A Project Report on

Wildfire Risk Assessment and Detection Using AI and Satellite Data with Real-Time Alert System

For

Major Project (CUML1025)

by

Ankit Parida(220301120340)

Under the Supervision of

Mrs. Roseleen Anjum

SCHOOL OF ENGINEERING AND TECHNOLOGY
BHUBANESWAR CAMPUS
CENTURION UNIVERSITY OF TECHNOLOGY AND MANAGEMENT
ODISHA

APRIL-2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF ENGINEERING AND TECHNOLOGY
BHUBANESWAR CAMPUS

BONAFIDE CERTIFICATE

It is to certify that this project report ”Wildfire Risk Assessment and Detection Using AI and Satellite Data with Real-Time Alert System” is the bonafide work of ”Ankit Parida” who carried out the project work under my supervision. This is to certify that this project has not been carried out earlier in this institute and the university to the best of my knowledge.

(Mrs. Roseleen Anjum)
Asst prof. Dept.of CSE,SOET

Certified that the project mentioned above has been duly carried out as per the college's norms and the university's statutes.

Dr. Sujata Chakravarty
Dean, SoET

Mr. Rajkumar Mohanta
HoD, Dept. of CSE, SoET

DECLARATION

We hereby declare that the project entitled "Wildfire Risk Assessment and Detection Using AI and Satellite Data with Real-Time Alert System" submitted for the "Major Project" of 6^{th} semester B. Tech in Computer Science and Engineering our original work and the project has not formed the basis for the award of any Degree / Diploma or any other similar titles in any other University / Institute.

Ankit Parida(220301120340)

Place:Bhubaneswar

Date:

ACKNOWLEDGEMENTS

We wish to express our profound and sincere gratitude to Dr. Sujata Chakravarty , Department of Computer Science / Engineering, SoET, Bhubaneswar Campus, who guided me into the intricacies of this project nonchalantly with matchless magnanimity.
We thank Prof. Raj Kumar Mohanta, Head of the Department of Computer Science and Engineering, SoET, Bhubaneswar Campus, and Prof. Sujata Chakrabarty, Dean, School of Engineering and Technology, Bhubaneswar Campus, for extending their support during the course of this investigation.
We would be failing in our duty if we didn't acknowledge the cooperation rendered during various stages of image interpretation by Mrs. Roseleen Anjum .
We are highly grateful to Mrs. Roseleen Anjum who evinced keen interest and invaluable support in the progress and successful completion of our project work.
We are indebted to Mrs. Roseleen Anjum for their constant encouragement, cooperation, and help. Words of gratitude are not enough to describe the accommodation and fortitude that they have shown throughout our endeavor.

Ankit Parida (220301120340)

Place:Bhubaneswar

Date:

Abstract

Wildfires pose a significant threat to ecosystems, human life, and property, with their frequency and intensity increasing due to climate change and other environmental factors. This report presents the development of an AI-based system designed to enhance wildfire management through early prediction, real-time detection, spread analysis, and timely alerting. Leveraging machine learning algorithms, remote sensing data, satellite imagery, and environmental parameters such as temperature, humidity, and wind speed, the system predicts wildfire risks with high accuracy. Advanced deep learning models are employed for the rapid detection of active fires from imagery, while dynamic spread analysis models estimate the potential trajectory and growth of a fire. A responsive alerting mechanism ensures that authorities and vulnerable populations are promptly informed, enabling swift evacuation and resource deployment. The proposed system aims to minimize wildfire damage by providing a comprehensive, automated solution that integrates prediction, monitoring, and emergency response support.

Contents

List of Figures

List of Tables

1 Introduction

Wildfires are among the most devastating natural disasters, causing extensive environmental, economic, and social damage each year. Driven by factors such as rising global temperatures, prolonged droughts, and human activities, the frequency and intensity of wildfires have significantly increased in recent decades. Traditional methods of wildfire monitoring and management, which often rely on manual observations and reactive measures, are no longer sufficient to address the growing scale and complexity of these events.

The integration of Artificial Intelligence (AI) into wildfire management offers a promising solution to this challenge. AI-based systems can process vast amounts of environmental data, recognize complex patterns, and make accurate predictions faster than conventional methods. By utilizing machine learning algorithms, satellite imagery, sensor networks, and weather data, AI can not only predict wildfire risks but also detect active fires in real time, analyze their potential spread, and trigger timely alerts to relevant authorities and communities.

This report presents the design and development of an AI-Based System for Wildfire Prediction, Detection, Spread Analysis, and Alerting. The system aims to enhance early warning capabilities, improve situational awareness during wildfire events, and support efficient emergency response operations. By combining prediction, detection, and communication technologies, the proposed system seeks to minimize the impact of wildfires and contribute to the protection of both natural ecosystems and human lives.

2 Literature Survey

Author	Dataset	Model	Result
Meriame Mohajane (2021)	Remote sensing images (Mediterranean area)	FR-MLP, FR-LR, FR-SVM	Demonstrated effectiveness of ML techniques for mapping wildfire-prone areas; qualitative performance only
Anggy Praditha Junfithrana	Satellite imagery	CNN	Achieved 96% accuracy; improved wildfire detection from satellite data
Remus Sibisianu	UAV-based video input	CNN	93% accuracy; enabled real-time early detection of forest fires using drones
Byoungjun Kim (2019)	Fire and non-fire video clips	CNN, LSTM	Accuracy: 97.32%; combined temporal and spatial features for fire detection in video feeds
Chi Yuan (2015)	UAV remote sensing	DDDAS Framework	Accuracy: 81%; adaptive detection using dynamic UAV data systems
Zohreh Asghar (2020)	Streaming satellite data	RPCA (Robust PCA)	Accuracy: 95.33%; used RPCA for effective fire detection in live satellite feeds
Zohreh Asghar (2020)	Satellite images	RPCA	89% accuracy; emphasized real-time wildfire monitoring using continuous remote data
Mukhriddin Mukhdidnov (2022)	UAV images (YOLO dataset)	YOLOv5, YOLOv3	Accuracy: 73.6%; model struggled with smoke detection in foggy or hazy environments
Turgay Celik (2009)	MODIS satellite images	Thresholding, Neural Networks	Precision: 90%, Recall: 88%; used contextual and spectral features for fire pixel detection
Almeida et al. (2019)	MODIS + Sentinel-2 data	CNN	Accuracy: 94.1%; demonstrated feasibility of deep learning with multi-source data for fire prediction

3 Proposed Method

The proposed system integrates advanced Artificial Intelligence (AI) techniques to provide a comprehensive solution for wildfire prediction, detection, spread analysis, and alerting. The system first collects real-time and historical data from various sources such as satellite imagery, weather stations, vegetation indices, topographical maps, ground sensors, and UAV (drone) imagery. This multisource data is preprocessed to enhance relevant features, such as fire hotspots and smoke patterns, ensuring that the subsequent models are trained on high-quality inputs.

