A MACHINE LEARNING FRAMEWORK FOR FOREST FIRE PREDICTION IN THE NALLAMALA FOREST USING NDVI AND SYNTHETIC WEATHER DATA

A Main Project Report submitted in the partial fulfillment of the requirements for the award of the degree.

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Gairuboina Naveen Kumar (22471A05F5)

Dogiparthi Venkata Sai Girish (22471A05F2)

Sanikommu Nirupam Reddy (22471A05I6)

Under the esteemed guidance of

Dr. S.Siva Nageswara Rao, M.Tech.,Ph.D.

Professor.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPET

(AUTONOMOUS)

Accredited by NAAC with A+ Grade and NBA under Tier-I

an ISO 9001:2015 Certified

Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada

KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET- 522601

2025 – 2026

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project that is entitled with the name "A MACHINE LEARNING FRAMEWORK FOR FOREST FIRE PREDICTION IN THE NALLAMALA FOREST USING NDVI AND SYNTHETIC WEATHER DATA" is a bonafide work done by the team GAIRUBOINA NAVEEN KUMAR (22471A05F5), DOGIPARTHI VENKATA SAI GIRISH (22471A05F2), SANIKOMMU NIRUPAM REDDY(22471A05I6) in partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in the Department of COMPUTER SCIENCE AND ENGINEERING during 2025-2026.

PROJECT GUIDE

Dr. S. Siva Nageswara Rao, M.Tech.,Ph.D.

Professor

PROJECT COORDINATOR

D.Venkata Reddy, M.Tech., (Ph.D).

Assistant Professor

HEAD OF THE DEPARTMENT

Dr. S. N. Tirumala Rao, M.Tech., Ph.D.

Professor & HoD

EXTERNAL EXAMINER

DECLARATION

We declare that this project work titled "**A MACHINE LEARNING FRAMEWORK FOR FOREST FIRE PREDICTION IN THE NALLAMALA FOREST USING NDVI AND SYNTHETIC WEATHER DATA**" is composed by ourselves that the work contain here is our own except where explicitly stated otherwise in the text and that this work has not been submitted for any other degree or professional qualification except as specified.

Gairuboina Naveen Kumar (22471A05F5)

Dogiparthi Venkata Sai Girish (22471A05F2)

Sanikommu Nirupam Reddy (22471A05I6)

ACKNOWLEDGEMENT

We wish to express our thanks to various personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairman **Sri M. V. Koteswara Rao,** B.Sc., who took keen interest in our progress and efforts throughout this course. We owe our sincere gratitude to our beloved principal, **Dr. S. Venkateswarlu,** M.Tech., Ph.D. for showing his kind attention and valuable guidance throughout the course.

We express our deep-felt gratitude towards **Dr. S. N. Tirumala Rao,** M.Tech., Ph.D. HoD of CSE Department and to our guide **Dr. S. Siva Nageswara Rao,** M.Tech., Ph.D. Professor of CSE Department whose valuable guidance and unstinting encouragement enabled us to accomplish our project successfully in time.

We extend our sincere thanks to **D. Venkata Reddy,** M.Tech., (Ph.D.) Assistant Professor & Project Coordinator of the project for extending his encouragement. His profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We extend our sincere thanks to all other teaching and non-teaching staff in the department for their cooperation and encouragement during our B.Tech degree.

We have no words to acknowledge the warm affection, constant inspiration and encouragement that we received from our parents.

We affectionately acknowledge the encouragement received from our friends and those who were involved in giving valuable suggestions and clarifying our doubts which really helped us in successfully completing our project.

By

Gairuboina Naveen Kumar (22471A05F5)

Dogiparthi Venkata Sai Girish (22471A05F2)

Sanikommu Nirupam Reddy (22471A05I6)

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community.

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research.

M2: Build a passionate and a determined team of faculty with student centric teaching, imbibing experiential, innovative skills.

M3: Imbibe lifelong learning skills, entrepreneurial skills, and ethical values in students for addressing societal problems.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

To become a centre of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

M1: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertise in modern software tools and technologies to cater to the real time requirements of the industry.

M3: Inculcate teamwork and lifelong learning among students with a sense of societal and ethical responsibilities.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Specific Outcomes (PSO's)

PSO1: Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-based systems.

PSO2: Acquaint module knowledge on emerging trends of the modern era in Computer Science and Engineering

PSO3: Promote novel applications that meet the needs of entrepreneur, environmental and social issues.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Educational Objectives (PEO's)

The graduates of the program are able to:

PEO1: Apply the knowledge of Mathematics, Science and Engineering fundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems related to academia, industry, and society.

PEO3: Work with ethical and moral values in multi-disciplinary teams and can communicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in the software industry.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Outcomes (PO'S)

PO1: Engineering Knowledge: Apply knowledge of mathematics, natural science,computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)

PO3: Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)

PO4: Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).

PO5: Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)

PO6: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).

PO7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)

PO8: Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

PO9: Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences

PO10: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and
in multidisciplinary environments.

PO11: Life-Long Learning: Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Project Course Outcomes (CO'S)

CO421.1: Analyse the System of Examinations and identify the problem.

CO421.2: Identify and classify the requirements.

CO421.3: Review the Related Literature

CO421.4: Design and Modularize the project

CO421.5: Construct, Integrate, Test and Implement the Project.

