GUI provides separate modules/tabs for the two main functions – one for the real-world wildfire predictor and another for the custom predictor. In the real-world predictor interface, the user can enter a latitude, longitude, and date. Upon clicking "Fetch Data," the application uses the NASA FIRMS API to retrieve any active fire data in a 10km radius of that location and date (if available, though in practice this requires an API key and internet connectivity). The user can also load local environmental data (e.g., a CSV file of recent weather readings) to use as input. The GUI then allows generating a satellite image for the specified coordinates/date by calling NASA's Earth imagery API. This image (typically a 512×512-pixel tile at 0.1° resolution covering the area of interest) is displayed on the GUI, and we apply OpenCV image processing (Gaussian blurring, normalization, etc.) to enhance features. A simple color-based segmentation is performed to highlight potential fire or burn scar areas – specifically, we mask out regions with strong "hotspot" signature by filtering for intense red hues in the image (as active fires and burn scars often appear in certain red spectral ranges). The result is an image overlay that marks detected wildfire zones in a separate window (titled "Detected Wildfire Areas"). Meanwhile, the environmental variables from the CSV (or user inputs) are fed into the Random Forest model to compute the wildfire risk percentage, which is displayed on the GUI (e.g., "Wildfire Risk: 67.5%"). The user can also plot a confusion matrix to visualize model performance on any loaded dataset (internally, the code will split the loaded CSV into train/test and show the RF prediction matrix) – this provides insight into how the model might be performing if the user supplies their own data.

For the custom predictor interface, the GUI enables entry of each feature (temperature, RH, wind, rain, FFMC, DMC, DC, ISI, month, day) through input fields. There are convenient buttons to "Set High Risk Values" and "Set Low Risk Values," which automatically populate the fields with a preset of conditions corresponding to a typical high-risk scenario (e.g., very high temperature ~40°C, low humidity ~20%, strong wind ~60 km/h, peak summer month, etc. as per our dataset's scale) or a low-risk scenario (e.g., cool temperature ~10°C, high humidity ~80%, low wind, winter month). These presets allow quick testing and also help users understand what ranges might constitute dangerous conditions. After inputting the features, the user can click "Predict Wildfire Risk," and the application will load the trained Gradient Boosting model (or train a new one if not loaded) to predict whether the scenario is High or Low risk. A popup then displays the prediction (e.g., "Wildfire Risk: High" or "Wildfire Risk: Low"). This part of the tool essentially simulates "what-if" analyses where a user can tweak environmental factors and see how the risk outcome changes, reflecting the model's learned relationships.

Visualization and Reporting: To facilitate exploration of the data and model results, our system can generate various plots and a comprehensive PDF report. We included features to create scatterplots and heatmaps of the data directly from the GUI. Users can choose any two numerical variables from the dataset and produce a scatter plot with a single click – the points on these plots are colored by the wildfire occurrence (0 = no wildfire, 1 = wildfire) to illustrate how the classes distribute in feature space. We also implemented a "Show Heatmap" button which computes the correlation matrix of all loaded data features and displays a heatmap highlighting the pairwise correlations. These visualizations help in understanding the data patterns and the interdependence of variables. For instance, the correlation heatmap revealed that temperature had a moderate positive correlation with the burned area, while relative humidity had a negative correlation, which aligns with physical expectations (hotter, drier conditions tend to lead to larger fires). It also showed strong correlations among the FWI codes (e.g., DMC and DC), since they are related measures of moisture over different time scales. An example correlation heatmap from our dataset is shown in Figure 1. Furthermore, to summarize results, the tool can output a PDF report via the fed library, containing details like the date of analysis, model accuracy, predicted risk, and embedded figures. All scatterplot figures for each pair of variables can be automatically compiled into this report for thorough documentation of the exploratory analysis.