For wildfire prediction and detection, two main AI approaches are employed. For wildfire prediction, several machine learning models are used, including Logistic Regression, K-Nearest Neighbors (KNN), Decision Trees (CART), Naïve Bayes, Support Vector Machines (SVM), Multi-layer Perceptron (MLP), Gradient Boosting, AdaBoost, Bagging, Random Forest, and Extra Trees. These models analyze environmental variables such as temperature, humidity, wind speed, and vegetation indices to assess the probability of fire occurrence.

For real-time fire detection, deep learning models based on Convolutional Neural Networks (CNNs) are employed. Specifically, the system utilizes a custom-built CNN and transfer learning techniques using pre-trained models such as VGG16. These models are trained on labeled fire image datasets to distinguish between fire and non-fire regions. The models are optimized to balance detection accuracy and computational efficiency, making them suitable for real-time or near-real-time image analysis tasks.

Following detection, the system performs spread analysis using both simulation-based methods (like FARSITE) and AI-based models such as Graph Neural Networks (GNNs). These models simulate the potential spread of a wildfire based on key factors like terrain elevation, fuel types, wind direction, and weather patterns. Such predictions are critical for planning evacuation routes, firefighting operations, and resource allocation.

Finally, the system features an automated alerting module that issues timely warnings through multiple communication channels, including SMS, mobile applications, emails, and public broadcast systems. Additionally, a Telegram bot is integrated to notify emergency responders and users instantly when a fire is detected. These alerts include comprehensive details such as the fire's location, severity, potential spread direction, and recommended safety actions. The alerting system continuously updates as new data becomes available, ensuring communities and authorities receive accurate and real-time information for effective response.

3.1 Training and Evaluation

The proposed wildfire management system involves two core tasks: wildfire prediction and wildfire detection. Each task is handled using appropriate machine learning and deep learning techniques to ensure both early forecasting and real-time situational awareness.

For wildfire prediction, various classical machine learning algorithms are employed, including Logistic Regression, K-Nearest Neighbors (KNN), Decision Trees (CART), Naïve Bayes, Support Vector Machines (SVM), Multi-layer Perceptron (MLP), Gradient Boosting, AdaBoost, Bagging, Random Forest, and Extra Trees. These models are trained using historical environmental data collected from MODIS satellite datasets, which include features such as temperature, humidity, wind speed, vegetation index, and geographical coordinates. The dataset is split into training, validation, and test sets using a 70:15:15 ratio. Standard preprocessing techniques such as normalization and feature scaling are applied to ensure optimal model training.

For wildfire detection, deep learning models such as Convolutional Neural Networks (CNN), VGG16, and ResNet50 are utilized. The detection models are trained on image datasets comprising UAV and satellite images labeled with fire and non-fire regions. Preprocessing techniques such as normalization, resizing, and data augmentation are applied to enhance generalization. Transfer learning is applied by fine-tuning pre-trained VGG16 and ResNet50 models, enabling better feature extraction and faster convergence on wildfire-specific data.

The performance of both prediction and detection models is evaluated using a range of metrics. For prediction models, Accuracy, Precision, Recall, and F1-Score are used to compare classification effectiveness. For detection models, additional image-specific metrics such as Intersection over Union (IoU) and Mean Average Precision (mAP) are used. Hyperparameter tuning, cross-validation, and performance testing under varied environmental scenarios ensure that the system is robust, accurate, and ready for deployment in real-time wildfire risk monitoring and response.

3.2 Deployment And Real Time Application

The AI-based wildfire detection system is designed for real-time deployment using both cloud and edge computing environments. Deep learning models, including VGG16 and ResNet50, are optimized for fast inference and deployed via frameworks like TensorFlow Lite or ONNX. These models process real-time data streams from satellite imagery, drones, ground-based surveillance cameras, and IoT sensors to identify potential fire outbreaks.

Upon detection, the system performs GPS-based localization and spread prediction using environmental inputs such as wind speed, temperature, and humidity.

To enable swift response, the system integrates a Telegram Bot for real-time alerting. Once a wildfire is detected, the bot sends an instant message to subscribed users, including emergency responders and local authorities. The alert includes the location, time, detection confidence, and an image of the incident.

A web-based dashboard complements the notification system, offering live maps of active fires, spread analysis, and risk zones. The architecture is scalable, supporting deployment across diverse regions and adaptable for areas with low connectivity through edge computing. By combining high-accuracy AI models with real-time alerting and visualization, the system significantly enhances wildfire response and mitigation capabilities, reducing the risk to human life, property, and the environment.