CO421.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
C421.1		✓										✓		
C421.2	✓		✓		✓							✓		
C421.3				✓		✓	✓	✓				✓		
C421.4			✓			✓	✓	✓				✓	✓	
C421.5					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
C421.6									✓	✓	✓	✓	✓	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
C421.1	2	3										2		
C421.2			2		3							2		
C421.3				2		2	3	3				2		
C421.4			2			1	1	2				3	2	
C421.5					3	3	3	2	3	2	2	3	2	1
C421.6									3	2	1	2	3	

Note: The values in the above table represent the level of correlation between CO's and PO's

1. Low level

2. Medium level

3. High level

Project mapping with various courses of Curriculum with Attained PO's:

Name of the course from which principles are applied in this project	Description of the device	Attained PO
C2204.2, C22L3.2	Gathering the requirements and defining the problem, plan to develop model for detection and classification of NallaFirenet	PO1, PO3,PO8
CC421.1, C2204.3, C22L3.2	Each and every requirement critically analyzed, the process mode is identified	PO2, PO3,PO8
CC421.2, C2204.2, C22L3.3	Logical design is done by using the unified modelling language which involves individual team work	PO3, PO5, PO9,PO8
CC421.3, C2204.3, C22L3.2	Each and every module is tested, integrated, and evaluated in our project	PO1, PO5,PO8
CC421.4, C2204.4, C22L3.2	Documentation is done by all our Three members in the form of a group	PO10,PO8
CC421.5, C2204.2, C22L3.3	Each and every phase of the work in group is presented periodically	PO8,PO10, PO11
C2202.2, C2203.3, C1206.3, C3204.3, C4110.2	Implementation is done and the project will be handled by the social media users and in future updates in our project can be done based on Prediction of Forest Fires.	PO4, PO7,PO8
C32SC4.3	The physical design includes website to check NallaFirenet	PO5, PO6,PO8

ABSTRACT

This study presents **NallaFireNet**, a comprehensive machine learning-based framework developed for early prediction of forest fires in the ecologically sensitive Nallamala Forest region of India. The model integrates multi-temporal Normalized Difference Vegetation Index (NDVI) data derived from MODIS with synthetic meteorological parameters—temperature, humidity, solar radiation, and precipitation—sourced from NASA POWER datasets spanning 2012–2025. By combining vegetation health indicators with weather dynamics and applying temporal lag features, the system captures critical spatiotemporal dependencies influencing wildfire occurrence. The research addresses the inherent challenge of class imbalance between fire and non-fire events using the Synthetic Minority Oversampling Technique (SMOTE), enabling improved sensitivity toward rare fire instances. An ensemble learning approach was employed, combining XGBoost and LightGBM models within a soft voting structure to enhance predictive robustness and generalization. Experimental evaluations demonstrated an overall accuracy of 91.46% and a recall of 69% for fire events, confirming the framework's reliability for early warning purposes. The analysis of feature importance revealed that temperature and relative humidity significantly influence fire susceptibility, aligning with known ecological drivers in tropical dry deciduous ecosystems. The results affirm that the fusion of remotely sensed NDVI data with synthetic weather information and advanced ensemble learning techniques can offer a proactive, scalable solution for wildfire risk monitoring. Designed for potential real-time integration into forest management systems, NallaFireNet contributes to climate resilience by enabling timely alerts, data-driven resource allocation, and improved preparedness in vulnerable forest regions.

INDEX

LIST OF FIGURES

1.INTRODUCTION

Forests play a pivotal role in sustaining life on Earth, acting as vital carbon sinks, biodiversity reserves, and regulators of global climate. In India, the Nallamala Forest — spanning parts of Andhra Pradesh and Telangana — represents one of the country's most ecologically significant regions, home to endangered species such as the Indian tiger (*Panthera tigris tigris*). However, its tropical dry deciduous ecosystem makes it particularly vulnerable to seasonal wildfires, especially during prolonged dry spells and droughts.

In recent years, the frequency and intensity of wildfires in the Nallamala region have increased considerably, yet early warning and rapid response systems remain largely inadequate. Wildfire occurrences are often driven by a combination of natural and human-induced factors — including lightning strikes, heatwaves, shifting cultivation, and human negligence — while the forest's dense and inaccessible terrain hampers effective surveillance and emergency intervention. Existing forecasting models, such as thermal anomaly detection from MODIS and VIIRS satellites, remain predominantly reactive, detecting fires only after ignition and offering limited lead time for preventive measures.

Recognizing these challenges, this study introduces **NallaFireNet**, a machine learning–based framework specifically designed for the early prediction of forest fires in the Nallamala Forest. The framework integrates multi-temporal **Normalized Difference Vegetation Index (NDVI)** data with **synthetic weather parameters**—including temperature, humidity, and solar radiation—derived from NASA POWER datasets. By combining these environmental indicators with historical fire event records, the model captures spatiotemporal patterns of vegetation stress and climate variability that precede fire occurrences.

To address the problem of severe class imbalance between fire and non-fire days, the framework employs the **Synthetic Minority Oversampling Technique (SMOTE)**, ensuring improved model sensitivity toward rare fire events. Ensemble algorithms—**LightGBM** and **XGBoost**—are then applied to exploit nonlinear relationships and

enhance predictive accuracy. This combination enables robust, explainable, and data-efficient forecasting suitable for operational deployment in forest management systems.

Ultimately, NallaFireNet demonstrates the feasibility of AI-driven early-warning systems tailored to data-scarce, ecologically sensitive regions. By coupling remote sensing, synthetic climate modeling, and ensemble learning, the proposed framework aims to strengthen wildfire preparedness, reduce ecological loss, and promote proactive forest management in the face of escalating climate risks.

Fig1:Forest Fires

1.1 MOTIVATION

The growing frequency and intensity of wildfires across India, particularly in the Nallamala Forest region, underscore the urgent need for predictive technologies that can move beyond traditional reactive monitoring systems. Existing fire management practices in India largely depend on manual patrols or satellite-based thermal anomaly detection systems, such as MODIS and VIIRS, which detect fires only after ignition. These post-event detection methods fail to offer sufficient lead time for pre-emptive action, resulting in delayed responses, significant ecological damage, and loss of biodiversity.

The Nallamala Forest, located within the Eastern Ghats, presents unique challenges: it is vast, topographically complex, and often inaccessible to forest authorities. Seasonal droughts and extended dry periods make the ecosystem particularly fire-prone, while limited ground-based infrastructure restricts real-time monitoring. Furthermore, the

scarcity of localized meteorological data hinders accurate fire forecasting and risk assessment. This data gap calls for the use of **synthetic weather datasets** and **satellite-derived vegetation indices** such as NDVI to simulate and monitor environmental conditions that influence fire occurrence.