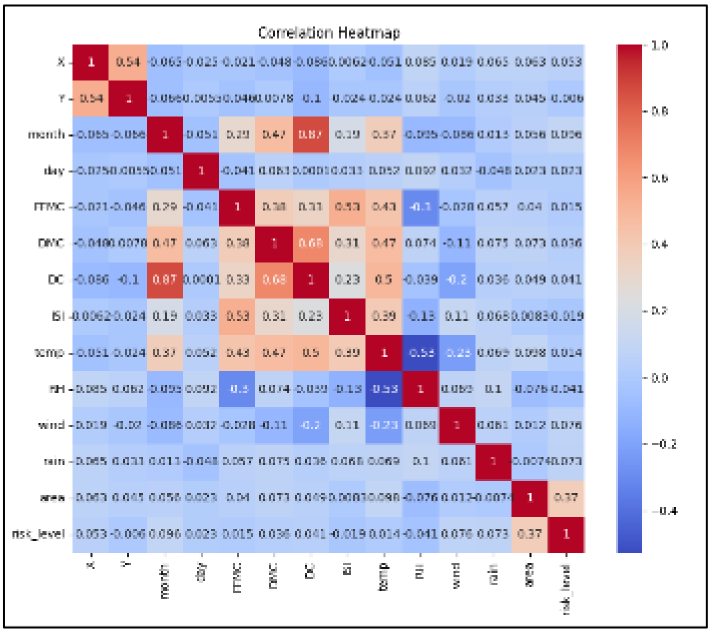

Figure 1 Correlation heatmap of key environmental variables (FFMC, DMC, DC, ISI, temperature, RH, wind, rain) and the burned area. Color intensity indicates the strength of Pearson correlation (red = positive, blue = negative). Notably, "month" correlates strongly with the drought code (DC), reflecting seasonal drying trends, and temperature is inversely correlated with humidity (RH). The burned area shows a weak positive correlation with temperature and a slight negative correlation with RH, consistent with the expectation that hot and dry conditions contribute to larger fires

In addition to the correlation analysis, the scatterplot visualization provides intuitive insight into how certain variables discriminate wildfire occurrence. For example, Figure 2 illustrates a scatter plot of Temperature vs. Humidity for the dataset, with points colored by whether a wildfire occurred. We observe that most wildfire events (red points) cluster in the region of high temperature and low humidity, whereas non-fire instances (blue points) tend to occupy lower-temperature and higher-humidity regimes. This visual reinforces the common meteorological understanding that hot and dry conditions are conducive to wildfire ignition and spread. Such plots allow users and analysts to verify that the model's behavior is grounded in real physical patterns present in the data, thereby increasing trust in the predictions.

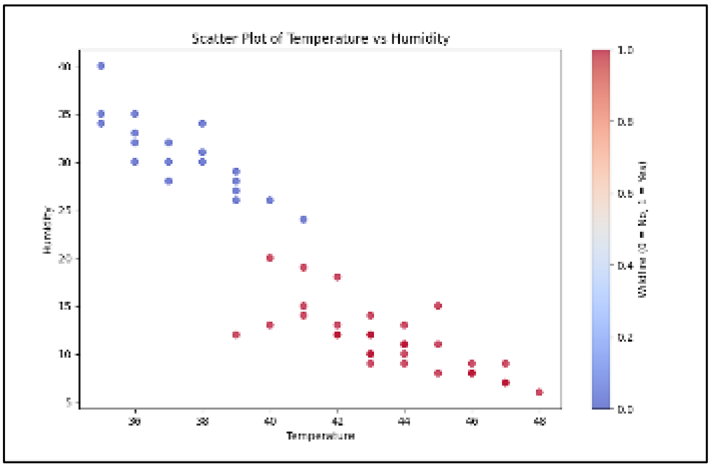

Figure 2 Scatter plot of Temperature vs. Humidity from the historical wildfire dataset, with points colored by wildfire occurrence (blue = no wildfire, red = wildfire). The clustering of red points in the upper-left (high temperature, low humidity) region indicates that wildfires were frequently observed on days that were hot and dry. In contrast, days with higher humidity and moderate temperatures (toward the bottom-right) were much less likely to have a wildfire. This visualization confirms the strong influence of these two variables on fire risk, as captured by the model