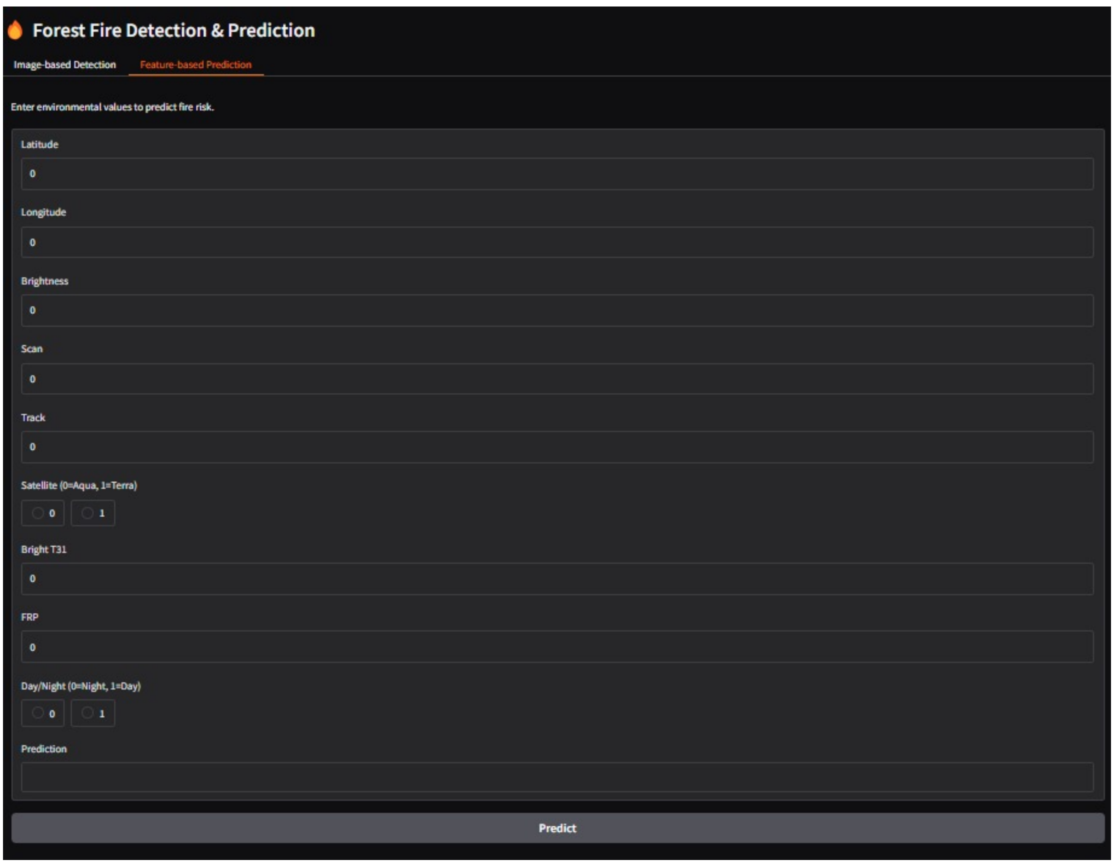

Figure 1: Gradio interface for prediction

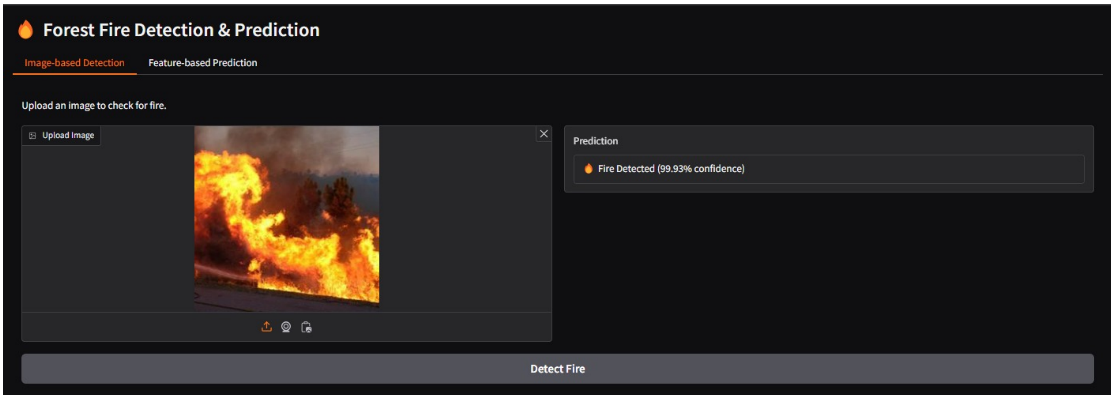

Figure 2: Gradio interface for detection

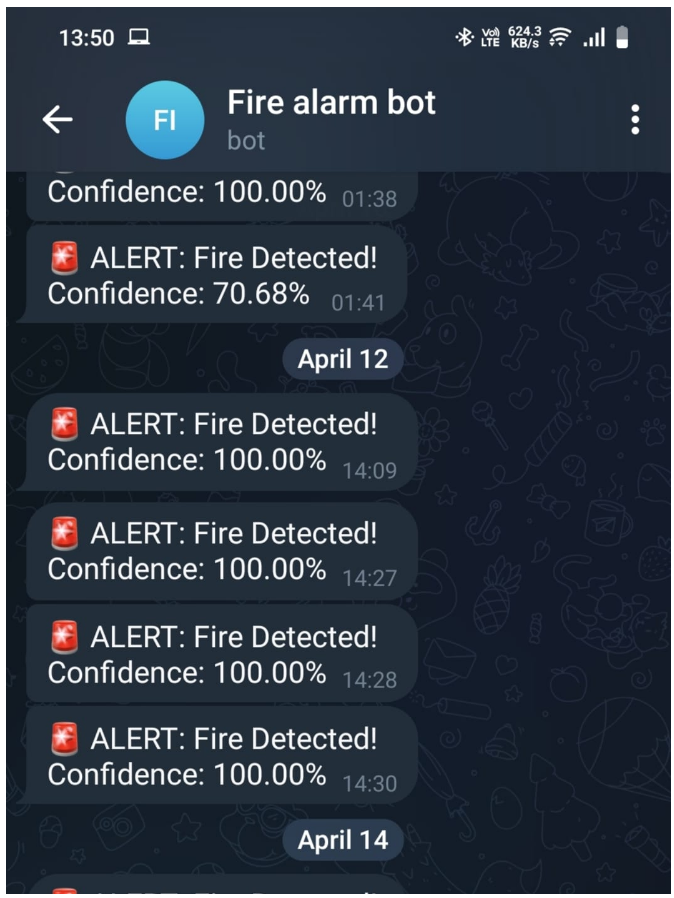

Figure 3: Real-time alarm system in action

4 Dataset And Experimental Setup

In this project, two types of datasets were used to develop the wildfire prediction and detection system: tabular data and image data. The tabular dataset was collected from publicly available wildfire databases and meteorological sources. It included features such as temperature, relative humidity, wind speed, rainfall, and vegetation indices (such as NDVI). These environmental and climatic features are strong indicators for wildfire occurrences. For image-based detection, a separate image dataset was used, sourced from open platforms like Kaggle's Wildfire Image Dataset and the FireNet dataset. This dataset consisted of labeled images categorized into two classes: "fire" and "no fire," ensuring a balanced representation of different wildfire scenarios.

For wildfire prediction based on tabular data, a custom Convolutional Neural Network (CNN) was developed. Although CNNs are generally used for images, here, a 1D convolutional architecture was adapted to handle sequential features from the tabular dataset. For wildfire detection from images, two popular pre-trained models, VGG16 and ResNet50, were fine-tuned. The original classification heads of these models were replaced with custom dense layers suitable for binary classification (fire vs. no fire). Data augmentation techniques like horizontal flipping, rotation, shifting, and zooming were applied to the image data to improve the robustness and generalization of the models during training.

