IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Wildfire Prediction & Early Detection System

[1]Shubham Anand, [2]Surajmohan Thakur, [3]Swoaham Upadhyay, [4]Dr. Sunil Chavan

[1,2,3]B.E. Students, Department of Computer Engineering, Smt. Indira Gandhi College of Engineering, Navi Mumbai, Maharashtra, India.

[4]Principal, Smt. Indira Gandhi College of Engineering, Navi Mumbai, Maharashtra, India.

Abstract: Wildfires are the most deadly and dangerous accidents. Many lives are lost, and billions of worth of property damages occur in wildfire every year. Wildfires are fueled and accelerated by several different factors, such as weather, climate, vegetation types, land cover, and human activities. We propose a systematic way to make fire risk prediction and detection models that analyses satellite data, weather data, and historical fire data to predict fire. In this project we are implementing wildfire prediction and detection system. For prediction we are using four different types of dataset which are fire-history, weather, remote sensing and vegetation. The main aim of the project is to answer the possibility of fire on a given date and a given location. This answer will help firefighters and the public to take necessary precautions at any time. The fire prediction takes three different factors to account - the weather, the land cover, and the fire history of the place. We can obtain new insights by combining all these data and have divided the datasets into weather-based data and remote sensing-based data. Then, we applied the machine learning models to these different datasets, and finally, we used an ensemble model to predict the fire Prediction. For detection we are using Convolutional Neural Networks (CNN) from video or image which detect fire as well as smoke using camera. After the detection of fire or smoke it sends message to nearest fire station.

***Index Terms* - SVM, RF, CNN, Machine Learning, dataset, fire, Prediction, Detection.**

I. INTRODUCTION

Wildfires are one of the most dangerous and common natural disasters that we encounter every year. Uncontrolled wildfires not only cause damage to the environment and disrupt the ecological balance and cause significant loss to life and property[1]. If we look at statistics, over 3000 lives were lost, and more than $23 billion in property damage was reported in wildfires[2]. Wildfires are both difficult to predict and fight because each wildfire is unique to the place where it occurs. A combination of various factors such as dry vegetation, gusty wind, terrain, weather, etc., further aggravates the situation. With so much impact on social and economic factors, it becomes necessary to build a near real-time solution for predicting and controlling wildfires. A myriad of research papers has been published addressing wildfire detection and prediction by using mathematical and statistical methods. Still, these models have a lot of limitations such as limited parameters, low accuracy of risk prediction, the complexity of equations, and lack of real-time decision-making processes. According to the recent wildfire survey, most of the wildfire emergency systems still use conventional wildfire detection and prediction approaches[3][4][5]. In statistical and mathematical methods, we infer the relationship among the variables, while in the machine learning (ML) models, the focus is to make the most accurate predictions possible. Hence with the advancement in Machine Learning and Neural Networks, we can leverage the advanced algorithms to improve the lagging outcomes of wildfire risk prediction and detection systems. Many types of research are focused on investigating the probability of the burning, while others focus on the intensity and effects of the wildfires[6]. Moreover, earlier studies also show the deployment of a limited number of parameters with limited accuracy. Therefore, we aim to include different parameters such as fire history, weather, remote sensing, and satellite data to improve the accuracy of fire risk prediction and detection models.

We have employed Machine Learning and Deep Learning techniques to solve fire risk prediction and detection with cutting edge accuracy. We have applied many machine learning (ML) models such as Support Vector Machine (SVM), eXtreme Gradient Boosting (XGBoost), Random Forest (RF), Deep Learning techniques such as Convolutional Neural Networks (CNN), and YOLOv8. We have used an ensemble model for fire risk prediction, and for fire detection, we have used YOLOv8. The ML models for both the tasks have been validated using the ROC curves, Confusion matrices and accuracy.

1.1 Problem Statement

To develop a system that predicts wildfires using various Machine Learning based algorithm approach and detect this fire in the images or videos using Convolutional Neural Network.Therefore, the proposed solution is designed to:

1. Train a prediction model using Random forest algorithm.
2. Predict the possibility of wildfire of the given set of attributes.
3. Detect fire in the same map space confirming the presence of wildfire in the predicted area.

II. Data Analysis

2.1 Weather Data

We collected the 2 years of weather data to provide data analytics and insights into the relationship between weather and the possibility of wildfire in India. The weather data collected from the National Center for Environmental Information (NCEI) provides valuable information about station, date, latitude, longitude, temperature, dew point, wind speed, gust, and precipitation. The collected data was highly imbalanced as there were very few fire occurrences in a year compared to no-fire events. Hence, we used Synthetic Minority Oversampling Technique (SMOTE) to oversample and generate samples of both data classes . To deal with the missing values for better performance, we filled them with the mean values.

2.2 Remote Sensing Data

We used remote sensing images for the fire prediction. Landsat 8 images contain 11 bands ranging from Band 1 to Band 11 based on different wavelengths and resolutions. We used the rasterio library of Python to process and visualize the details of the images. In satellite imaging, each place on earth is referred to by a path and row, hence we filtered the images for India that were covered by two coordinates - (path=40, row=37) and (path=39, row=37). The cloud cover value in the dataset indicates the amount of cloudiness on that day. If the cloud coverage is more, the image will not have enough features for analysis. Therefore, we have filtered the dataset for cloud cover < 10. Study and visualize: The parameters tiled, the number of bands, size of the band image, and size of each tile were explored using the profile data of the raster image. Tiling is the process by which each scene is split into smaller windows. scene covers 230.73km x 234.63km on the Ground, which is significant for most of the studies. Hence, we created small windows covering the smaller area of 14km x 14km out of the entire scene.

Ex :- Fig. shows the tiles generated for one scene of India, and here, we can observe that the size of the one block in an image is 512 x 512 while the size of each image is 7691 x 7691.

$$\boxed{band\_toa = M_p * ds + A_p}$$

2.3 Fire History Data

We used the fire history data provided by Fire and Resource Assessment Program (FRAP) for 2 year. The geographic locations of the past fires were divided into 1 x 1 km grids. We denoted the presence of fire and no fire with the binary variables.

2.4 Vegatation Data

The vegetation data is used to determine the exact land cover of a place and helps to calculate the Normalized Difference Vegetation Index (NDVI values). We can predict the fire risk with greater accuracy if we know the exact type of vegetation. shows the vegetation layout of the India area in which we generated coloured labels using QGIS visualization. For example, we gave different shades of green to grasslands, scrub, and woodland. We labeled purple to our developed area of interest. NDVI is a graphical indicator to determine the live vegetation in the image. NDVI, as a value, shows the amount of chlorophyll content reflected from a place on earth. From previous studies, we found that NDVI values drop significantly during a wildfire [7]. We also combined red and nir bands to create NDVI images to be used for classification. The number of fire images in the resulting dataset was 1425. We Combined different years of data from the above data sources and stratified samples for model building formulation after statistically analyzing all the samples. Then, we split the dataset randomly to 80% and 20% as training and test datasets. Further, we divided the training datasets into 4:1 ratio as training and validation datasets.

III. Methodology

Uncontrolled wildfires not only cause damage to the environment and disrupt the ecological balance and cause significant loss to life and property. Wildfires are both difficult to predict and fight because each wildfire is unique to the place where it occurs. A combination of various factors such as dry vegetation, gusty wind, terrain, weather, etc., further aggravates the situation. With so much impact on social and economic factors, it becomes necessary to build a near real-time solution for predicting and controlling wildfires. We have employed Machine Learning and Deep Learning techniques to solve fire risk prediction and detection with cutting edge accuracy. We have applied many machine learning (ML) models such as Random Forest Regressor, Linear Regression, Ridge Regression, K-Nearest Neighbors Regressor, Lasso Regression, Support vector Regressor, Random Forest (RF), XGBoost Classifier, Decision Tree Classifier and Logistic Regression. In Deep Learning techniques we are using Convolutional Neural Networks (CNN) such as YOLOv8. We have used an ensemble model for fire risk prediction, and for fire detection, we have used YOLOv8, after detection Email will send to near Fire Station.

3.1 PROPOSED SYSTEM

Data Collection: For Fire Risk Predicting Collect data on various environmental factors such as temperature, humidity, wind speed, and precipitation, as well as vegetation cover, topography, and other relevant information about the forest. This data could be collected through weather stations, remote sensing, and ground-based sensors then convert into dataset. For Detection Collect data from video footage such as different types of websites as well as news that video include smoke and fire after that convert into Frame Image(dataset).

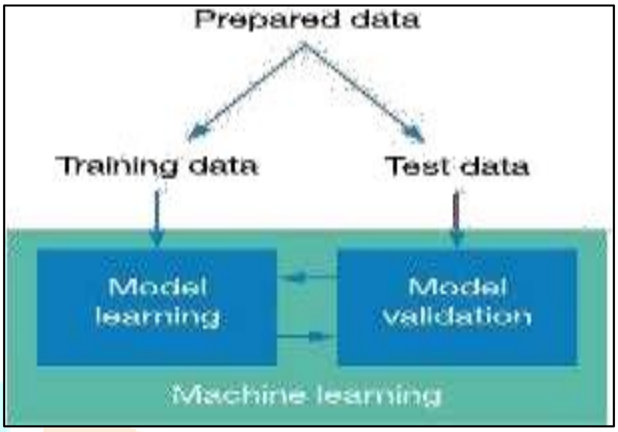


Figure: Collection of Data

Selection of attributes: Attribute or Feature selection includes the selection of appropriate attributes for the prediction system. This is used to increase the performance of the system. Various attributes of the Forest like Temperature noon, Relative Humidity, Wind Speed, Rain etc , are selected for the prediction. The Correlation matrix is used for attribute selection for this model. For detection Feature Selection is not required.