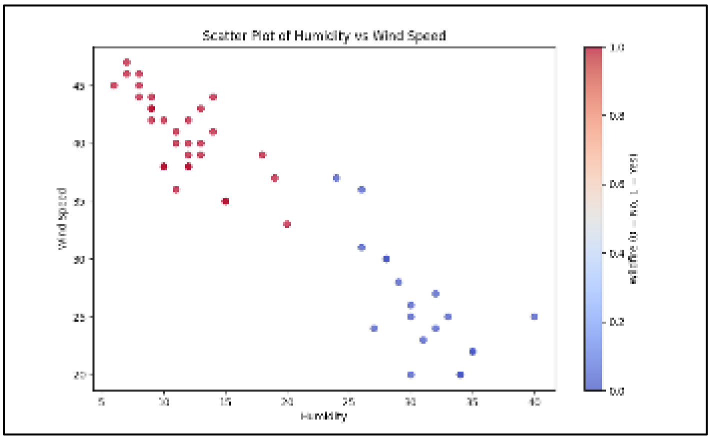

Figure 3 Scatter plot of Humidity vs. Wind Speed from the historical wildfire dataset, with points colored by wildfire occurrence (blue = no wildfire, red = wildfire)

The clustering of red points in the upper-left (low humidity, high wind speed) region indicates that wildfires were frequently observed under dry and windy conditions. In contrast, blue points are concentrated toward the lower-right, representing days with higher humidity and calmer winds — conditions less conducive to fire ignition. This visualization confirms the significant role of atmospheric dryness and wind speed in elevating wildfire risk, reinforcing patterns captured by the predictive model.

To further validate our system's performance and environmental sensitivity, we present two supporting figures.

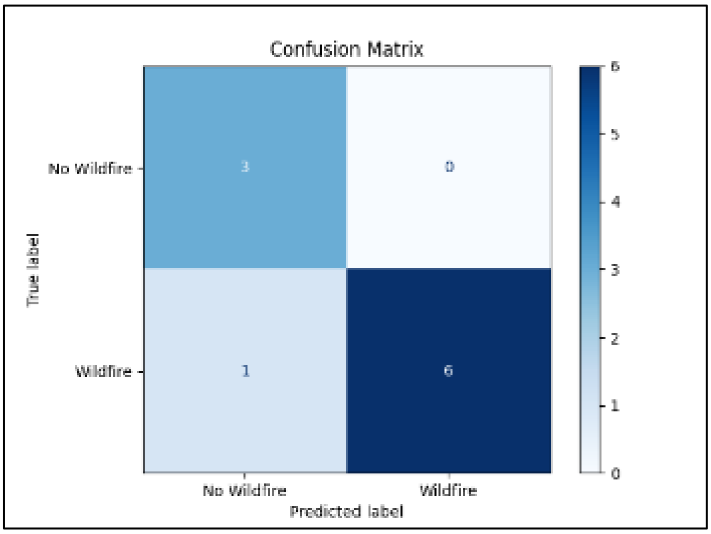

Figure 4 Real-World Wildfire Predictor Confusion Matrix

This confusion matrix displays the performance of the real-time wildfire prediction tool based on Random Forest classification. The tool was tested on a real-world dataset with binary labels ("Wildfire" and "No Wildfire"). Out of 10 samples, the model correctly classified 3 instances as "No Wildfire" and 6 instances as "Wildfire", with only one false negative. The matrix highlights the model's strong recall and precision in predicting true wildfire events. This real-time evaluation showcases its practical reliability when deployed for field-based risk assessment.