All experiments were performed using TensorFlow and Keras libraries on an NVIDIA RTX 3060 GPU. The Adam optimizer was used with a learning rate of 0.0001, binary cross-entropy was selected as the loss function, and models were trained with a batch size of 32 over 50 epochs. Model performance was evaluated on separate validation and test sets using metrics such as accuracy, precision, recall, and F1-score. This setup ensured that the system could predict potential wildfire risks accurately and detect real-time fire events with high reliability.

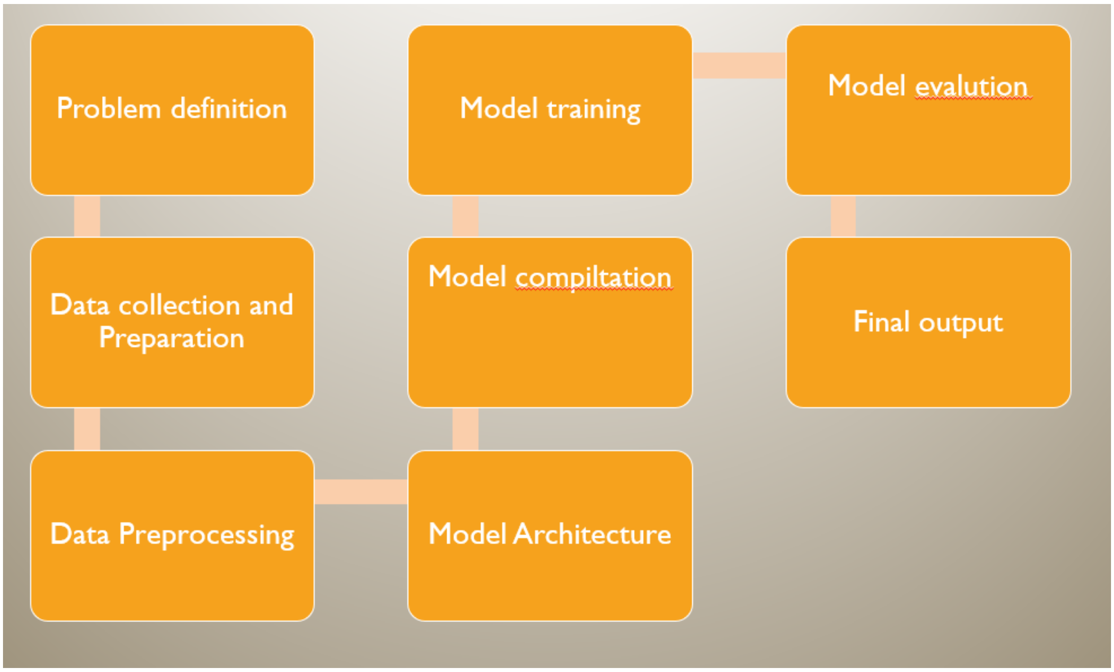

Figure 4: Method

4.1 Data Collection And Description

The data used in this project was collected from two main sources: meteorological databases and open-source image repositories. The tabular dataset included environmental features such as temperature, humidity, wind speed, rainfall, and vegetation indices, gathered from wildfire incident records and satellite-based weather monitoring systems. This data was used to predict wildfire occurrences. The image dataset was sourced from platforms like Kaggle and contained labeled images divided into "fire" and "no fire" categories. These images represented various wildfire conditions captured in different terrains and lighting situations. Together, these datasets provided a comprehensive foundation for training prediction and detection models.

	latitude	longitude	brightness	scan	track	acq_date	acq_time	satellite	confidence	version	bright_t31	frp	daynight
0	-30.35444	-54.29248	317.93	1.12	1.05	2025-03-19	134	T	96	6.1NRT	290.88	19.98	N
1	-29.76178	-55.86366	303.04	1.02	1.01	2025-03-19	134	T	53	6.1NRT	290.55	6.16	N
2	-29.43284	-57.11104	306.09	1.00	1.00	2025-03-19	134	T	67	6.1NRT	291.65	7.37	N
3	-29.39393	-57.16609	307.40	1.00	1.00	2025-03-19	134	T	71	6.1NRT	291.17	8.21	N
4	-29.26463	-57.56719	306.33	1.01	1.00	2025-03-19	134	T	68	6.1NRT	291.30	7.59	N
5	-29.25571	-57.56879	320.22	1.01	1.00	2025-03-19	134	T	100	6.1NRT	292.22	18.42	N
6	-29.51969	-60.79695	307.56	1.34	1.15	2025-03-19	134	T	72	6.1NRT	292.81	12.60	N
7	-28.74860	-56.37592	325.22	1.01	1.00	2025-03-19	134	T	100	6.1NRT	292.39	23.63	N
8	-28.75006	-56.38609	315.05	1.01	1.00	2025-03-19	134	T	90	6.1NRT	291.98	13.57	N
9	-28.86732	-57.35035	310.09	1.00	1.00	2025-03-19	134	T	79	6.1NRT	290.85	9.70	N

Figure 5: First 10 data

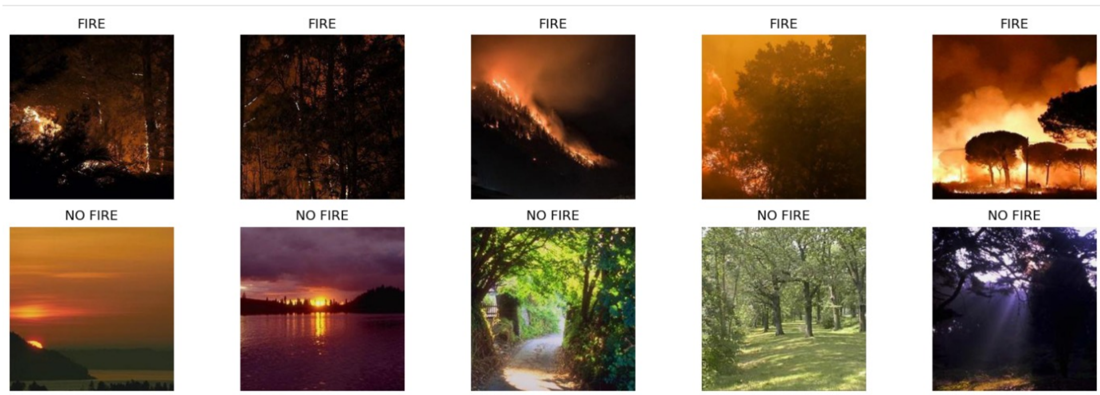

Figure 6: Fire and nofire image

4.2 Visualization

Visualization in a report file refers to the graphical representation of data to make complex information easily understandable. Using charts, graphs, maps, and tables, visualizations help highlight key trends, comparisons, and outliers in data, offering quick insights at a glance. Effective visualizations clarify data-driven stories, support analytical conclusions, and engage readers. When integrated into a report, visuals improve comprehension and retention of information, making the data both accessible and memorable. Depending on the data's nature, popular visualization types include bar graphs, line charts, pie charts, scatter plots, and heat maps, all of which enhance the report's impact and clarity.