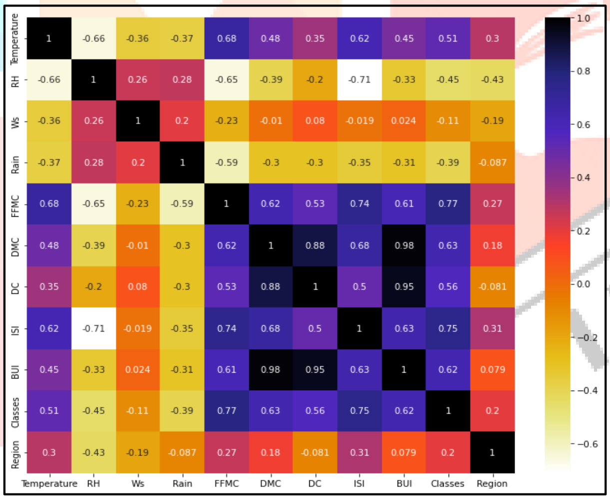


Figure: Correlation matrix

Pre-processing of Data: For Prediction we perform Exploratory Data Analysis (EDA) to extract insights from the dataset to know which features have contributed more in predicting Forest fire by performing Data Analysis using Pandas and Data visualization using Matplotlib & Seaborn. It is always a good practice told by data scientist to understand the data first and try to gather as many insights from it. Detection involves collecting, annotating, cleaning, augmenting, normalizing, splitting, and formatting the data to prepare it for model training or inference. This ensures that the input data is accurate, diverse, standardized, and properly formatted for the YOLOv8 model to learn to detect forest fires and smoke effectively.

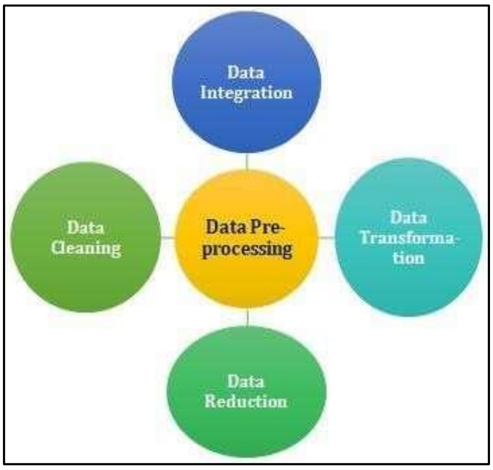


Figure: Data Pre-processing

Balancing of Data: For forest fire prediction, we have use Synthetic Minority Oversampling Technique (SMOTE) to address the class imbalance problem and improve the performance of our predictive model. In this case, we may have a dataset with a relatively small number of positive examples (i.e., forest fires) compared to negative examples (i.e., non-fire instances).

To use SMOTE for forest fire prediction, we can follow these steps:

1. Split the dataset into training and testing sets.
2. Train our machine learning model on the original imbalanced training dataset.
3. Apply SMOTE to the positive (i.e., fire) class in the training set to generate synthetic examples and balance the dataset.
4. Use the balanced training dataset to train our machine learning model.
5. Test the performance of our model on the original imbalanced testing dataset and evaluate its accuracy.

By using SMOTE to oversample the positive class, we have improve the accuracy of our model in detecting forest fires and reduce the chances of false negatives. However, it is important to choose the oversampling ratio carefully to avoid overfitting to the minority class.

For forest fire and smoke detection using YOLOv8, we have use SMOTE to balance the dataset if we have an imbalanced dataset with a small number of positive examples (i.e., instances with fire and smoke) compared to negative examples (i.e., instances without fire and smoke).

To use SMOTE for forest fire and smoke detection using YOLOv8, we can follow these steps:

1. Collect and preprocess the dataset of images or videos with and without fire and smoke instances.
2. Split the dataset into training and testing sets.
3. Train our YOLOv8 model on the original imbalanced training dataset.
4. Apply SMOTE to the positive (i.e., fire and smoke) class in the training set to generate synthetic examples and balance the dataset.
5. Use the balanced training dataset to fine-tune our YOLOv8 model.
6. Test the performance of our model on the original imbalanced testing dataset and evaluate its accuracy.

By using SMOTE to oversample the positive class, we can improve the accuracy of our YOLOv8 model in detecting forest fires and smoke and reduce the chances of false negatives. However, as mentioned earlier, it is important to choose the oversampling ratio carefully to avoid overfitting to the minority class.

ML Model Training: The selected features would be used to train an ML model such as a Decision Tree, Random Forest, or Neural Network. The model would be trained on historical data to predict forest fire outbreaks based on the environmental variables.

Object Detection: The pre-processed data would be fed into the YOLOv8 object detection model, which would detect and classify objects in the images or videos. The model would be trained on a large dataset of forest fire and smoke images to accurately detect these objects in real-time.

Alert Generation: When the model detects a forest fire or smoke in the images or videos, it would generate an alert to inform the authorities. The alert could be in the form of an email, text message, or a visual notification on a dashboard.

Continuous Model Improvement: The ML model as well as CNN would be continuously monitored and improved over time to ensure its accuracy and effectiveness in predicting forest fires as well as in detecting smoke and Fire. This could involve retraining the model with new data or modifying its architecture to improve its performance.

IV. Working of System

4.1 System Architecture

The system architecture gives an overview of the working of the system. Dataset collection is collecting data which contains forest fire details. Attributes selection process selects the useful attributes for the prediction of wildfire. After identifying the available data resources, they are further selected, cleaned, made into the desired form. Different classification techniques as stated will be applied on preprocessed data to predict the accuracy of wildfire. Accuracy measure compares the accuracy of different classifiers.

For detection forest fire and smoke detection involves the preparation of a dataset, training the YOLOv8 model on a training set, tuning its hyperparameters, and evaluating its performance on a test set. Segmentation is performed on the input data, allowing the YOLOv8 model to detect forest fires and smoke in each segment, which are then combined to generate a final output. An alert system is triggered based on the output, and mitigation strategies can be put in place to prevent the fire from spreading or to control the smoke. The system continuously monitors the input data for any changes or new occurrences of forest fires and smoke.

For Prediction:

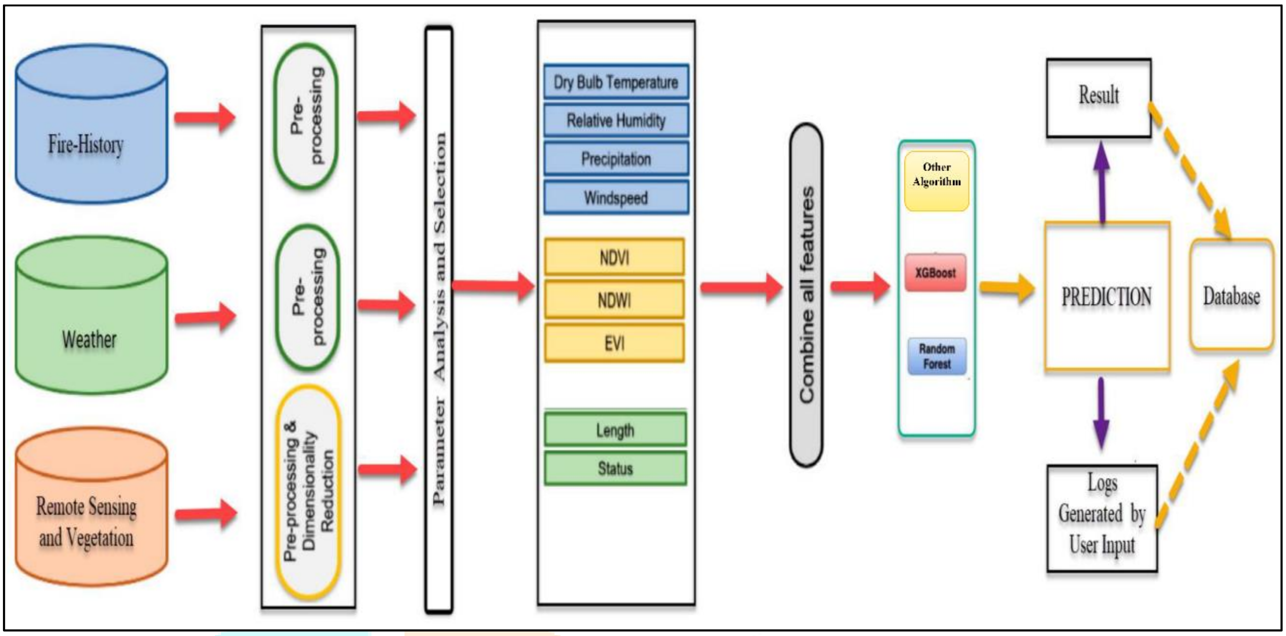


Figure: System Architecture for Prediction

For Detection:

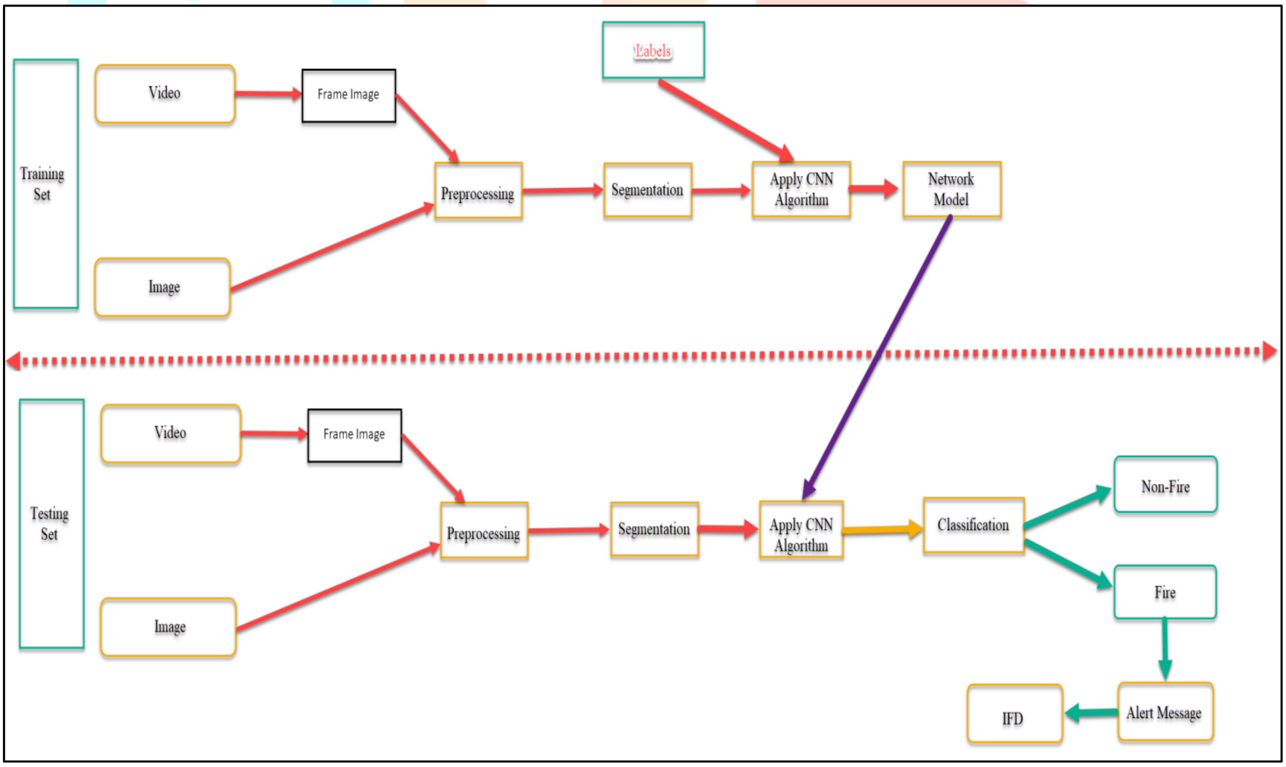


Figure: System Architecture for Detection

Flowchart of Prediction and Early Detection System

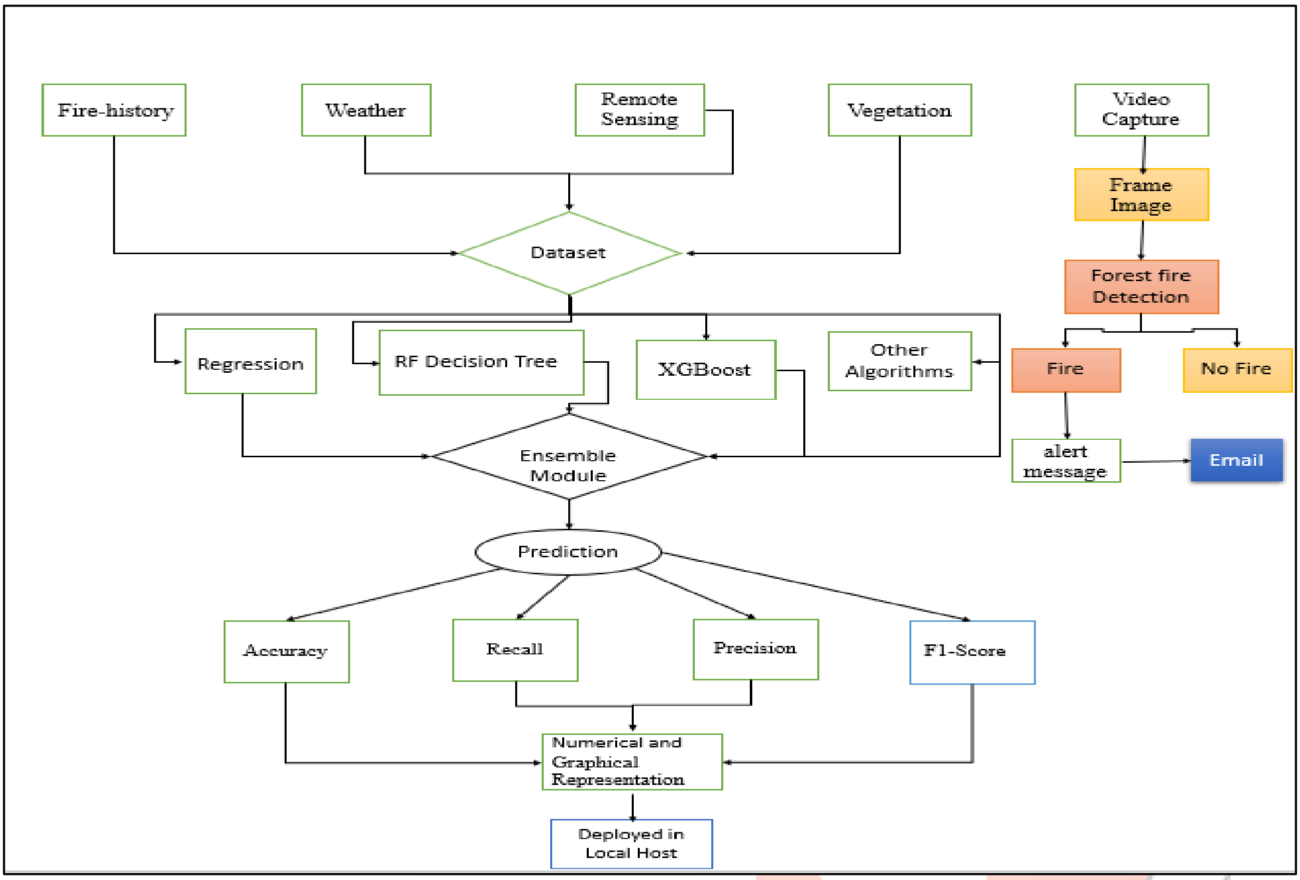


Figure: System Flowchart for Prediction and Detection

V. Algorithm Development

➢ **Prediction**

A. Regression

1. Linear Regression

Linear regression can be used for fire prediction by building a model that estimates the likelihood of a fire occurring based on various predictors, such as temperature, humidity, wind speed, and fuel moisture content.The first step is to collect data on past fires and the associated weather and environmental conditions. This data can be used to train a linear regression model that predicts the probability of a fire occurring given certain weather and environmental conditions.Once the model has been trained, it can be used to predict the likelihood of a fire occurring in the future based on current or forecasted weather and environmental conditions. For example, if the model indicates a high probability of a fire occurring based on the current temperature, wind speed, and fuel moisture content, fire officials can take preemptive measures such as issuing a fire warning, initiating fire prevention measures, or allocating firefighting resources to prevent or control the fire.It is important to note that linear regression is just one of many statistical methods that can be used for fire prediction, and its effectiveness will depend on the quality and quantity of the data used to train the model, as well as the assumptions made in the modeling process.

2. Lasso Regression

Lasso regression, a type of regularized linear regression, can be used for fire prediction by building a model that work same as that of linear regression. Lasso regression can be particularly useful for fire prediction because it includes a regularization term that can help to reduce overfitting and select the most important predictors. In this way, lasso regression can help to identify the key environmental factors that contribute to the likelihood of a fire, which can then be used to guide fire prevention and management strategies.It is important to note that lasso regression, like any statistical method, has its limitations and assumptions. Careful consideration must be given to the quality and quantity of the data used to train the model, as well as the selection of predictors and regularization parameter.

3. Ridge Regression

Ridge regression, another type of regularized linear regression, can also be used for fire prediction by building a model that work same as that of linear regression.Ridge regression can be particularly useful for fire prediction because it includes a regularization term that helps to reduce the impact of multicollinearity, which is when predictors are highly correlated with each other. In fire prediction, this could occur, for example, if temperature and humidity are highly correlated, making it difficult to discern the relative importance of each predictor. Ridge regression can help to mitigate this issue by shrinking the coefficients of highly correlated predictors towards each other.As with lasso regression, careful consideration must be given to the quality and quantity of the data used to train the model, as well as the selection of predictors and regularization parameter. Ridge regression, lasso regression, and linear regression each have their own strengths and weaknesses, and the choice of which method to use may depend on the specific data and modeling goals.

4. Support Vector Regressor

Support Vector Regression (SVR) is a type of regression analysis that uses support vector machines (SVMs) to build a predictive model. SVR can also be used for fire prediction.SVR models work by finding a hyperplane that maximizes the margin between the predicted values and the actual values, while still allowing for a certain amount of error. In fire prediction, this means finding the best hyperplane that separates the data into two classes: fire and no fire. It is important to note that SVR can be sensitive to the selection of hyperparameters, such as the kernel function, the regularization parameter, and the kernel coefficient. Therefore, careful consideration must be given to the selection of these parameters to ensure the best possible model performance. Additionally, as with any predictive model, the quality and quantity of the data used to train the model are critical to its accuracy and effectiveness.

5. Random Forest Regressor

Random Forest Regression is a type of ensemble learning method that can also be used for fire prediction.Random Forest Regression works by building a large number of decision trees on randomly sampled subsets of the data and predictors. Each decision tree in the forest independently predicts the probability of a fire occurring based on its specific subset of the data and predictors. The predictions from each tree are then combined to produce the final predicted probability of a fire occurring.Random Forest Regression has several advantages for fire prediction, including its ability to handle both linear and nonlinear relationships between predictors and the predicted variable, its ability to handle missing data and outliers, and its ability to identify the most important predictors for fire prediction.The number of decision trees and other hyperparameters of the model must be selected carefully to achieve the best possible performance.

6. K-Nearset Neighbors (KNN) Regressor

K-Nearest Neighbors (KNN) regression is a type of non-parametric regression that can be used for fire prediction. KNN regression works by calculating the distance between the new data point (i.e., the current or forecasted weather and environmental conditions) and the existing data points in the training set. The K closest data points to the new data point are then used to estimate the probability of a fire occurring, based on the average of their associated outcomes (i.e., whether a fire occurred or not). KNN regression has several advantages for fire prediction, including its ability to handle both linear and nonlinear relationships between predictors and the predicted variable, its ability to handle missing data and outliers, and its simplicity and interpretability.However, KNN regression also has some limitations, including its sensitivity to the choice of the distance metric used to calculate the distances between data points, and its computational complexity for large datasets. The value of K (i.e., the number of closest data points used for prediction) must be carefully chosen to achieve the best possible performance.