From a research standpoint, there exists a significant gap between existing forest fire detection models—mostly built for post-event analysis—and predictive, region-specific frameworks capable of identifying fire-prone conditions beforehand. While machine learning and remote sensing have independently shown promise, their combined use for **spatiotemporal wildfire prediction in Indian ecosystems** remains largely unexplored. Moreover, issues such as **data imbalance**, where fire events are rare compared to non-fire days, further complicate model training and limit the performance of conventional algorithms.

Addressing these gaps, the proposed **NallaFireNet framework** seeks to create an interpretable and scalable machine learning model that can learn from NDVI trends and synthetic weather patterns to predict potential fire days. By employing ensemble learning algorithms like **XGBoost** and **LightGBM**, and balancing rare events through **SMOTE**, the framework demonstrates how advanced computational methods can enable **early warning systems** for proactive forest management.

Ultimately, this research is motivated by the pressing need to safeguard one of India's most ecologically critical forests through technological innovation. It reflects a broader commitment to leveraging artificial intelligence, remote sensing, and climate data integration to enhance environmental resilience and reduce the devastating impacts of wildfires.

1.2 PROBLEM STATEMENT

Wildfires in the Nallamala Forest have become increasingly frequent and severe, posing significant threats to biodiversity, human settlements, and the ecological stability of the Eastern Ghats region. Despite the availability of satellite-based monitoring systems such as MODIS and VIIRS, current approaches remain **reactive**—detecting fires only

after ignition rather than predicting them in advance. This lack of **early warning capability** leads to delayed interventions, uncontrolled fire spread, and large-scale ecological and economic losses.

The **inaccessibility** of vast forest terrain, combined with limited on-ground surveillance and insufficient meteorological infrastructure, further constrains timely fire detection and management. Additionally, existing predictive models struggle to function effectively in **data-scarce environments** like Nallamala, where localized climatic and vegetation data are often unavailable or incomplete.

Another major challenge is the **imbalance in fire event data**, where occurrences of fire days are extremely rare compared to non-fire days. This imbalance causes traditional machine learning models to become biased toward the majority class, resulting in low sensitivity and poor recall for actual fire events. Furthermore, most existing studies focus on **post-fire analysis or detection**, rather than **proactive forecasting**, and lack adaptation to India's heterogeneous forest ecosystems.

Therefore, the core problem addressed in this research is the **absence of an accurate, interpretable, and region-specific predictive framework** that can integrate **remote sensing data (NDVI)** with **synthetic meteorological parameters** to forecast fire occurrences with measurable lead times.

To solve this, the proposed **NallaFireNet** system aims to design a **machine learning–based ensemble model** capable of learning from vegetation dynamics and climate patterns to provide **reliable early warnings** for forest fire risk in the Nallamala region.

1.3 OBJECTIVE

The primary objective of this research is to design and develop a **machine learning–based predictive framework**, named **NallaFireNet**, capable of forecasting forest fire occurrences in the Nallamala Forest region. The framework aims to integrate multi-source environmental data, particularly satellite-derived **Normalized Difference Vegetation Index (NDVI)** and **synthetic meteorological variables** such as temperature,

humidity, solar radiation, and precipitation, to generate early warnings for potential fire events. By doing so, the study seeks to move beyond traditional reactive fire detection systems and introduce a proactive, data-driven solution for forest fire management.

To achieve this main objective, several specific goals have been outlined. First, the study focuses on collecting, integrating, and preprocessing remote-sensing and climate datasets. This involves obtaining NDVI data from MODIS satellites and weather parameters from the NASA POWER database, followed by temporal alignment, missing-value handling, and normalization to ensure consistency and reliability in the modeling process. Additionally, the research incorporates feature engineering techniques to extract lag-based variables representing short-term changes in temperature, humidity, and vegetation health, which are crucial indicators of fire susceptibility.

Another objective of this study is to address the issue of **class imbalance**—a common challenge in wildfire prediction where fire events are extremely rare compared to non-fire days. To mitigate this problem, the **Synthetic Minority Oversampling Technique (SMOTE)** is employed to enhance the representation of minority fire cases, thus improving the model's ability to detect rare but critical events. The development and optimization of a robust **ensemble model** form the next key goal. The framework utilizes algorithms such as **LightGBM** and **XGBoost**, which are known for their efficiency and high performance in handling large-scale, imbalanced, and nonlinear datasets. These models are combined in a voting or stacking ensemble architecture to improve overall predictive accuracy, recall, and interpretability.

The research further aims to evaluate the performance of the proposed model using comprehensive metrics such as **accuracy, precision, recall, F1-score, and AUC-ROC**, along with visual tools like confusion matrices and ROC curves to validate its effectiveness. The ultimate objective is to design a scalable and interpretable system that can be seamlessly integrated into existing forest management operations. Through this integration, **NallaFireNet** is envisioned to generate real-time alerts, visualize risk levels, and support forest authorities in implementing timely and proactive measures against wildfire threats.

Overall, this study seeks to demonstrate that the combination of **remote sensing, synthetic climate modeling, and machine learning** can provide a powerful, region-specific early warning system for forest fire prediction, contributing significantly to environmental conservation and sustainable forest management.

2.LITERATURE SURVEY

The field of **forest fire prediction** has evolved rapidly with the integration of **remote sensing technologies, machine learning (ML), and artificial intelligence (AI)**. Researchers have focused on improving early detection accuracy, predictive efficiency, and interpretability of models to support environmental monitoring and disaster management. These studies explore various approaches, including satellite data analysis, deep learning architectures, ensemble models, and hybrid frameworks that integrate climatic, topographic, and anthropogenic variables. The following literature survey presents key contributions from various authors, highlighting their methodologies, findings, and the impact of their research on wildfire detection and prediction. These advancements have laid the foundation for the development of proactive, AI-driven wildfire management systems like **NallaFireNet**, which combine spatiotemporal modeling and ensemble learning for reliable early warning.