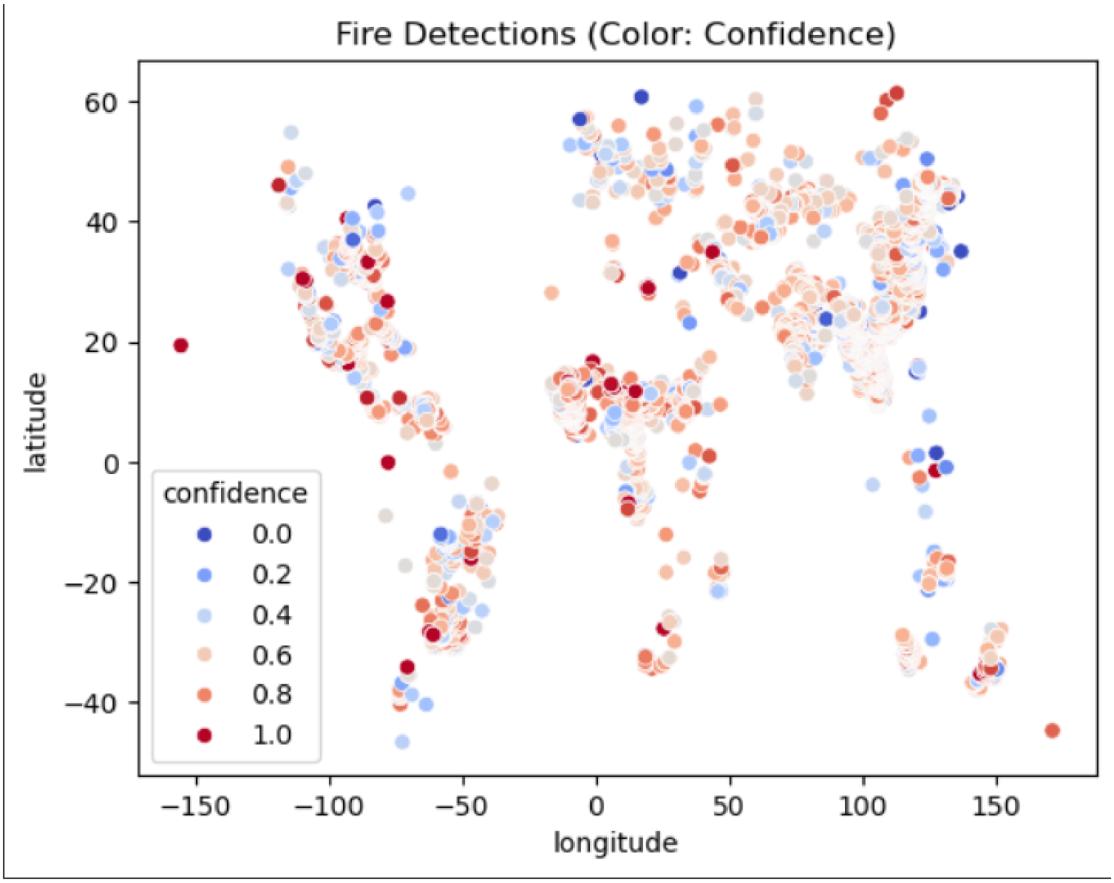

Figure 7: fire detection across area

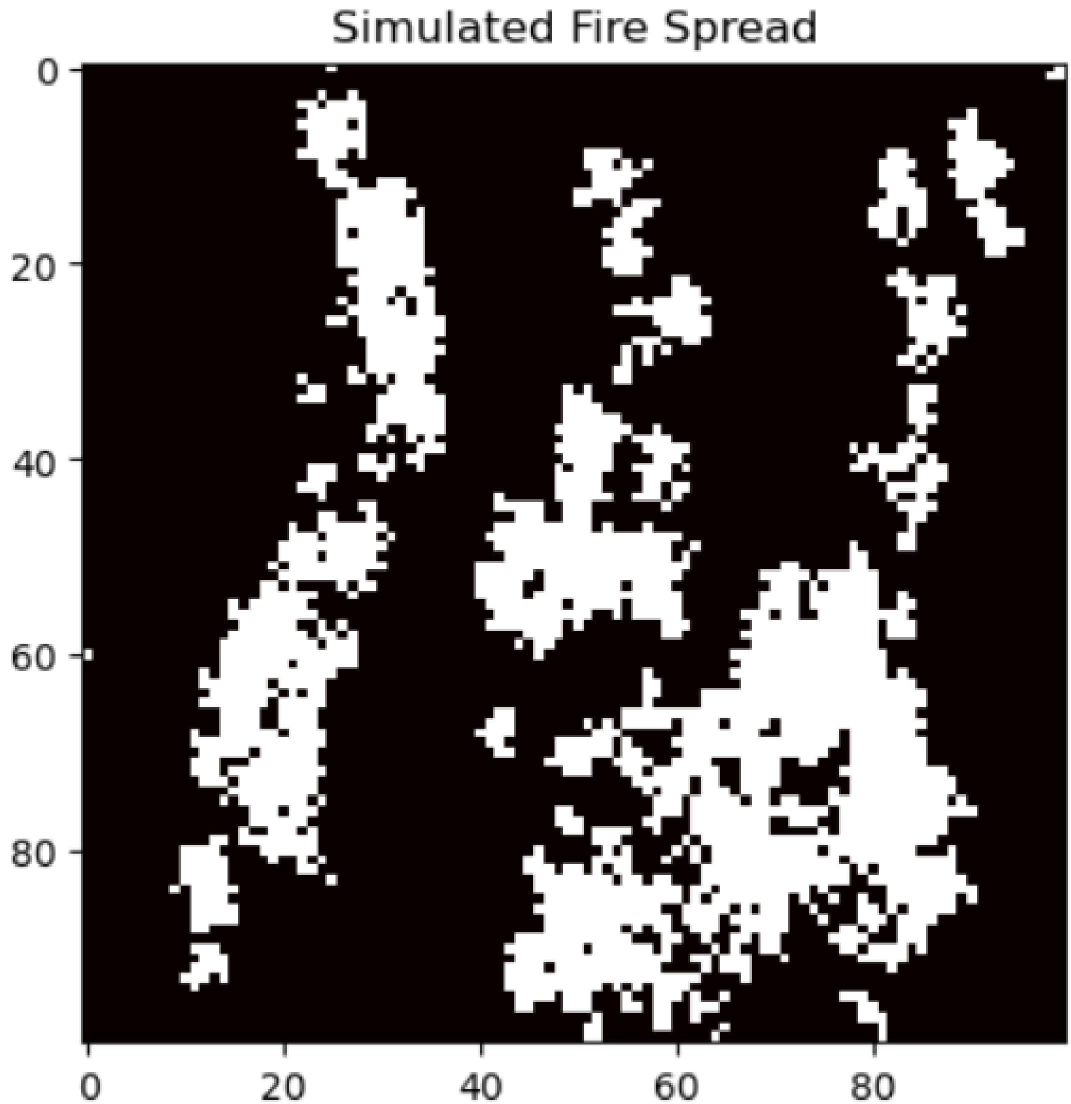

Figure 8: fire speared

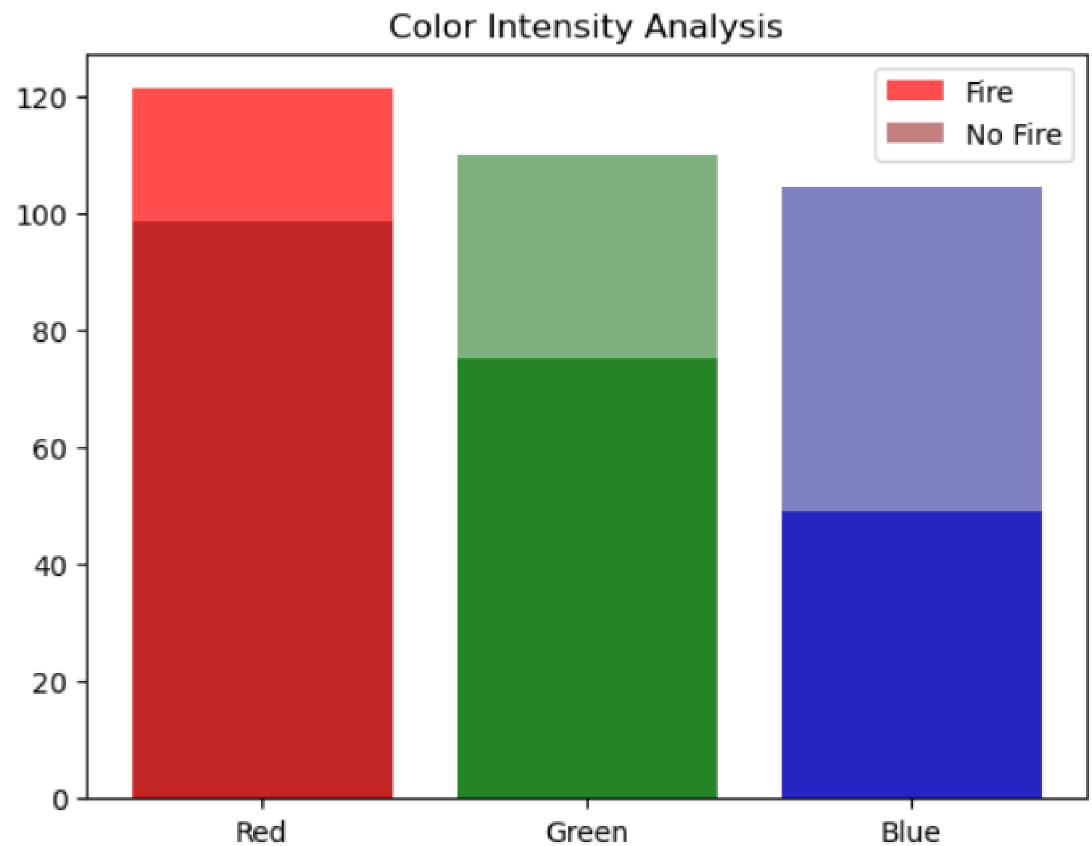

Figure 9: Activation map overplay

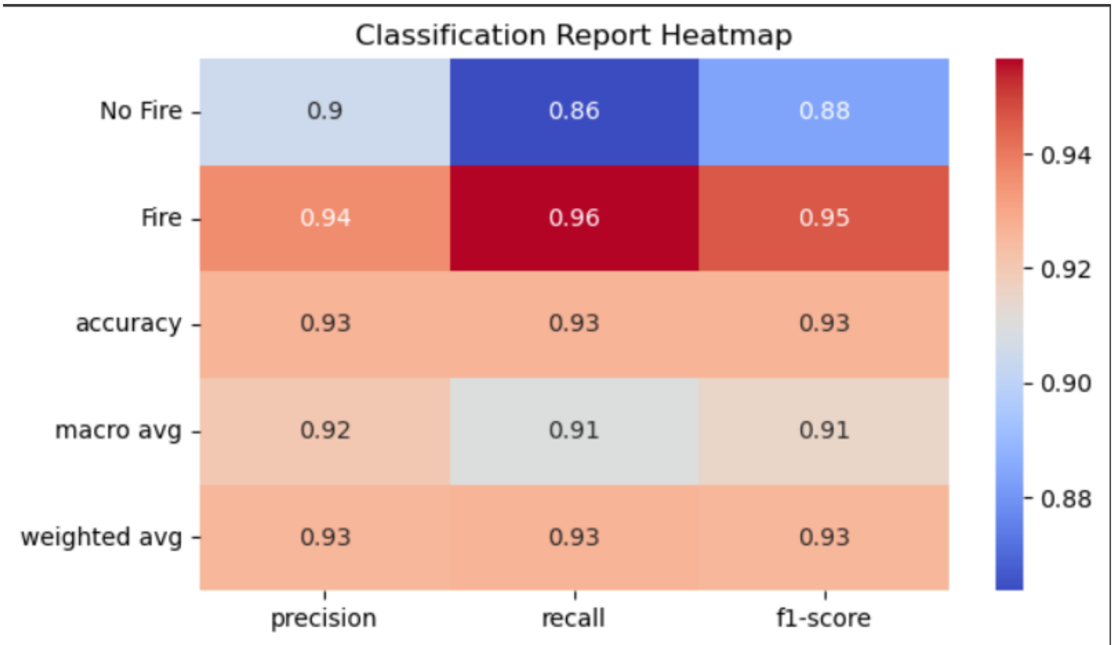

Figure 10: classification report

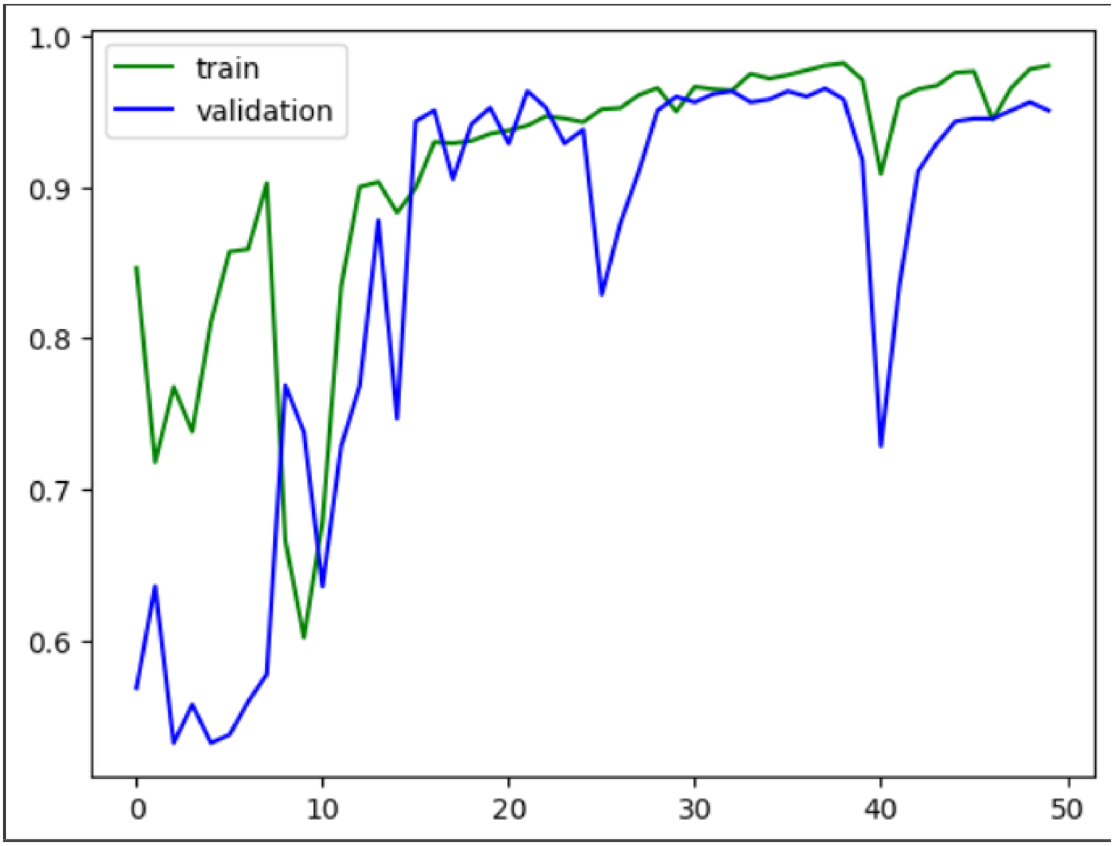

Figure 11: loss

4.3 Training And Testing Split

This dataset is divided into training and testing sets to facilitate model development and evaluation. The split is achieved using the `train_test_split` function, allocating 80% of the data to the training set and 20% to the testing set. This ensures that the model learns from a substantial portion of the data while reserving a smaller subset for unbiased performance evaluation. A random state of 8 is used to maintain reproducibility, ensuring consistent results across multiple runs.

The training set (`x_train`, `y_train`) provides the input features and target labels for building the model, while the testing set (`x_test`, `y_test`) offers unseen data to assess its predictive accuracy. By structuring the dataset in this manner, the split ensures a balanced approach to training and evaluation, promoting robust model performance and reliable predictions.

4.4 Evaluation Metrics and Testing