B. Classification

1. Logistic Regression

Logistic Regression is a type of regression analysis that can also be used for fire prediction by building a model that estimates the likelihood of a fire occurring based on various predictors, such as temperature, humidity, wind speed, and fuel moisture content. To use logistic regression for fire prediction, the first step is to collect data on past fires and the associated weather and environmental conditions. This data can be used to train a logistic regression model that predicts the probability of a fire occurring given certain weather and environmental conditions. Logistic regression works by estimating the probability of a binary outcome (i.e., whether a fire occurred or not) based on a linear combination of the predictors. The logistic regression model converts the linear combination into a probability using a logistic function, which maps the output to a value between 0 and 1. Once the logistic regression model has been trained, it can be used to predict the likelihood of a fire occurring in the future based on current or forecasted weather and environmental conditions. Logistic regression has several advantages for fire prediction, including its ability to handle both linear and nonlinear relationships between predictors and the predicted variable, its simplicity and interpretability, and its ability to provide probabilistic predictions. However, logistic regression also has some limitations, including its sensitivity to outliers and the presence of correlated predictors, and its assumption of a linear relationship between the predictors and the log-odds of the outcome. It is important to note that like any predictive model, the quality and quantity of the data used to train the model are critical to its accuracy and effectiveness. Additionally, feature engineering and selection may be required to improve the performance of the logistic regression model for fire prediction.

2. Decision Tree

Decision Trees are a type of machine learning algorithm that can be used for fire prediction. Decision trees work by recursively partitioning the data into subsets based on the value of the predictor variables, until each subset is as homogeneous as possible with respect to the predicted variable (i.e., whether a fire occurred or not). The resulting decision tree is a hierarchical structure that consists of decision nodes (representing the predictor variables and their values) and leaf nodes (representing the predicted outcome).Once the decision tree model has been trained, it can be used to predict the likelihood of a fire occurring in the future based on current or forecasted weather and environmental conditions, by traversing the decision tree from the root node to a leaf node that corresponds to a particular outcome (i.e., whether a fire will occur or not). Decision trees have several advantages for fire prediction, including their ability to handle both categorical and continuous predictor variables, their interpretability, and their ability to handle interactions between predictors. However, decision trees also have some limitations, including their tendency to overfit the training data, their sensitivity to the choice of the splitting criteria and stopping rules, and their lack of robustness to small changes in the data. Pruning and regularization techniques may be used to improve the performance of the decision tree model for fire prediction.

3. Random Forest

Random Forest is a type of ensemble machine learning algorithm that can be used for fire prediction. Random Forest works by building a large number of decision trees on random subsets of the data and predictor variables, and then combining the predictions of the individual trees to produce a final prediction. Each decision tree in the Random Forest is built by recursively partitioning the data into subsets based on the value of the predictor variables, until each subset is as homogeneous as possible with respect to the predicted variable (i.e., whether a fire occurred or not). The randomization in Random Forest comes from the fact that each decision tree is built on a random subset of the data and predictor variables, which helps to reduce overfitting and increase the robustness of the model to small changes in the data. Once the Random Forest model has been trained, it can be used to predict the likelihood of a fire occurring in the future based on current or forecasted weather and environmental conditions, by aggregating the predictions of the individual decision trees in the forest. Random Forest has several advantages for fire prediction, including its ability to handle both categorical and continuous predictor variables, its robustness to noisy and missing data, and its ability to capture complex nonlinear relationships between the predictor variables and the predicted variable. However, Random Forest also has some limitations, including its lack of interpretability (due to the large number of decision trees and the randomization), its tendency to overfit the data if the forest is too large, and its computational complexity. Tuning the hyperparameters of the Random Forest model (such as the number of trees in the forest and the maximum depth of each tree) may be necessary to achieve optimal performance for fire prediction.

4. K-Nearest Neighbors (KNN) Classifier

K-Nearest Neighbors (KNN) is a type of supervised machine learning algorithm that can be used for fire prediction. KNN works by finding the K nearest neighbors (i.e., data points with similar predictor variable values) to a new data point, and then classifying the new data point based on the majority class of its K nearest neighbors. For fire prediction, the KNN model would classify a new set of weather and environmental conditions as either "fire" or "no fire" based on the majority class of the K nearest neighbors to those conditions in the training data. The choice of K (the number of neighbors to consider) is an important hyperparameter in KNN and can have a significant impact on the performance of the model. A smaller value of K (e.g., K=1) can result in a model that is more sensitive to noise and outliers, while a larger value of K can result in a model that is more stable but less accurate. Once the KNN model has been trained, it can be used to predict the likelihood of a fire occurring in the future based on current or forecasted weather and environmental conditions, by finding the K nearest neighbors to the new data point and classifying it based on their majority class. KNN has several advantages for fire prediction, including its simplicity, speed, and ability to handle high-dimensional data. However, KNN also has some limitations, including its sensitivity to the choice of K and the distance metric used to measure similarity between data points, its inability to capture complex nonlinear relationships between the predictor variables and the predicted variable, and its susceptibility to overfitting if the training data is not representative of the population. Feature selection and normalization techniques may be used to improve the performance of the KNN model for fire prediction.

5. XGBoost Model

XGBoost (Extreme Gradient Boosting) is a popular machine learning algorithm that can be used for fire prediction by building a model that estimates the likelihood of a fire occurring based on various predictors, such as weather conditions, environmental variables, and historical fire data. To use XGBoost for fire prediction, the first step is to collect and preprocess the data. This can involve gathering data on past fires, weather conditions, and other relevant variables, and cleaning and formatting the data in a way that can be used by the XGBoost algorithm. Next, the data is split into training and testing sets. The training data is used to fit the XGBoost model to the data, while the testing data is used to evaluate the performance of the model. The XGBoost model works by fitting an ensemble of decision trees, where each tree is trained to predict the likelihood of a fire occurring based on the predictor variables. The trees are built in a way that minimizes the loss function of the model, which measures the difference between the predicted values and the actual values. During training, the XGBoost algorithm iteratively adds new trees to the ensemble, with each tree trained to correct the errors of the previous trees. This results in a highly accurate and robust model that can capture complex nonlinear relationships between the predictor variables and the predicted variable. Once the XGBoost model has been trained, it can be used to predict the likelihood of a fire occurring in the future based on current or forecasted weather and environmental conditions. The model takes as input the predictor variables and outputs a probability score between 0 and 1, indicating the likelihood of a fire occurring. XGBoost has several advantages for fire prediction, including its ability to handle high-dimensional data, its ability to capture complex nonlinear relationships between the predictor variables and the predicted variable, and its ability to automatically handle missing data. However, XGBoost also has some limitations, including its sensitivity to the choice of hyperparameters and the potential for overfitting if the model is not regularized properly. Overall, XGBoost is a powerful and flexible machine learning algorithm that can be used for fire prediction, and its performance can be improved by careful selection of hyperparameters and regularization techniques.

➢ Detection

A. You Look Only Once v8 (YOLOv8)

YOLO (You Only Look Once) is a state-of-the-art object detection algorithm that can be used for detecting fires and smoke in images and videos. YOLOv8 is a version of YOLO that has been trained on a large dataset of fire and smoke images to improve its detection accuracy for these specific classes. To use YOLOv8 for fire and smoke detection, the first step is to obtain and preprocess the image or video data. This may involve resizing the images or frames, converting them to the appropriate format, and removing any irrelevant information. Next, the YOLOv8 model is applied to the preprocessed data to detect any instances of fires or smoke. The YOLOv8 algorithm works by dividing the input image or video into a grid of cells, and for each cell, predicting the likelihood of different objects being present, as well as their location and size. During training, YOLOv8 is optimized to minimize a combination of classification loss (to predict the class of each object, i.e., fire or smoke) and localization loss (to accurately predict the position and size of each object). This results in a highly accurate and efficient object detection model that can detect fires and smoke in real-time. Once the YOLOv8 model has been applied to the input data, the resulting output can be further processed to extract useful information about the detected fires or smoke, such as their size, location, and intensity. This information can then be used to take appropriate action, such as alerting emergency services or triggering an alarm. Overall, YOLOv8 is a powerful and versatile object detection algorithm that can be used for a wide range of applications, including fire and smoke detection. Its ability to accurately detect objects in real-time makes it a useful tool for early warning and detection systems, and its performance can be further improved by fine-tuning the model on specific datasets or optimizing its hyperparameters.

VI. Dataset Details

Table-1 lists out the attributes used for prediction of wildfire. Meteorological factors include the factors that are concerned with the processes and phenomenon of the atmosphere like relative humidity, rain, temperature. Topographical factors include the factors related to the arrangement or accurate representation of physical distribution of features of an area like land surface temperature (LST), burnt area. Vegetation factors like Normalized Difference Vegetation Index (NDVI) that can be used to analyse remote sensing measurements that can be used for assessing whether the target being observed contains live green vegetation. The components of Fire Weather Index (FWI) are meteorologically based components used worldwide to estimate fire danger. It consists of different components that account for the effects of fuel moisture and wind on fire behavior and spread. Calculation of these components is based on consecutive daily observations of temperature, relative humidity, wind speed, and 24-hour precipitation. They include Fine Fuel Moisture Code (FFMC), Duff Moisture Code (DMC), Drought Code (DC), and Initial Spread Index (ISI).

Table -1: List of attributes used for Prediction

Attributes	Explanation
FFMC	The Fine Fuel Moisture Code is an indicator of the relative ease of ignition and the flammability of fine fuel.
DMC	The Duff Moisture Code is an indication of fuel consumption in moderate duff layers and medium size woody material.
DC	The Drought Code is an indicator of seasonal drought effects on forest fuels and the amount of smoldering in deep duff layers and large logs.
ISI	The Initial Spread Index combines the effects of wind and the FFMC on rate of spread without the influence of variable quantities of fuel.
Temperature	It is the temperature in degree Celcius
Rh	Relative humidity (RH) is the ratio of the amount of moisture in the air to the amount of moisture necessary to saturate the air at the same temperature and pressure
Wind	Wind increases the supply of oxygen, which results in the fire burning more rapidly. It also removes the surface fuel moisture, which increases the drying of the fuel
Rain	Rainfall in open measured in mm
Area	Area covered in hectares
NDVI	The normalized difference vegetation index is asimple graphical indicator that can be used to analyze remote sensing measurements, assessing whether or not the target being observed contains live green vegetation.
LST	Land Surface Temperature (LST) is the radiative skin temperature of the land derived from solar radiation.
Burnt Area	This index highlights burned land in the red to near-infrared spectrum, by emphasizing the charcoal signal in post-fire images.