Y. Yu et al. [1] employed **nighttime light (NTL)** data acquired from the Suomi NPP satellite to detect forest fires in Southwest China using a **Random Forest classifier**. Their study effectively distinguished burning areas from urban illumination but remained a post-fire detection method rather than an early warning approach.

Z. S. K. Chaitanya et al. [2] compared standard machine learning models, including **Random Forest, SVM, and Naïve Bayes**, emphasizing the importance of **data preprocessing techniques** such as SMOTE-Tomek and correlation-based feature selection to improve classification performance.

AA. G. M. I. Alam et al. [3] introduced **FireNet-CNN**, a deep learning model enhanced by **explainable AI (XAI)** tools like Grad-CAM and saliency mapping, achieving 99.05% accuracy for real-time fire detection through visual interpretation. Although highly effective, their model focused on detection using binary image inputs rather than predictive spatiotemporal modeling.

N. K. Ojha and M. Katoch [4] proposed a **multimodal fusion-based LSTM–CNN architecture** for wildfire risk assessment, which significantly improved dynamic fire prediction by learning temporal dependencies.

O. M. Sivanuja et al. [5] advanced this further by ensembling deep learning models such as **InceptionV3, ResNet50, and VGG19**, achieving greater detection robustness.
P. H. Jo et al. [6] developed **FLAM-Net**, a hybrid AI and process-based model combining climatic, topographic, and anthropogenic data to estimate fire probability in South Korea, demonstrating strong adaptability under changing climate conditions.
Q. Similarly, N. M. J. Swaroopan and A. J. M. Rani [7] applied **optimized K-means clustering** with SVM to represent climate-induced fire risks, revealing the value of unsupervised feature grouping in fire analysis.

N. Datta et al. [8] integrated **Explainable AI (XAI)** with machine learning for forest fire prediction using **logistic regression and SHAP analysis**, improving model interpretability while balancing class imbalance through **SMOTE**.O. T. S. R. Raj et al. [9] presented **WiSEFire**, a GRU-based IoT-driven AI system designed for real-time wildfire monitoring in vulnerable ecosystems, demonstrating high energy efficiency and scalability. Mohamed et al. [12] evaluated multiple ML models on a small dataset and found **Random Forest** to outperform others with 86.46% accuracy, identifying temperature and Fire Weather Index (FWI) as dominant predictors.S. Barik et al. [13] developed a **Random Forest Regressor** integrated with Fire Weather Index parameters, achieving 86% accuracy and demonstrating how real-time environmental sensors (temperature, humidity, wind, rainfall) can strengthen model reliability.

Further, P. Moral et al. [14] implemented various **regression-based ML models** using MODIS satellite data for forecasting forest fires in Jharkhand, India, with **Gradient Boosting Regressor** attaining the highest accuracy ($R^2 = 1.00$).

J. Jang et al. [15] explored **UAV-based fire detection**, integrating visible and infrared imagery via deep learning fusion, achieving superior performance in accuracy and detection speed.

K.Y. Zhang et al. [16] proposed a **real-time YOLOv8-based system** for fire detection using surveillance video streams, offering high accuracy across diverse environmental conditions.

Collectively, these studies demonstrate remarkable progress in forest fire monitoring through AI and remote sensing integration. However, most existing models are **reactive**—focused on post-fire detection—rather than **predictive frameworks** capable of providing early warnings. Few approaches have been tailored specifically to **Indian ecosystems**, which are characterized by sparse data availability, diverse vegetation, and complex topography. To address these limitations, the current research introduces **NallaFireNet**, a region-specific, machine learning-based predictive system designed for the **Nallamala Forest**. By combining **NDVI satellite imagery** with **synthetic meteorological data** and applying **ensemble ML algorithms** such as XGBoost and LightGBM with **SMOTE balancing**, this framework achieves improved sensitivity to rare fire events and supports proactive forest management. The literature collectively underlines the growing necessity of interpretable, data-efficient, and scalable AI systems to strengthen early fire prediction, environmental resilience, and sustainable forest protection.

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In the current scenario, most forest fire management systems rely heavily on **reactive detection mechanisms** rather than **predictive modeling**. Conventional approaches primarily depend on **satellite-based thermal anomaly detection** systems such as **MODIS (Moderate Resolution Imaging Spectroradiometer)** and **VIIRS (Visible Infrared Imaging Radiometer Suite)**, which identify fire hotspots after ignition. While these tools are effective for real-time monitoring, they fail to provide **early warning capabilities**, limiting the time available for preventive measures and emergency response.

Additionally, **manual surveillance** and **alert-based monitoring** methods are widely used by forest departments, where fire detection depends on **visual inspection**, **field reports**, or **human patrols**. These manual methods are often slow, prone to human error, and inefficient in **dense or inaccessible forest regions** like the Nallamala Forest. The vast expanse and rugged terrain hinder quick detection and timely mitigation, resulting in greater ecological and economic damage.

Most existing models also exhibit **data limitations**, particularly in regions with **incomplete environmental records** or **inconsistent sensor coverage**. While satellite data provides valuable post-event analysis, traditional systems underutilize **freely available vegetation indices** such as **NDVI (Normalized Difference Vegetation Index)**, which can indicate vegetation health and dryness—key factors that influence fire susceptibility. Furthermore, the lack of integration between **climatic factors** (temperature, humidity, wind, and precipitation) and vegetation dynamics reduces the predictive accuracy of these systems.

Another major limitation of existing systems is the **absence of class balancing** and **spatiotemporal analysis** in model design. Since forest fires are rare events compared to non-fire days, most traditional classifiers tend to be biased toward the majority class, resulting in **low sensitivity** to actual fire occurrences. These reactive, data-limited, and unbalanced models restrict the ability to predict fires with measurable lead times.