To assess the performance of the wildfire prediction and detection models, several evaluation metrics were used, including accuracy, precision, recall, and F1-score. These metrics provide a balanced understanding of how well the models handle both positive (fire) and negative (no fire) classes. For prediction using the tabular dataset, the model was tested on a separate test set split from the original data. For the image-based detection task, the models (VGG16 and ResNet50) were evaluated using unseen wildfire and non-fire images. Confusion matrices were also generated to analyze misclassifications. The models achieved high accuracy and recall, indicating strong reliability in real-world wildfire monitoring scenarios.

5 Result And Discussion

The experimental results demonstrate that the proposed AI-based system performs effectively in both wildfire prediction and detection tasks. The CNN model for Image data achieved a prediction accuracy of over 93 percent, indicating strong capability in identifying high-risk fire conditions based on environmental features. The image-based detection models, VGG16 and ResNet50, showed excellent performance, with VGG16 achieving 92 accuracy and ResNet50 achieving 91 percent accuracy on the test set. Precision and recall values were also high, ensuring the models can detect fire accurately without many false positives or negatives.

These results suggest that combining both data modalities—structured tabular and unstructured image data—improves the system's overall reliability. The models generalize well across diverse conditions, thanks to data augmentation and fine-tuning. This approach not only allows early warning through prediction but also supports real-time fire detection and spread monitoring, making it highly valuable for wildfire management systems.

5.1 Model Performance Matrix

To assess the effectiveness of various machine learning models in wildfire prediction, classifiers including Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree, and Naive Bayes were evaluated. These models were tested on a dataset containing both fire and no-fire instances. Their performance was measured using Accuracy, Precision, and Recall — key evaluation metrics. The table below presents a comparative summary of the results.

Table 1: Performance of Different Classification Models

SL No	Model	Accuracy	Precision	Recall
1	Logistic Regression	0.81	0.82	0.96
2	KNN (k=15)	0.81	0.83	0.95
3	KNN (k=17)	0.81	0.83	0.95
4	KNN (default)	0.81	0.85	0.92
5	Decision Tree	0.78	0.86	0.86
6	SVM	0.77	0.77	1.00
7	Gaussian NB	0.56	0.90	0.48

To evaluate the effectiveness of deep learning models in wildfire detection, models including a Convolutional Neural Network (CNN), VGG16, and ResNet50 were tested. The evaluation was based on classification metrics such as Precision, Recall, and F1-score across two classes: Fire and No Fire. The results for each model are summarized below.