6.1 Performance Analysis

For evaluating the models, various evaluation metrics like accuracy, confusion matrix, precision, recall, and f1-score are considered.

<u>For Prediction:</u>

1. Regression

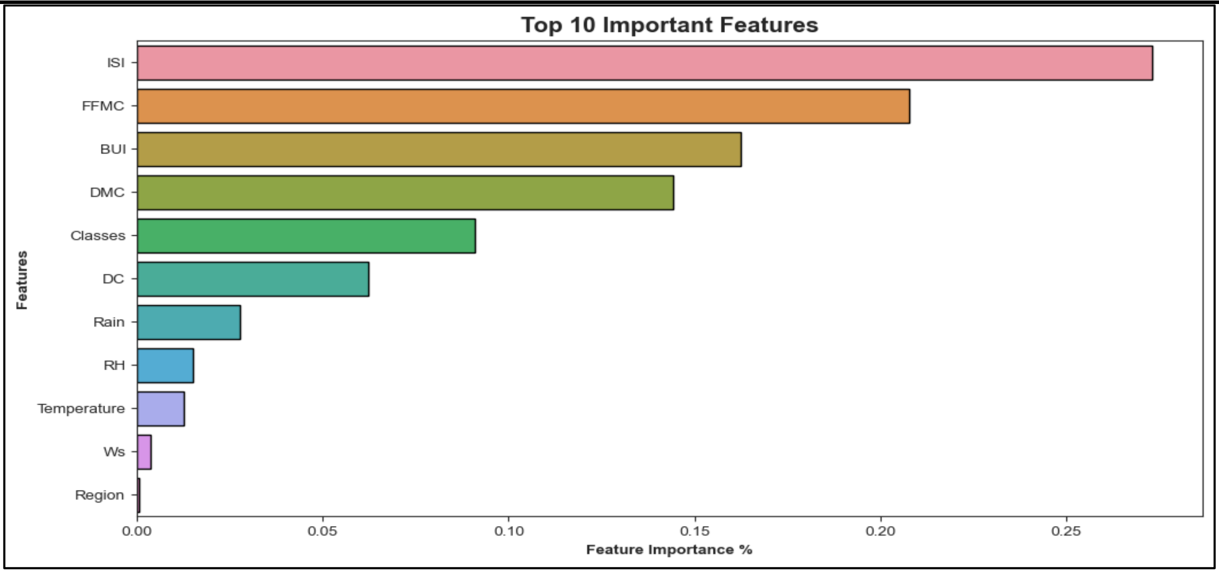
Accuracy score Results Summary

Models	Accuracy score
Random Forest Regressor	98.18 %
Linear Regression	97.00 %
Ridge Regression	96.90 %
K Neighbors Regressor	94.22 %
Lasso Regression	93.77 %
Support Vector Regressor	93.40 %

As evident from Accuracy score results summary, Random Forest regressor has performed best out of all models then we are doing Hyperparameter Tuning on Random Forest Regressor for increasing accuracy.

```
Random Forest Tuned
R2 Score value: 0.9695
MAE value: 0.6537
```

We will take 5 out of top 10 features. Then the accuracy of model increases to 98.18%.

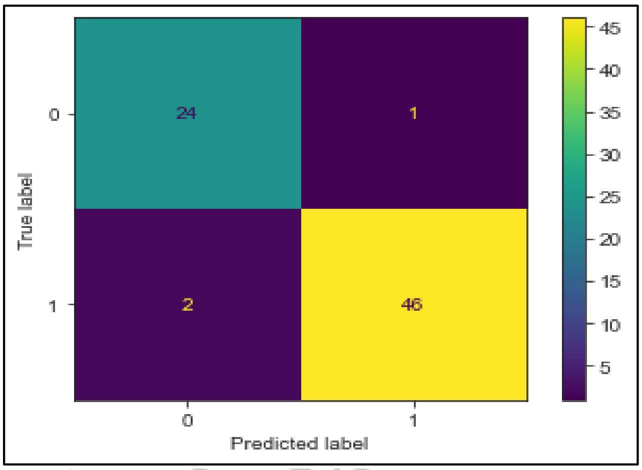


2. Classification

Logistic Regression

Logistic Regression
Accuracy Score value: 0.9589

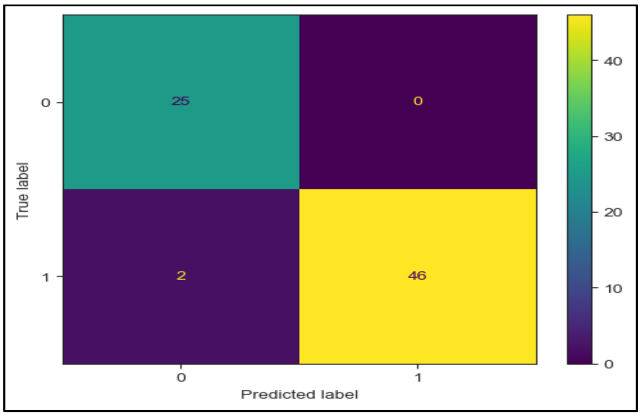
	precision	recall	f1-score	support
0	0.92	0.96	0.94	25
1	0.98	0.96	0.97	48
accuracy			0.96	73
macro avg	0.95	0.96	0.95	73
weighted avg	0.96	0.96	0.96	73



Decision Tree

Decision Tree
Accuracy Score value: 0.9726

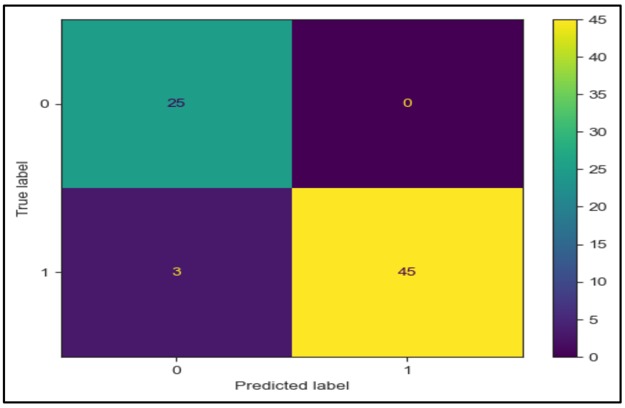
	precision	recall	f1-score	support
0	0.93	1.00	0.96	25
1	1.00	0.96	0.98	48
accuracy			0.97	73
macro avg	0.96	0.98	0.97	73
weighted avg	0.97	0.97	0.97	73



Random Forest

Random Forest
Accuracy Score value: 0.9589

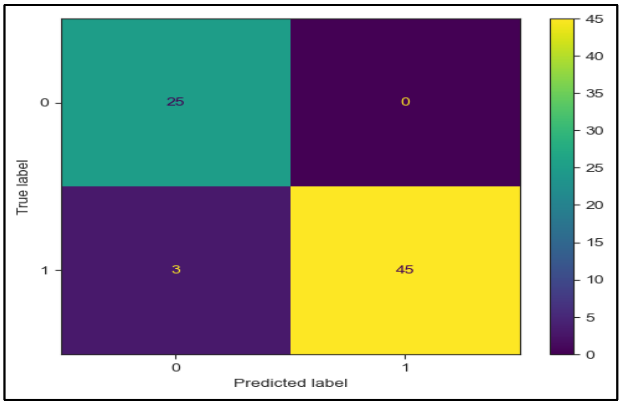
	precision	recall	f1-score	support
0	0.89	1.00	0.94	25
1	1.00	0.94	0.97	48
accuracy			0.96	73
macro avg	0.95	0.97	0.96	73
weighted avg	0.96	0.96	0.96	73



K-NEAREST NEIGHBORS CLASSIFIER

KNeighbors Classifier
Accuracy Score value: 0.9589

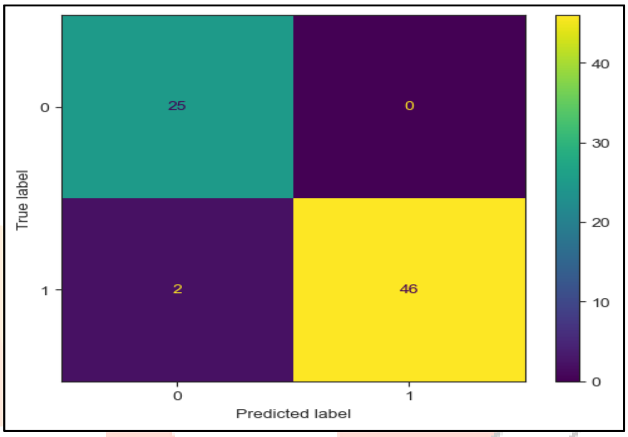
	precision	recall	f1-score	support
0	0.89	1.00	0.94	25
1	1.00	0.94	0.97	48
accuracy			0.96	73
macro avg	0.95	0.97	0.96	73
weighted avg	0.96	0.96	0.96	73



XGBOOST MODEL

XGboost Classifier
Accuracy Score value: 0.9726

	precision	recall	f1-score	support
0	0.93	1.00	0.96	25
1	1.00	0.96	0.98	48
accuracy			0.97	73
macro avg	0.96	0.98	0.97	73
weighted avg	0.97	0.97	0.97	73