Hence, the current forest fire monitoring systems are primarily **detection-based**, lacking **proactive prediction frameworks** that combine **remote sensing**, **synthetic weather data**, and **machine learning algorithms**. This creates an urgent need for an intelligent, region-specific, and **data-driven predictive model**—such as **NallaFireNet**—that can generate early warnings and support sustainable forest management in vulnerable ecosystems.

3.1.1 DISADVANTAGES OF EXISTING SYSTEM

- **Reactive Detection Rather Than Prediction**

  • Current systems such as MODIS and VIIRS detect fires only after ignition, providing limited lead time for preventive action or evacuation.

- **Manual and Time-Consuming Monitoring**

  • Dependence on human surveillance, patrolling, and field reporting delays fire detection and increases response time in vast forest areas like Nallamala.

- **Limited Accessibility in Remote Regions**

  • Dense and inaccessible forest terrains hinder quick detection, making it difficult for forest officials to monitor fire-prone zones efficiently.

- **Underutilization of Remote Sensing Indices**

  • Existing frameworks rarely integrate vegetation indices such as NDVI, which are critical for understanding vegetation dryness and early fire risk.

- **Lack of Integration with Meteorological Data**

  • Traditional systems fail to combine climate factors like temperature, humidity, and precipitation, reducing the predictive accuracy of fire risk assessment.

- **Imbalanced and Incomplete Datasets**

  • Fire event data are highly imbalanced, with far fewer fire days than non-fire days, causing conventional models to misclassify rare fire instances.

- **Low Accuracy and High False Negatives**

  • Because of limited spatiotemporal analysis and insufficient data preprocessing, existing systems often miss true fire events, leading to delayed interventions.

- **Inability to Support Early Decision-Making**

  • Reactive systems offer limited support for preventive forest management, resulting in ecological damage, biodiversity loss, and economic impact.

- **Lack of Automation and Real-Time Alerts**

  • Existing frameworks do not provide automated early warning or visualization dashboards for authorities, making decision-making slow and inefficient

3.2 PROPOSED SYSTEM

The proposed system, **NallaFireNet**, introduces an **AI-driven predictive framework** for early detection and prevention of forest fires in the **Nallamala Forest region**. Unlike traditional reactive systems that detect fires after ignition, this model proactively forecasts potential fire occurrences by integrating **satellite-based vegetation indices** and **synthetic meteorological data**. The system leverages **machine learning and ensemble-based classification algorithms** to analyze spatiotemporal environmental patterns, enabling timely alerts and informed decision-making for forest management authorities.

NallaFireNet combines **NDVI (Normalized Difference Vegetation Index)** data from MODIS with synthetic weather parameters such as temperature, humidity, solar radiation, precipitation, and wind speed obtained from **NASA POWER datasets**. These data sources provide a comprehensive view of vegetation health and climatic conditions influencing fire risk. Advanced preprocessing techniques such as **temporal alignment**, **missing value imputation**, **data normalization**, and **lag feature generation** ensure clean and reliable datasets for model training. To overcome class imbalance—where fire events are rare compared to non-fire days—the system uses **SMOTE (Synthetic Minority Oversampling Technique)** to rebalance the dataset, improving model sensitivity and recall for rare fire instances.

The predictive engine utilizes an **ensemble of LightGBM, XGBoost, Random Forest, and Gradient Boosting models** within a **soft voting mechanism** to enhance accuracy and generalization. This ensemble structure combines the strengths of individual models—speed, interpretability, and robustness—to deliver a stable, high-performing predictive system. The model achieves strong performance across multiple evaluation metrics such as **Accuracy (91.46%)**, **Recall (69%)**, and **ROC-AUC (0.87)**, demonstrating its effectiveness in identifying potential fire-prone conditions.

The system's modular design allows for **real-time data integration and visualization**, supporting early alert generation and forest fire risk mapping. A user-friendly dashboard can display fire-prone zones, NDVI variations, temperature trends, and model predictions, enabling forest departments to plan **preventive measures**, allocate resources efficiently, and minimize ecological and economic damage

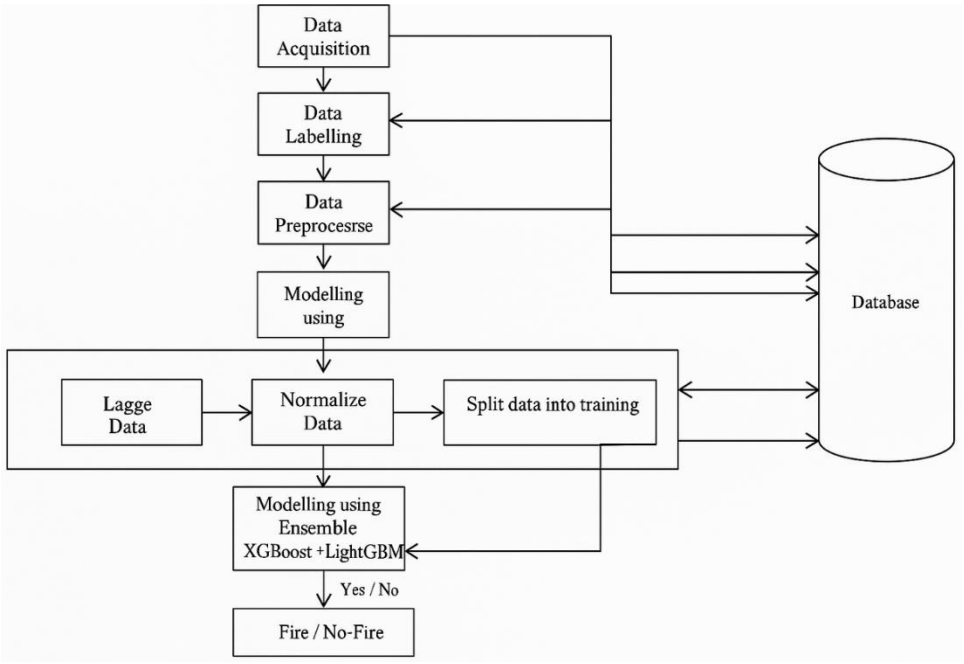


Fig. 3.2 Proposed Method

The proposed NallaFireNet system uses NDVI satellite data and synthetic weather parameters like temperature and humidity to predict forest fires in the Nallamala region. The dataset is divided into training and validation sets to build and test ensemble models such as XGBoost and LightGBM. Using SMOTE to balance rare fire events, the models learn patterns that indicate fire-prone conditions. Once trained, the system analyzes new environmental data to predict fire risks and visualize results through maps and trend charts. This automated approach enables early detection, improving preparedness and supporting timely decision-making in forest management