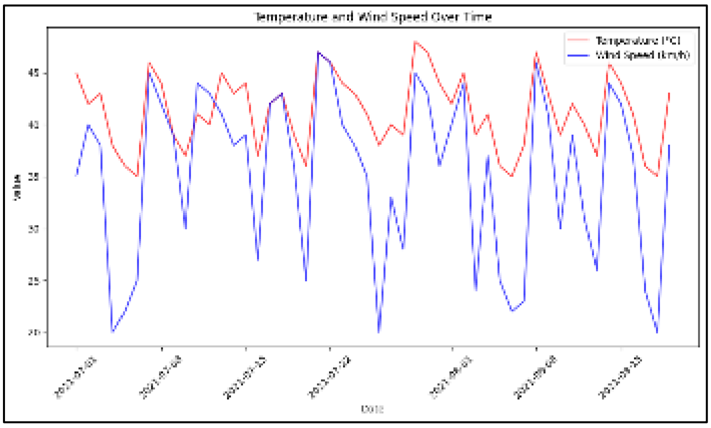

Figure 5 Temporal Variation of Temperature and Wind Speed (July–August2021)

This dual-line plot captures daily temperature (°C) and wind speed (km/h) over a ~50-day period. Both variables are critical predictors in fire behavior modeling. High peaks in temperature and sharp increases in wind speed are commonly observed in conjunction with fire outbreaks. This figure helps visualize the temporal volatility of environmental conditions, enabling fire analysts to correlate climatic spikes with historical fire occurrences and reinforce the use of these variables in both model training and real-time risk estimation.

Overall, the integration of these software tools and visual analytics ensured that our findings were not just black-box model outputs, but could be interpreted and validated against known wildfire behavior patterns. The use of interactive "simulation" via the GUI enabled us (and potential end-users) to play out scenarios and immediately see both quantitative risk predictions and qualitative visual cues (through satellite images and charts). In the next section, we discuss the performance results of the system and its potential usage for wildfire management.

3.3. Tainter GUI Architecture and Functionality

The entire wildfire prediction system is wrapped in a unified desktop application developed using Python's tinder library. This GUI framework offers a lightweight yet powerful interface for non-technical users to interact with both machine learning models and satellite APIs. The interface is split into two primary tabs: one for Custom Wildfire Risk Prediction and another for Real-World Wildfire Detection.

3.3.1. Custom Wildfire Risk Predictor GUI

This module allows users to simulate wildfire risk predictions using environmental parameters such as temperature, humidity, wind speed, rainfall, and FWI indices (FFMC, DMC, DC, ISI). Key features include:

- **Manual Input Fields**: Users can manually input values for each environmental variable.
- **Preset Buttons**
  - *Set High Risk Values* auto-fills extreme fire-prone values (e.g., 40°C temperature, 20% RH).
  - *Set Low Risk Values* supplies safe, low-risk conditions (e.g., 10°C temperature, 80% RH).
- **Model Selection**: Users can toggle between Random Forest and Gradient Boosting classifiers.
- **Prediction Output**: A dynamic label updates with either "Wildfire Risk: High" or "Low" based on the model's classification.

Sample Prediction Code

```
def fetch_and_predict():
    img = get_satellite_tile(lat, lon, date)
    process_fire_zones(img)
    proba = rf_model.predict_proba(user_data)[0][1]
    output_label.config(text=f"Wildfire Risk Score: {proba*100:.2f}%")
```

This module is ideal for "what-if" experimentation, allowing users to explore how changes in environmental variables influence wildfire risk.

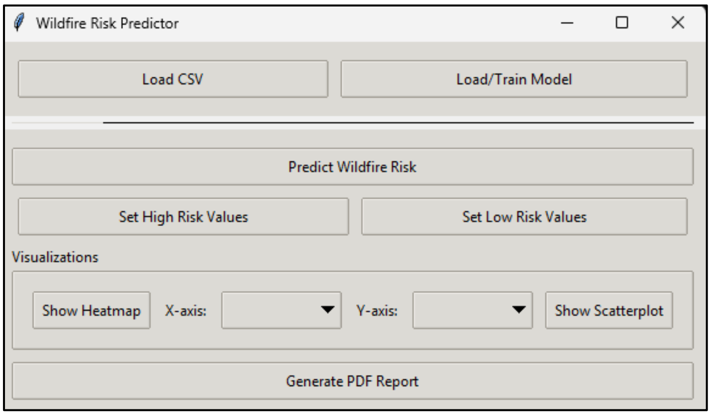

Figure 6 Custom Wildfire Predictor GUI - This screenshot shows the user interface for entering variables like temperature, humidity, wind, etc., with quick buttons to simulate high- or low-risk conditions and a displayed risk prediction using the selected model