Accuracy score Results Summary

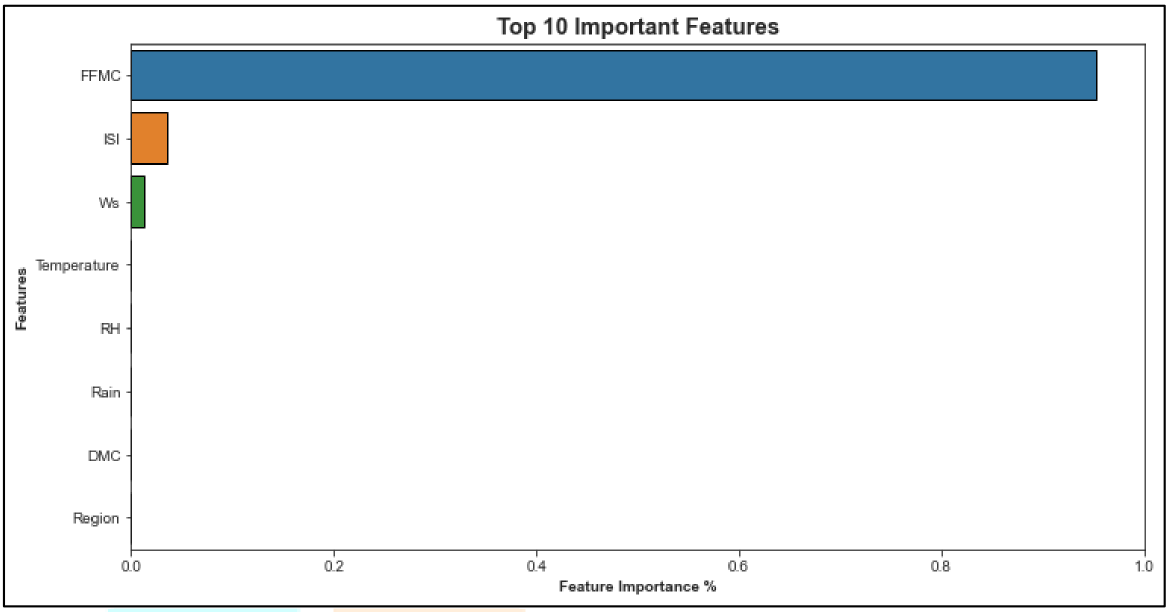
Models	Accuracy score
XGboost classifier	97.26 %
Decision Tree Classifier	97.26 %
Logistic Regression Accurracy	95.89 %
KNeighbors Classifier	95.89 %
Random Forest Classifier	95.89 %

As evident from Accuracy score result summary, XGboost Classifier has performed best out of all models then we are doing Hyperparameter Tuning on XGboost Classifier to increase its accuracy.

Final Model XGB
Accuracy Score value: 0.9726

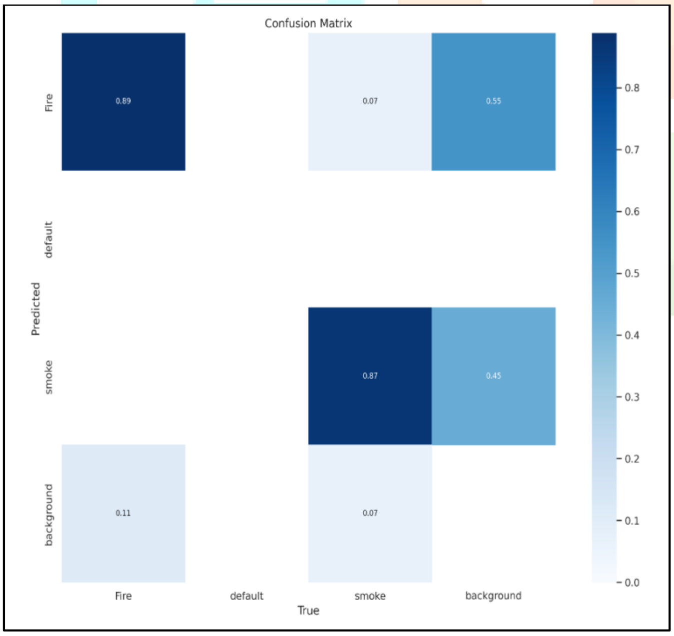
	precision	recall	f1-score	support
0	0.89	1.00	0.94	25
1	1.00	0.94	0.97	48
accuracy			0.96	73
macro avg	0.95	0.97	0.96	73
weighted avg	0.96	0.96	0.96	73

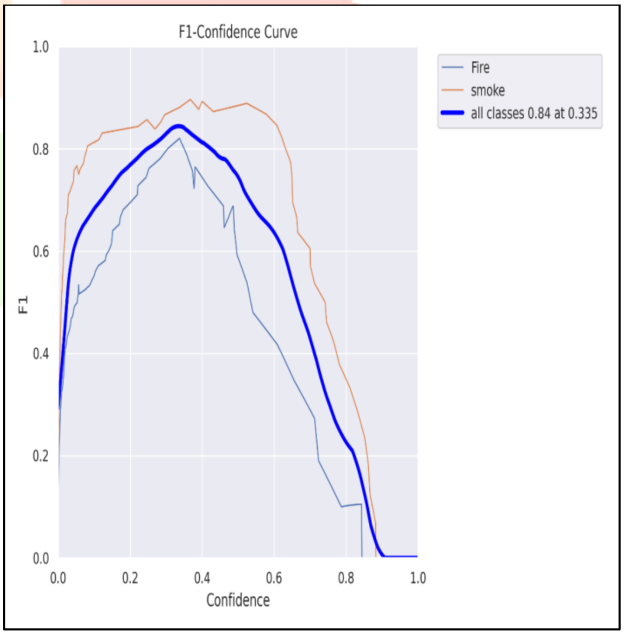
We will take 5 out of 10 top features.Then the accuracy of model increases to 97.26%.

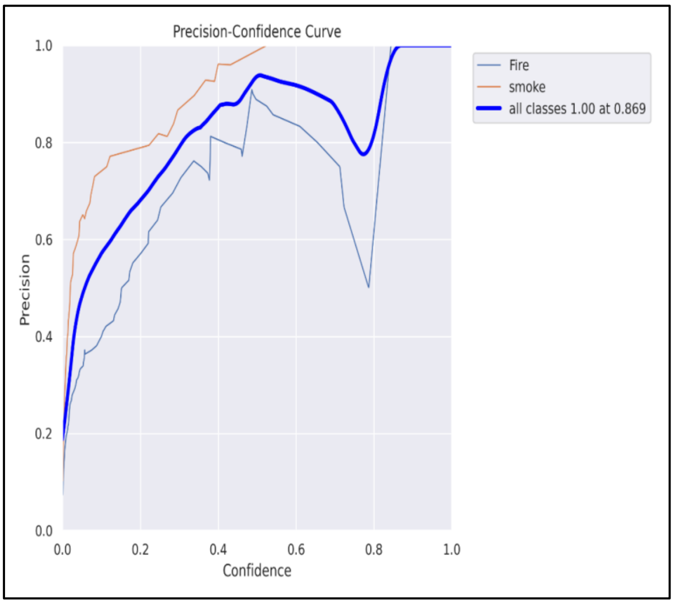


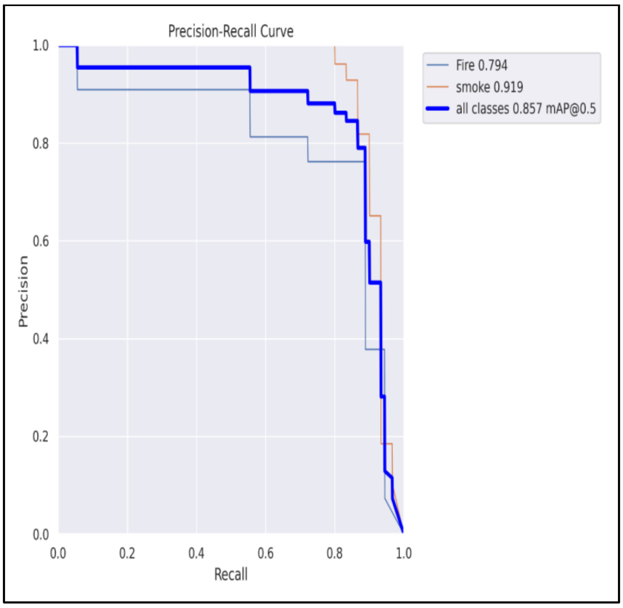
For Detection:

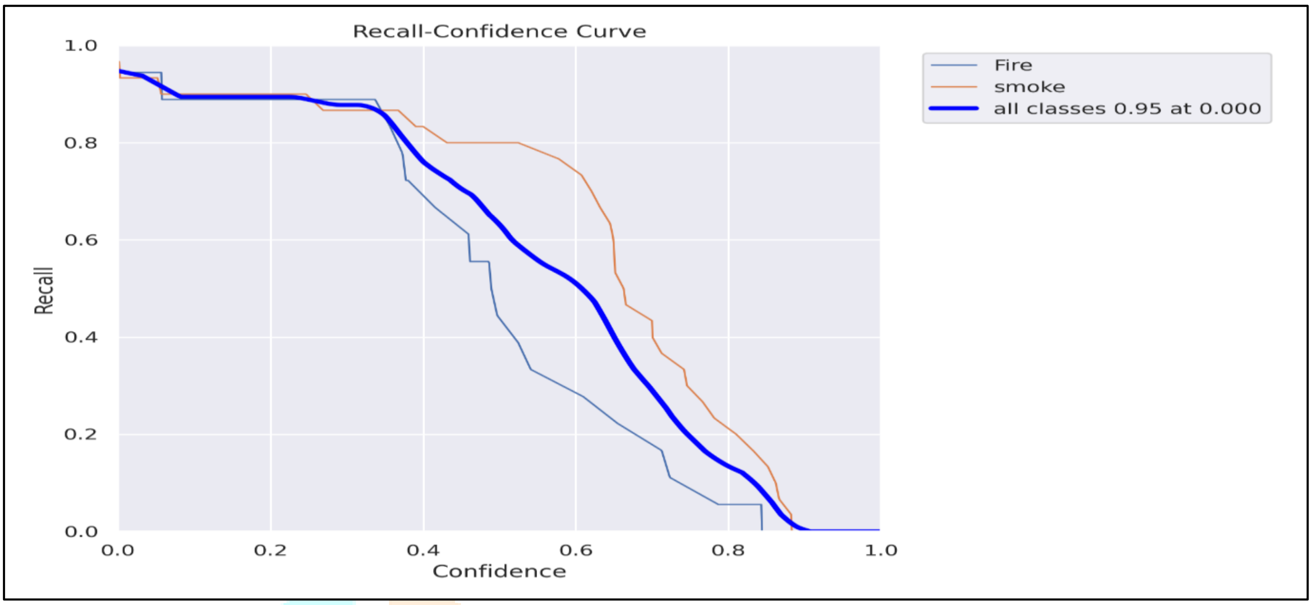
Performance analysis of the YOLOv8 model involves evaluating its accuracy, precision, recall, F1 score, and other metrics on a test set of data.