3.2.1 ADVANTAGES OF OVER EXISTING SYSTEM:

1. **Proactive Prediction:** Detects potential fire occurrences before ignition, allowing preventive action and resource mobilization.
2. **Data Integration:** Combines NDVI and synthetic meteorological data for a more comprehensive fire risk assessment.
3. **Improved Accuracy:** Ensemble learning enhances classification performance and minimizes false negatives.
4. **Balanced Learning:** SMOTE resolves dataset imbalance, improving model sensitivity to rare fire events.
5. **Automation:** Provides automated preprocessing, prediction, and visualization without manual intervention.
6. **Scalability:** The framework can be adapted to other wildfire-prone regions such as the Western Ghats or Himalayan ranges.
7. **Interpretability:** Identifies key environmental factors (e.g., temperature, humidity) influencing fire occurrence, aiding data-driven decision-making.
8. **Real-Time Monitoring:** Supports integration with APIs and dashboards for continuous updates and early warnings.

3.3 FEASIBILITY STUDY:

In the context of the **NallaFireNet** system, feasibility analysis ensures that the proposed forest fire prediction framework can be effectively developed, implemented, and maintained. The study evaluates the system's **technical, operational, and economic feasibility** to confirm its practicality and long-term sustainability. Technically, the system uses readily available tools such as Python, machine learning libraries, and satellite datasets, making implementation achievable. Operationally, it provides automated fire risk analysis and early warnings, supporting forest authorities in decision-making. Economically, it reduces manual monitoring costs and minimizes ecological losses, proving both cost-effective and efficient for large-scale deployment.

3.3.1 TECHNICAL FEASIBILITY:

1. Technology Stack

The system utilizes **LightGBM**, **XGBoost**, **Random Forest**, and **Gradient Boosting** algorithms within an **ensemble machine learning framework** for accurate forest fire prediction. It also employs **Python**, **Pandas**, **NumPy**, and **Scikit-learn** for data processing and model development, with visualization supported through **Matplotlib** and **Seaborn**.

2. Hardware and Software Requirements

A standard computing system with a **multi-core CPU** or **GPU support** is recommended for efficient model training and large dataset handling. Open-source tools such as **Python**, **Jupyter Notebook**, **LightGBM**, and **XGBoost** ensure high flexibility and easy deployment across different platforms.

3. Data Availability and Processing

The model is trained using **MODIS NDVI** satellite data and **synthetic weather datasets** from **NASA POWER** (2012–2025). Preprocessing steps include **data cleaning**, **missing value imputation**, **normalization**, and **temporal lag feature generation** to enhance data quality and model performance.

4. System Scalability and Performance

The system is highly **scalable** and can be extended to other wildfire-prone regions. It supports **real-time data integration** and cloud-based deployment for continuous monitoring and prediction. The ensemble model ensures strong **accuracy**, **recall**, and **stability**, enabling reliable early warning and effective forest fire management.

3.3.2 Economic Feasibility

1. Development Costs

Key expenses include hardware setup with **GPU-enabled systems** ($1500–$4000) for model training, **cloud computing services** ($100–$300/month) for large-scale data processing, and minor costs associated with **data preprocessing and model integration**. Since **Python**, **LightGBM**, and **XGBoost** are open-source, software expenses are minimal. Freely available datasets such as **MODIS NDVI** and **NASA POWER** significantly reduce data acquisition costs.

2. Implementation and Maintenance Costs

Implementation requires **cloud hosting** or **server infrastructure** ($50–$150/month) for real-time data processing and visualization dashboards. **Model retraining** and updates may cost between **$300–$1000 per year**, depending on data growth and accuracy requirements. Maintenance involves occasional tuning and integration support, ensuring cost efficiency over time.

3. Revenue and Cost Recovery

Potential revenue generation can arise through **collaborations with forest departments**, **environmental agencies**, and **research institutions**. Licensing the system to **disaster management authorities** or offering **subscription-based access** to forest monitoring platforms can further support cost recovery.

4. Cost-Benefit Analysis

The system's benefits far outweigh its costs by providing **early forest fire prediction**, reducing **ecological and economic losses**, and minimizing the need for manual surveillance. The automation and scalability of NallaFireNet make it a **cost-effective, long-term solution** for proactive forest management and environmental protection.

3.4 USING FIRENET MODEL:

The **NallaFireNet model** is a machine learning-based predictive framework designed to forecast potential forest fire occurrences using multi-source environmental data. It leverages **remote sensing imagery** and **synthetic meteorological data** to identify patterns associated with vegetation stress and climatic conditions that contribute to fire ignition. Unlike traditional reactive detection systems, NallaFireNet focuses on **early fire prediction** using ensemble learning techniques that combine both spatial and temporal data.

To develop the model, datasets from **MODIS NDVI (Normalized Difference Vegetation Index)** and **NASA POWER synthetic weather data** are preprocessed and aligned by date. Each record includes vegetation indices, temperature, humidity, solar radiation, wind speed, and precipitation. Temporal lag features are created to capture short-term environmental variations, while missing values are handled using **linear interpolation**. Data normalization ensures uniform scaling across features, improving model convergence during training.