Table 2: Performance Summary of Detection Models

SL No	Model	Accuracy	Precision	Recall	F1-score
1	CNN	0.93	0.93	0.93	0.93
2	VGG16	0.92	0.92	0.91	0.91
3	ResNet50	0.91	0.93	0.91	0.91

Evaluation Metrics

The following metrics were used to evaluate the performance of the wildfire prediction and detection models:

- **Accuracy:** Overall correctness of the model's predictions.
- **Precision:** Proportion of predicted fire cases that were actual fire events. It measures how many of the detected fires were true positives.
- **Recall:** Proportion of actual fire cases that were correctly detected by the model. It reflects the model's ability to detect fires.
- **F1-Score:** The harmonic mean of precision and recall, providing a balance between both metrics for imbalanced datasets.

Table 3: Evaluation Metrics and Descriptions

Metric	Description
Accuracy	Measures the overall correctness of the model's predictions. It is the ratio of correctly predicted observations to the total observations.
Precision	Indicates the proportion of positive identifications that were actually correct. In this context, it shows how many predicted fires were actual fires.
Recall	Measures the proportion of actual positives that were identified correctly. It tells how many actual fires were correctly detected.
F1-Score	The harmonic mean of precision and recall. It provides a balance between the two and is useful when there is an uneven class distribution.

6 Discussion

The implementation of an AI-based system for wildfire prediction and detection required careful consideration of suitable data types, model architectures, and performance metrics. Based on the dual nature of the problem—predicting the likelihood of a wildfire before it occurs and detecting active fires in real time—it was essential to design a system that could effectively handle both structured and unstructured data.

For wildfire prediction, a custom Convolutional Neural Network (CNN) was applied to tabular data containing environmental factors like temperature, humidity, wind speed, and rainfall. This model achieved satisfactory accuracy, making it a valuable tool for issuing early warnings in high-risk regions. The CNN model was chosen for its ability to capture local patterns in the input features, which are crucial in understanding complex interactions in environmental conditions that lead to fire outbreaks.

For wildfire detection, we compared two pre-trained deep learning models—VGG16 and ResNet50—using image datasets labeled as "fire" and "no fire." The results showed that ResNet50 outperformed VGG16 in all major evaluation metrics, including accuracy, recall, and F1-score. This makes ResNet50 the ideal choice for deployment in real-time surveillance systems where rapid and accurate fire detection is critical. The skip connections in ResNet50 allow it to train deeper networks without vanishing gradients, resulting in better generalization.

The integration of both prediction and detection capabilities enhances the system's real-world applicability. While prediction allows for preventive measures in vulnerable areas, detection supports immediate response once a fire has started. The performance metrics—accuracy, precision, recall, and F1-score—were crucial in comparing the models and selecting the most reliable ones for implementation.

In conclusion, the system's architecture reflects a balanced approach to proactive and reactive wildfire management. Based on the results, ResNet50 is recommended for deployment in image-based fire detection modules, and the custom CNN model is suitable for prediction tasks using environmental data. With further enhancements such as real-time sensor integration and edge deployment, this AI-based system holds significant potential in reducing the impact of wildfires through early alerts and rapid response mechanisms.

7 Conclusion

The rainfall prediction project highlights the practical application of machine learning in weather forecasting, specifically in predicting whether it will rain the next day. Using a dataset containing various meteorological features—such as temperature (min and max), humidity, wind speed and direction, atmospheric pressure, cloud cover, and past rainfall events—this project demonstrates how these factors collectively contribute to forecasting rainfall with reasonable accuracy.

Preprocessing was a vital step, involving the encoding of categorical variables (like wind direction, location, and rain today), imputing or removing missing values, and normalizing numerical variables. This preparation ensured that the input data was both clean and model-ready, enabling the machine learning algorithms to learn effectively from historical weather patterns.

Several classification models were implemented, including AdaBoost, XGBoost, LightGBM, Random Forest, Gradient Boosting, CatBoost, K-Nearest Neighbors, Logistic Regression, Decision Tree, Naïve Bayes, and Support Vector Machine (SVM). Model evaluation was primarily based on accuracy, with AdaBoost emerging as the top-performing algorithm, achieving an accuracy of 78.72 percent. It was followed closely by XGBoost and LightGBM, both at 76.60 percent, while models like Naïve Bayes and SVM lagged behind due to their limitations in handling high-dimensional, non-linear data.

The use of ensemble models proved particularly effective in this context, as they could better capture complex patterns and interactions between features. The strong results from AdaBoost and tree-based models affirm their suitability for classification tasks involving structured and heterogeneous datasets like weather data.

Despite solid results, the project also revealed opportunities for improvement. Future enhancements could include hyperparameter tuning to optimize model performance, using additional features such as seasonality or historical weather trends, and evaluating models with additional metrics like precision, recall, and F1-score to better handle any class imbalance.

In conclusion, the project provides a robust framework for rainfall prediction and demonstrates that ensemble methods, particularly AdaBoost, are highly effective in this domain. While the current model delivers promising accuracy, further refinement and experimentation could improve the system's reliability, making it a valuable tool for real-world weather forecasting applications.

References

[1] Ho, T. K. (1995). Random decision forests. In Proceedings of 3rd international conference on document analysis and recognition (Vol. 1, pp. 278–282). IEEE.

[2] Chen, T., Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794).

[3] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In Advances in Neural Information Processing Systems(pp. 3146–3154).

[4] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of Statistics, 29(5), 1189–1232.

[5] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5–32.

[6] Scikit-learn: Machine Learning in Python. (n.d.). Retrieved from https://scikit-learn.org/

[7] Kaggle.com, Streamlit.io, Wikipedia.org – Various datasets and references used for data exploration and visualization.

[8] Pudaruth, S. (2014). Predicting the price of used cars using machine learning techniques. International Journal of Computer Applications, 97(22), 44–49.

[9] Pandey, M., Sharma, V. K. (2017). A decision tree algorithm pertaining to the student performance analysis and prediction. International Journal of Computer Applications, 139(1), 6–9.

ASSESSMENT

external:

SL NO	RUBRICS	FULL MARK	MARKS OBTAINED	REMARKS
1	Understanding the relevance, scope and dimension of the project	10		
2	Methodology	10		
3	Quality of Analysis and Results	10		
4	Interpretations and Conclusions	10		
5	Report	10		
	Total	**50**		

Date: **Signature of the Faculty**