6.2 Output Data Collection and Processing

Prediction

```python
# Create Dataframe and Read the dataset using Pandas
dataset = pd.read_csv('Algerian_forest_fires_dataset_UPDATE.csv', header=1)
dataset.head()
```

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes
0	01	06	2012	29	57	18	0	65.7	3.4	7.6	1.3	3.4	0.5	not fire
1	02	06	2012	29	61	13	1.3	64.4	4.1	7.6	1	3.9	0.4	not fire
2	03	06	2012	26	82	22	13.1	47.1	2.5	7.1	0.3	2.7	0.1	not fire
3	04	06	2012	25	89	13	2.5	28.6	1.3	6.9	0	1.7	0	not fire
4	05	06	2012	27	77	16	0	64.8	3	14.2	1.2	3.9	0.5	not fire

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 243 entries, 0 to 242
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   day          243 non-null    int32  
 1   month        243 non-null    int32  
 2   year         243 non-null    int32  
 3   Temperature  243 non-null    int32  
 4   RH           243 non-null    int32  
 5   Ws           243 non-null    int32  
 6   Rain         243 non-null    float64
 7   FFMC         243 non-null    float64
 8   DMC          243 non-null    float64
 9   DC           243 non-null    float64
 10  ISI          243 non-null    float64
 11  BUI          243 non-null    float64
 12  FWI          243 non-null    float64
 13  Classes      243 non-null    object 
 14  Region       243 non-null    int32  
dtypes: float64(7), int32(7), object(1)
memory usage: 22.0+ KB
```

Detection

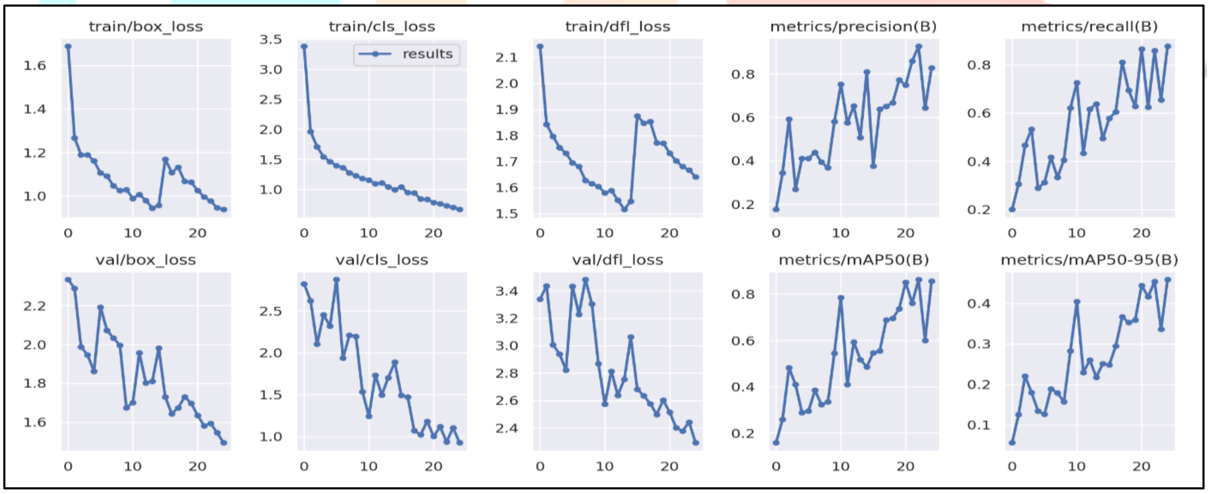
```
!ls {HOME}/runs/detect/train/

args.yaml                                              train_batch1.jpg
confusion_matrix.png                                   train_batch2.jpg
events.out.tfevents.1675348815.43529bab29be.37247.0    train_batch825.jpg
F1_curve.png                                           train_batch826.jpg
P_curve.png                                            train_batch827.jpg
PR_curve.png                                           val_batch0_labels.jpg
R_curve.png                                            val_batch0_pred.jpg
results.csv                                            val_batch1_labels.jpg
results.png                                            val_batch1_pred.jpg
train_batch0.jpg                                       weights
```

6.3 Statistical Measures Of Data

Prediction

	count	mean	std	min	25%	50%	75%	max
day	243.0	15.761317	8.842552	1.0	8.00	16.0	23.00	31.0
month	243.0	7.502058	1.114793	6.0	7.00	8.0	8.00	9.0
year	243.0	2012.000000	0.000000	2012.0	2012.00	2012.0	2012.00	2012.0
Temperature	243.0	32.152263	3.628039	22.0	30.00	32.0	35.00	42.0
RH	243.0	62.041152	14.828160	21.0	52.50	63.0	73.50	90.0
Ws	243.0	15.493827	2.811385	6.0	14.00	15.0	17.00	29.0
Rain	243.0	0.762963	2.003207	0.0	0.00	0.0	0.50	16.8
FFMC	243.0	77.842387	14.349641	28.6	71.85	83.3	88.30	96.0
DMC	243.0	14.680658	12.393040	0.7	5.80	11.3	20.80	65.9
DC	243.0	49.430864	47.665606	6.9	12.35	33.1	69.10	220.4
ISI	243.0	4.742387	4.154234	0.0	1.40	3.5	7.25	19.0
BUI	243.0	16.690535	14.228421	1.1	6.00	12.4	22.65	68.0
FWI	243.0	7.035391	7.440568	0.0	0.70	4.2	11.45	31.1
Region	243.0	1.497942	0.501028	1.0	1.00	1.0	2.00	2.0

Detection



VII. RESULTS