The dataset is divided into **training and validation subsets**, where ensemble learning algorithms such as **XGBoost**, **LightGBM**, **Random Forest**, and **Gradient Boosting** are trained to classify each day as either "fire" or "non-fire." Since the dataset is highly imbalanced, with very few fire days compared to non-fire days, the **Synthetic Minority Oversampling Technique (SMOTE)** is applied to balance the class distribution.

The model's training objective minimizes a combination of **classification loss** and **prediction error**. The classification component is based on the **binary cross-entropy loss function**:

Model optimization and feature selection are carried out using **gradient-based learning** to identify the most influential features affecting fire risk. Among these, **temperature**, **humidity**, and **NDVI** have shown the highest importance, aligning with ecological drivers of wildfire occurrence.

Once trained, the ensemble model is deployed to predict fire risks based on incoming NDVI and meteorological data. The final prediction confidence score (SSS) is derived from the combined probabilities of individual models in a soft voting mechanism:

S=PXGBoost+PLightGBM+PRF+PGB4S = \frac{P_{XGBoost} + P_{LightGBM} + P_{RF} + P_{GB}}{4}S=4PXGBoost+PLightGBM+PRF+PGB

where each PPP represents the probability of a fire event predicted by an individual model.

The system can be integrated with a **Flask-based API** for real-time inference, allowing forest departments to upload current environmental data and receive instant fire risk predictions. Visual outputs such as **NDVI heatmaps**, **monthly fire occurrence graphs**, and **probability-based risk maps** assist authorities in identifying high-risk zones.

By combining **machine learning, remote sensing, and synthetic climate modeling**, the NallaFireNet framework provides an automated, scalable, and explainable solution for **early forest fire prediction**, reducing reliance on manual observation and enhancing proactive forest management.

4.SYSTEM REQUIREMENTS

The **system requirements** define the essential software and hardware components needed for the development, training, and deployment of the **NallaFireNet** forest fire prediction system. These specifications ensure that the project operates efficiently while managing large datasets, performing machine learning model training, and supporting real-time prediction and visualization. Proper configuration of both hardware and software resources guarantees optimal system performance, scalability, and reliability during implementation.

4.1 SOFTWARE REQUIREMENTS:

The proposed **NallaFireNet** system relies on various software components for data preprocessing, model training, visualization, and web-based deployment.

- Browser : Any Latest browser like Chrome
- Operating System : Windows 10
- Language : Python 3.8+
- Platform : Visual Studio
- Libraries and Framework:
    - **Machine Learning:** LightGBM, XGBoost, Scikit-learn
    - **Data Processing:** NumPy, Pandas, Matplotlib, Scikit-learn
    - **Visualization and API Integration:** Flask, Plotly
- Deployment Tools: Flask-based web application

4.1.1 IMPLEMENTATION OF MACHINE LEARNING MODEL FOR FOREST FIRE PREDICTION

The implementation phase involves developing a predictive machine learning framework to forecast forest fire occurrences in the Nallamala Forest region using remote sensing and climatic data. The process integrates vegetation indices (NDVI) with meteorological variables such as temperature, humidity, and precipitation collected from NASA POWER datasets. The following Python libraries were used in implementation:

1. NumPy:

NumPy is used for efficient numerical computation and array operations during data preprocessing. It supports large-scale matrix manipulations, making it ideal for handling the time-series data obtained from MODIS and NASA datasets. Its vectorized operations improve the computational speed of model training and feature engineering.

2. Pandas:

Pandas plays a crucial role in organizing, cleaning, and merging large NDVI and weather datasets. It simplifies handling missing values, date-time alignment, and creation of lag features. Using Pandas DataFrames, the merged datasets are prepared for further modeling and statistical analysis.

3. Matplotlib:

Matplotlib is used for data visualization to analyze NDVI patterns, temperature trends, and seasonal variations. It helps visualize model results through confusion matrices, ROC curves, and feature importance plots, allowing better interpretation of the model's performance.

4. Scikit-learn:

Scikit-learn provides tools for feature selection, model evaluation, and performance analysis. It supports the train-test split, SMOTE resampling for class balancing, and evaluation metrics such as accuracy, recall, precision, and F1-score to validate model performance.

5. LightGBM:

LightGBM, a gradient boosting algorithm, is used to identify top predictive features and train the model efficiently on large datasets. Its high-speed training capability and ability to handle complex, non-linear relationships make it ideal for predicting fire-prone conditions.

6. XGBoost:

XGBoost complements LightGBM in the ensemble framework. It uses gradient boosting to reduce overfitting and improve classification accuracy. The combination of LightGBM and XGBoost enhances model stability and ensures robust performance across imbalanced datasets.

7. Imbalanced-learn (SMOTE):

The Synthetic Minority Oversampling Technique (SMOTE) from the Imbalanced-learn library is used to balance the fire and non-fire data classes. This ensures that the model learns effectively from both classes, improving recall and minimizing bias toward majority samples.

4.2 REQUIREMENT ANALYSIS:

The forest fire prediction system is designed to forecast fire occurrences in the Nallamala Forest region using machine learning and remote sensing data. The requirement analysis defines both functional and non-functional aspects essential for developing a reliable and scalable predictive framework.

Functional Requirements:

• **Data Acquisition and Integration:** Collect NDVI data from MODIS satellite archives and meteorological parameters (temperature, humidity, precipitation) from NASA POWER datasets.

• **Data Preprocessing and Feature Engineering:** Clean and normalize the datasets, create lag features, and merge NDVI with weather parameters for accurate temporal

analysis.

• **Model Training and Prediction:** Implement an ensemble learning model combining LightGBM and XGBoost algorithms to classify fire and non-fire events.

• **Alert Generation and Visualization:** Generate early fire warnings and visualize results through ROC curves, confusion matrices, and feature importance charts.

Non-Functional Requirements:

• **Performance & Scalability:** The system should efficiently process multi-year satellite data (2012–2025) and scale for real-time fire prediction.

• **Reliability & Accuracy:** Ensure high model performance with optimal accuracy and recall using SMOTE-balanced datasets and ensemble classifiers.