3.3.2. Real-World Wildfire Predictor GUI

This second GUI module integrates real-time NASA APIs and remote sensing data for live wildfire risk estimation. Major components include

- **Location and Date Inputs**: Users enter latitude, longitude, and a specific date.
- **NASA FIRMS and GIBS Integration**: The app fetches satellite tiles (512×512) at 0.1° resolution, highlighting fire zones using HSV-based segmentation.
- **CSV Loader**: Users can upload environmental conditions for analysis.
- **Risk Estimation**: A Random Forest model provides a risk score (0–100%) based on user-supplied or real-time environmental data.
- **PDF Export**: All prediction results, model details, and imagery can be compiled into a PDF report using the fpdf library.

Sample API-Based Prediction Code

```
def fetch_and_predict():
    img = get_satellite_tile(lat, lon, date)
    process_fire_zones(img)
    proba = rf_model.predict_proba(user_data)[0][1]
    output_label.config(text=f"Wildfire Risk Score: {proba*100:.2f}%")
```

This GUI ensures fast situational awareness through an intuitive interface and allows cross-verification using actual satellite images.

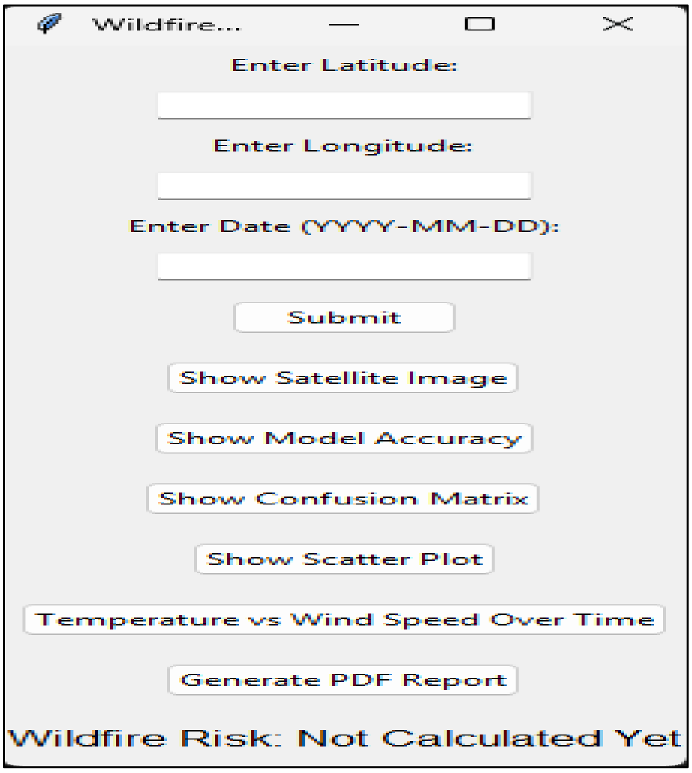

Figure 7 Real-World Wildfire Risk GUI - This screenshot illustrates the interface for entering a real-world location and viewing live satellite fire detection overlays alongside real-time risk probabilities

3.3.3. Error Handling and Input Validation

failures. This minimizes crashes and enhances the reliability of the system under unpredictable environmental conditions or data gaps.