```
final_df
```

	Temperature	Ws	FFMC	DMC	ISI	FWI	Classes
0	29	18	65.7	3.4	1.3	0.5	0
1	29	13	64.4	4.1	1.0	0.4	0
2	26	22	47.1	2.5	0.3	0.1	0
3	25	13	28.6	1.3	0.0	0.0	0
4	27	16	64.8	3.0	1.2	0.5	0
...	...	...	...	...	...	...	...
238	30	14	85.4	16.0	4.5	6.5	1
239	28	15	41.1	6.5	0.1	0.0	0
240	27	29	45.9	3.5	0.4	0.2	0
241	24	18	79.7	4.3	1.7	0.7	0
242	24	15	67.3	3.8	1.2	0.5	0

243 rows × 7 columns

Prediction

- Regression

Accuracy score Results Summary

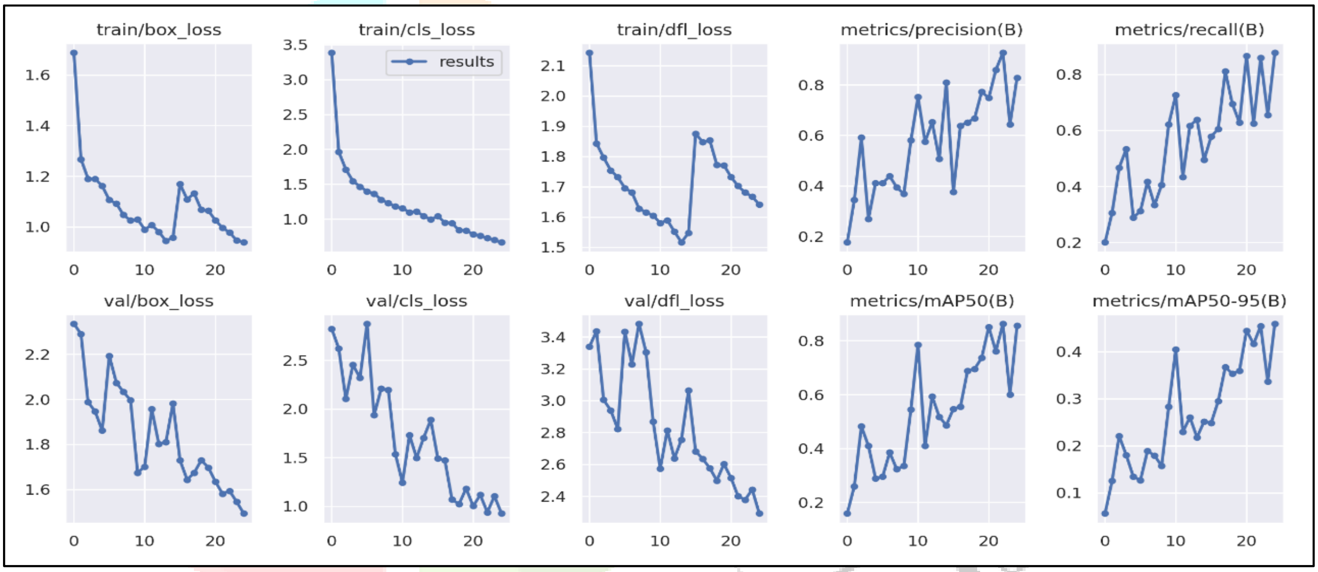
Models	Accuracy score
Random Forest Regressor	98.18 %
Linear Regression	97.00 %
Ridge Regression	96.90 %
K Neighbors Regressor	94.22 %
Lasso Regression	93.77 %
Support Vector Regressor	93.40 %

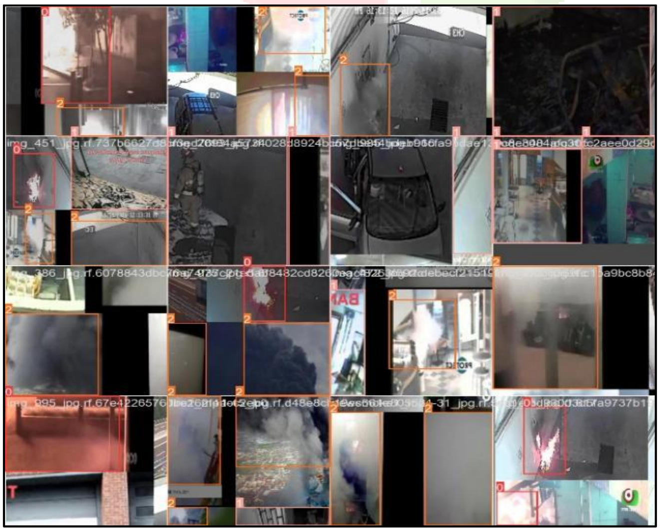
- Classification

Accuracy score Results Summary

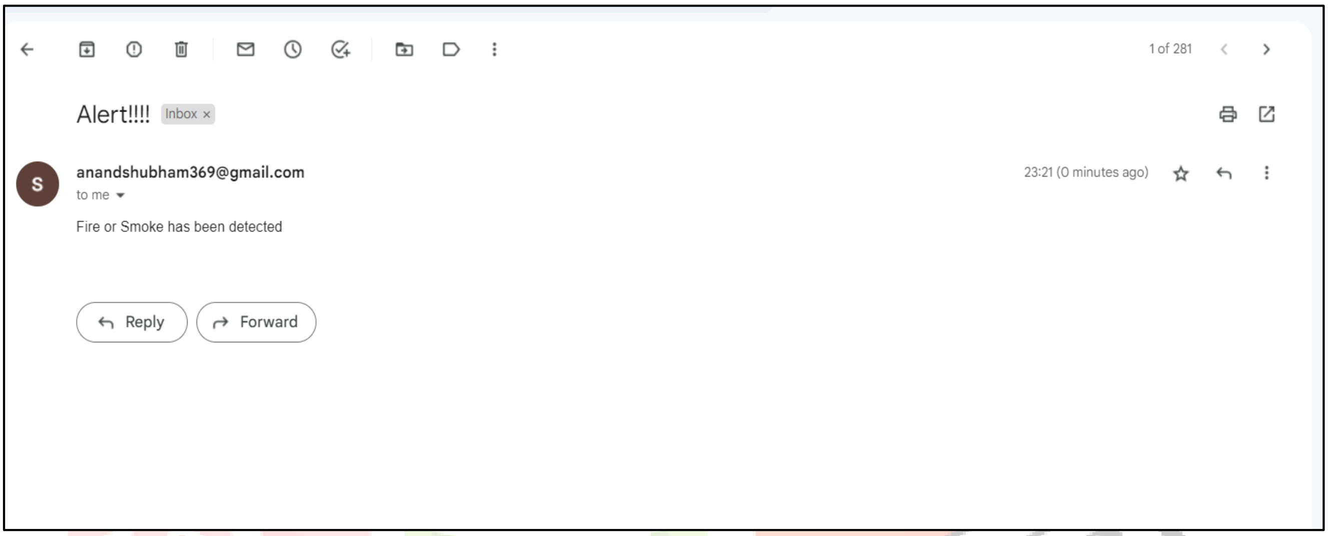
Models	Accuracy score
XGboost classifier	97.26 %
Decision Tree Classifier	97.26 %
Logistic Regression Accuracy	95.89 %
KNeighbors Classifier	95.89 %
Random Forest Classifier	95.89 %

Detection

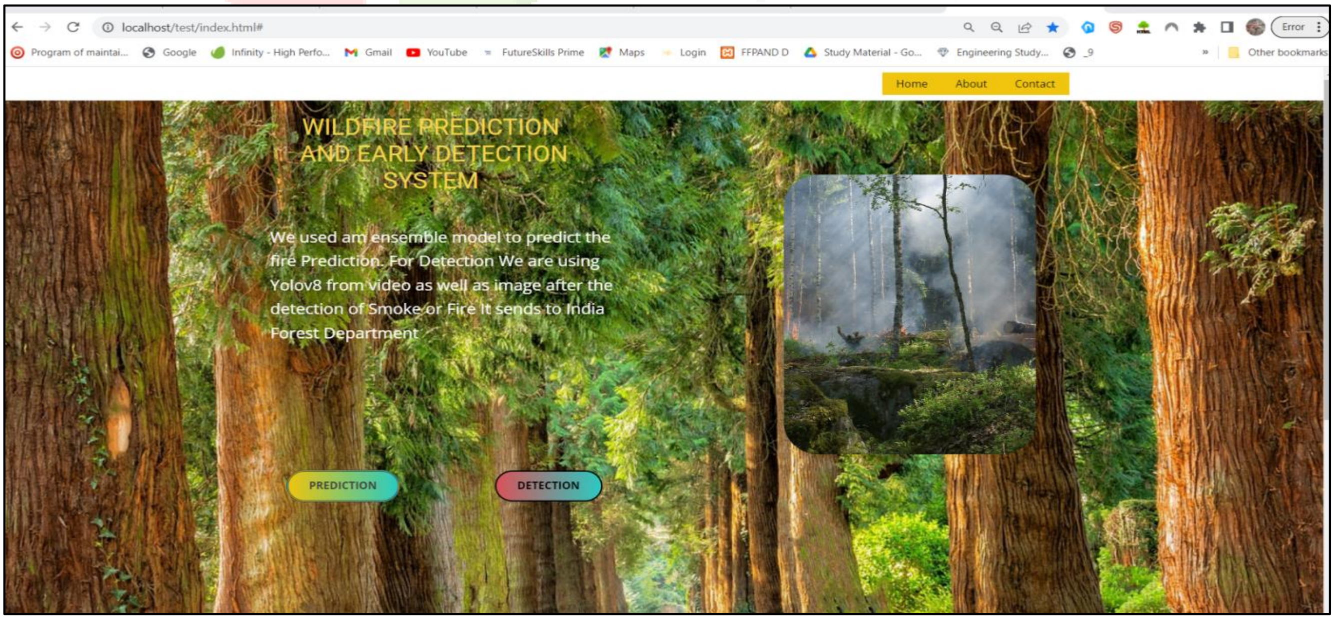


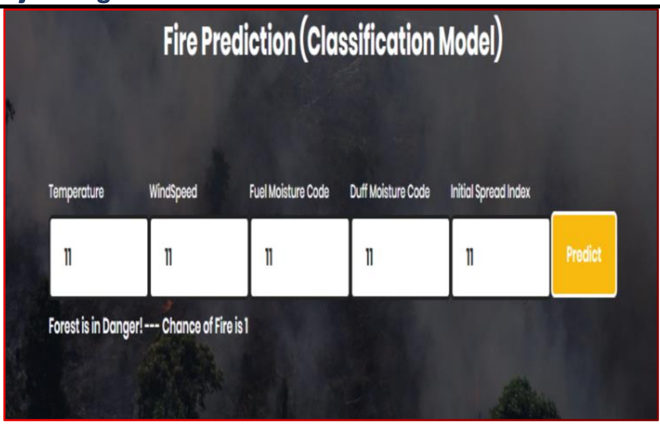


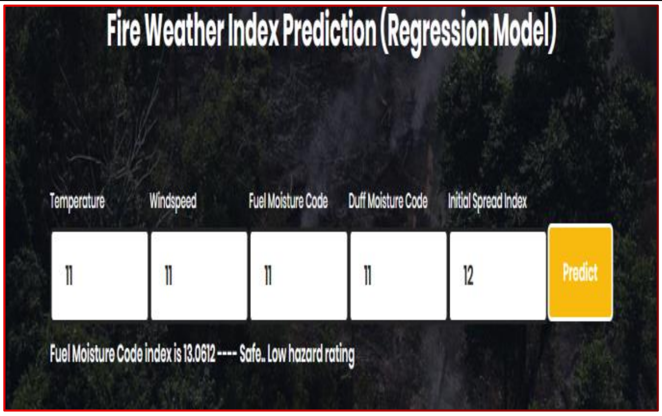
Alert message:



Graphical User Interface:







VIII. Conclusion and future work

Through this project we developed a tool to help mitigate and prevent the losses seen due to forest fires. Innovative solution which performs the prediction based on the prevalent conditions and then detect the forest fire. Through this project we learned to build a model using various Machine Learning Algorithm and Deep Learning. In our research, we have explored the effect of various types of data to study fire risk prediction and detection using machine learning approaches. Unlike other research that examines either fire risk detection or fire risk prediction with limited data and parameters, our work focuses on understanding these concepts using past fire events, weather, remote sensing, and satellite data. In the future, we plan to develop a real-time intelligent fire system that can provide information about the fire risk prediction, detection, and fire spread pattern.This project carries a broad prospective for future. Moreover it is a need for great research to be done in this field in the coming years. In future, our project can be extended towards finding an efficient way of localization of the fire, gravity of fire, direction of spread, area burnt and many more. Moreover, we can include the region specific meteorological data in the dataset for generating model for prediction.

IX. Reference

[1] W. J. Bond and J. E. Keeley, "Fire as a global 'herbivore': the ecology and evolution of flammable ecosystems," Trends Ecol. Evol., vol. 20, pp. 387-394, July 2005.
[2] M. E. Alsahaft, M. A. Alharthi, and M. Arif, "Role of Machine Learning Algorithms in Forest Fire Management: A Literature Review," J. Robot. Autom., vol. 5, pp. 212-226, February 2018.
[3] B. J. Stocks, B.D. Lawson, M.E. Alexander, C.E. Van Wagner, R.S. McAlpine, T.J. Lynham, and D.E. Dube, "The Canadian Forest Fire Danger Rating System: An Overview," For. Chron., vol. 65, pp. 450-457, December 1989.
[4] J. Škvarenina, J. Mindáš, J. Holécy, and J. Tuček, "An analysis of the meteorological conditions during two largest forest fire events in the Slovak Paradise National Park," Meteorl. Jour., vol. 7, pp. 167-171, January 2004.
[5] C. Andrés, C. Ana, and M. Tomàs, "Response time assessment in forest fire spread simulation: An integrated methodology for efficient exploitation of available prediction time," Environ. Model Softw., vol. 54, pp. 153-164, April 2014.
[6] Tiziano, A. Bachisio, P. Grazia, and D. Pierpaolo, "An Improved Cellular Automata for Wildfire Spread," Procedia Comput. Sci., vol. 51, pp. 2287-2296, June 2015.
[7] Radke, A. Hessler, and D. Ellsworth, "FireCast: Leveraging Deep Learning to Predict Wildfire Spread," Twenty-Eighth International Joint Conference on Artificial Intelligence, August 2019.