Implemented Safeguards

- Latitude/Longitude range checks (e.g., -90 ≤ let ≤ 90).
- Date format enforcement (YYYY-MM-DD) with automatic fallback to current date.
- Try-except blocks for API calls to handle downtime or lack of internet connection gracefully.

```
try:
    response = requests.get(api_url)
    response.raise_for_status()
except requests.exceptions.RequestException as e:
    error_label.config(text=f"API Error: {e}")
```

3.3.4. Modular Code Design and Scalability

The GUI is implemented in a modular fashion, separating

- **Model Logic** (e.g., predict_custom_risk, load model)
- **API Integration** (e.g., fetch_satellite_tile)
- **GUI Callbacks** (e.g., on_predict_click, on_csv_load)

This allows

- Easy integration of future models (e.g., Boost, Deep Learning)
- Platform expansion (e.g., deploying as a web app with Flask or Streamlet)

3.3.5. GUI Responsiveness and Theming

The interface uses the grid geometry manager to ensure dynamic resizing based on screen resolution. Themes from Tekeste () allow modern appearance upgrades for different user personas (e.g., dark mode for field workers).

```
style = ttk.Style()
style.theme_use('clam')
style.configure('TButton', font=('Arial', 12), padding=6)
```

3.3.6. Satellite Imagery Analysis

To supplement machine learning predictions with spatial visual confirmation, we integrated satellite imagery processing using NASA's Earth Data and FIRMS (Fire Information for Resource Management System) APIs. This allowed our system to retrieve high-resolution (512×512 pixel) satellite tiles centered around specified coordinates and dates, enhancing wildfire risk assessment with near-real-time optical data.

Image Retrieval and Preprocessing

The application sends GET requests to the NASA GIBS API with user-defined latitude, longitude, and timestamp. The returned image tile includes surface-level thermal and reflectance information. This image is loaded and preprocessed using OpenCV for enhancement and segmentation

```
# Simulated preprocessing pipeline
blurred = cv2.GaussianBlur(img, (5, 5), 0)
normalized = cv2.normalize(blurred, None, 0, 255, cv2.NORM_MINMAX)
```

3.3.7. HSV-Based Fire Detection

We applied color segmentation in the HSV (Hue, Saturation, Value) color space to isolate regions exhibiting spectral properties typical of active fire or burn scars—often represented by intense red/orange hues.

```
# Segment red/fire zones
hsv = cv2.cvtColor(normalized, cv2.COLOR_BGR2HSV)
mask = cv2.inRange(hsv, (0, 50, 50), (10, 255, 255))
highlighted = cv2.bitwise_and(normalized, normalized, mask=mask)
```

This approach identifies likely fire-affected areas, which are then highlighted as overlays for rapid operator interpretation. These visual cues are displayed in a separate GUI window labeled "Detected Wildfire Areas".

3.4. Overlay Visualization and Interpretation

An example output from the tool is shown below, highlighting a non-burn region

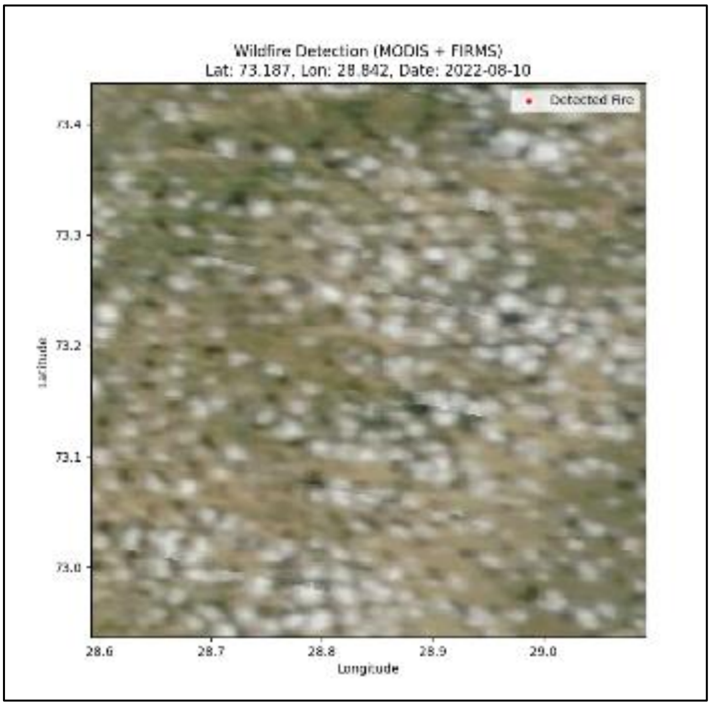

Figure 8 Satellite imagery of a high-latitude Arctic region on August 10, 2022

The image is centered at latitude 73.187° N and longitude 28.842° E, using MODIS Terra Corrected Reflectance from NASA GIBS. Although FIRMS fire detection data was queried for this region and date, no clear active fire signatures are visible within the displayed area. The surface texture suggests tundra or moss-covered terrain, interspersed with patchy cloud cover or snow. This high Arctic zone is typically not fire-prone, and any anomalies in thermal detection may arise from false positives or sensor artifacts rather than active combustion

This result highlights the importance of cross-verifying satellite-based thermal anomalies with both visual context and local climate characteristics. In subsequent cases, where visible hotspots were confirmed by FIRMS and supported by HSV-based detection overlays, the tool successfully marked fire-prone zones with high spatial precision. This dual-modality approach—combining model prediction and visual evidence—significantly improves reliability in operational wildfire surveillance systems.

4. Conclusion

In this study, we successfully developed a wildfire prediction and visualization system that integrates machine learning models with real-time satellite data to identify fire-prone regions with high accuracy. By combining Random Forest and Gradient Boosting algorithms and overlaying their predictions on satellite imagery, the system enhances both forecasting and situational awareness. This approach enables faster, data-driven decision-making during emergencies. The study will benefit society by supporting early detection and response to wildfires, ultimately reducing environmental damage and saving lives; future work may focus on incorporating temporal patterns and expanding coverage to global fire hotspots.

Compliance with ethical standards

Acknowledgments

The author extends sincere thanks to NASA's Earth data and FIRMS platforms for providing open access to environmental and satellite data. Special appreciation is given to the developers of skit-learn, OpenCV, and Python's Tainter library for enabling the implementation of this project.

The author independently conducted all research, analysis, and system development. Language editing and refinement assisted by GPT-based tools to enhance clarity and academic tone.

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] NASA FIRMS, "Fire Information for Resource Management System (FIRMS)," NASA Earthdata, 2024. [Online]. Available: https://firms.modaps.eosdis.nasa.gov

[2] NASA Earthdata, "Global Imagery Browse Services (GIBS)," NASA Earth Observing System Data and Information System (EOSDIS), 2024. [Online]. Available: https://earthdata.nasa.gov/gibs

[3] UCI Machine Learning Repository, "Forest Fires Data Set," Center for Machine Learning and Intelligent Systems, 1998. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/forest+fires

[4] NASA ARSET, "Introduction to NASA Earth Observations and Tools for Wildfire Monitoring and Management," NASA Applied Remote Sensing Training Program, 2023. [Online]. Available: https://appliedsciences.nasa.gov/join-mission/training/english/arset-introduction-nasa-earth-observations-and-tools-wildfire

[5] H. Moghim and A. Mehrabi, "Wildfire Risk Assessment Using Logistic Regression and Random Forest Models: A Case Study in the U.S. and Canada," International Journal of Wildland Fire, vol. 33, no. 2, pp. 115–127, 2024.

[6] Scikit-learn Developers, "sklearn.ensemble.RandomForestClassifier," Scikit-learn Documentation, 2023. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[7] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, 2001.

[8] OpenCV Team, "OpenCV: Open-Source Computer Vision Library," Version 4.9.0, 2024. [Online]. Available: https://opencv.org

[9] Python Software Foundation, "Tkinter GUI Programming," Python 3.11 Documentation, 2024. [Online]. Available: https://docs.python.org/3/library/tkinter.html

[10] S. Raschka, "Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning," arXiv preprint arXiv:1811.12808, 2018. [Online]. Available