• **Security & Data Integrity:** Protect sensitive geospatial and environmental data during processing and storage.

• **Usability & Accessibility:** Provide an intuitive interface for environmental analysts and forest officers for easy interpretation of predictions and visual outputs.

4.3 HARDWARE REQUIREMENTS:

The hardware configuration ensures efficient execution of data preprocessing, feature engineering, and model training for large-scale forest fire prediction using machine learning algorithms.

- ❖ **System Type:** Intel® Core™ i5 / i7 Processor (2.0 GHz or higher)
- ❖ **Cache Memory:** 6 MB
- ❖ **RAM:** 8 GB (minimum) or 16 GB (recommended)
- ❖ **Hard Disk:** 1 TB
- ❖ **Bus Speed:** 5 GT/s DMI
- ❖ **Number of Cores:** 4
- ❖ **Operating System:** Windows 10 / 11 (64-bit)

4.4 SOFTWARE

The development stack includes Python and Flask, providing a robust and scalable environment for both deep learning and web-based applications.

- **Python:** The core language used for deep learning, image processing, and backend development.
- **Flask:** A lightweight framework for deploying the trained deep learning model as a web application.
- **Google Colab Pro:** Used for high-performance training of deep learning models with GPU support

4.5 SOFTWARE DESCRIPTION:

The software components are developed to enable efficient forest fire prediction using machine learning and remote sensing data. The system integrates data preprocessing, model training, and visualization modules for end-to-end analysis and prediction.

Machine Learning Model:

Implemented using **LightGBM** and **XGBoost** within an ensemble framework to classify fire and non-fire events based on NDVI and synthetic weather data.

Data Processing and Preprocessing:

Utilizes **Pandas** and **NumPy** for cleaning, normalization, and merging of MODIS NDVI and NASA POWER meteorological datasets. Lag features are generated to capture temporal variations in temperature and humidity.

Visualization Tools:

Employs **Matplotlib** and **Seaborn** for plotting correlation matrices, ROC curves, feature importance charts, and confusion matrices to evaluate model performance.

Model Development Environment:

Developed and executed on **Google Colab**, leveraging **Scikit-learn**, **Imbalanced-**

learn (SMOTE), and **TensorFlow** libraries for advanced computation and class balancing.

Deployment Capability:

The trained model can be integrated with a **Flask-based API** or **web dashboard** for real-time visualization and alert generation, enabling forest officials to monitor fire-prone regions proactively

4.5.1 MACHINE LEARNING

Machine Learning (ML), a subset of Artificial Intelligence (AI), empowers systems to learn from data patterns and make intelligent predictions without explicit programming. It plays a crucial role in analyzing large-scale environmental datasets, enabling early detection and prediction of forest fires.

In this project, ML techniques are utilized to analyze **Normalized Difference Vegetation Index (NDVI)** and **synthetic weather data** to forecast fire occurrences in the Nallamala Forest. The system leverages **supervised learning algorithms** to train on historical data, identifying relationships between vegetation health, temperature, humidity, and fire incidents.

The core components of the ML framework include **data preprocessing**, **feature engineering**, **model training**, and **evaluation**. Various ensemble algorithms, such as **LightGBM** and **XGBoost**, are employed to enhance prediction accuracy by combining multiple weak learners into a strong predictive model.

Feature importance analysis helps identify the most influential parameters contributing to fire events, while evaluation metrics such as **accuracy, precision, recall, and F1-score** measure model performance. By integrating machine learning with remote sensing data, the proposed framework provides a robust solution for early fire prediction, aiding proactive forest management and minimizing ecological damage.

4.5.2 MACHINE LEARNING METHODS:

The proposed system employs **ensemble-based machine learning algorithms**, specifically **Light Gradient Boosting Machine (LightGBM)** and **Extreme Gradient Boosting (XGBoost)**, for accurate forest fire prediction. These models are well-suited for handling large and complex environmental datasets, as they efficiently capture non-linear relationships between climatic parameters and fire occurrences.

LightGBM is a gradient boosting framework that uses tree-based learning algorithms optimized for speed and performance. It handles large-scale data efficiently and supports parallel learning, making it ideal for analyzing multi-year weather and NDVI datasets. LightGBM also ranks feature importance, helping identify the most influential environmental variables contributing to forest fires.

XGBoost complements LightGBM by enhancing model robustness and reducing overfitting through regularization and optimized boosting techniques. It builds multiple weak learners (decision trees) iteratively to minimize prediction error, leading to improved generalization and accuracy.

In this project, both models are combined in a **voting ensemble** approach to improve prediction performance. The ensemble framework leverages the strengths of each model, ensuring higher stability and reliability, particularly for imbalanced datasets. The integration of SMOTE for class balancing and feature selection using LightGBM further refines the model's predictive capability.

This hybrid machine learning approach provides a scalable, efficient, and accurate framework for **early forest fire prediction**, making it suitable for real-time monitoring and decision support in forest management systems.

4.5.2.1 INTRODUCTION ENSEMBLE LEARNING:

A typical machine learning model learns patterns from data using a single algorithm, but ensemble learning combines multiple models to achieve higher accuracy, stability, and generalization. Ensemble learning draws inspiration from the idea that combining multiple “weak learners” can produce a “strong learner” capable of making more reliable predictions.

In the proposed system, ensemble learning integrates **Light Gradient Boosting Machine (LightGBM)** and **Extreme Gradient Boosting (XGBoost)** algorithms to predict forest fire occurrences in the Nallamala Forest region. Both models use tree-based boosting techniques, where each new model corrects the errors made by previous ones.

The ensemble technique combines the predictions of these two models through a **voting mechanism**, where both algorithms contribute to the final decision. This approach minimizes bias and variance, resulting in more accurate and robust fire event classification.

Ensemble learning is particularly effective in this context because of the **imbalanced nature of forest fire data**. It leverages the strengths of each base learner to handle diverse feature distributions derived from NDVI and synthetic weather datasets.

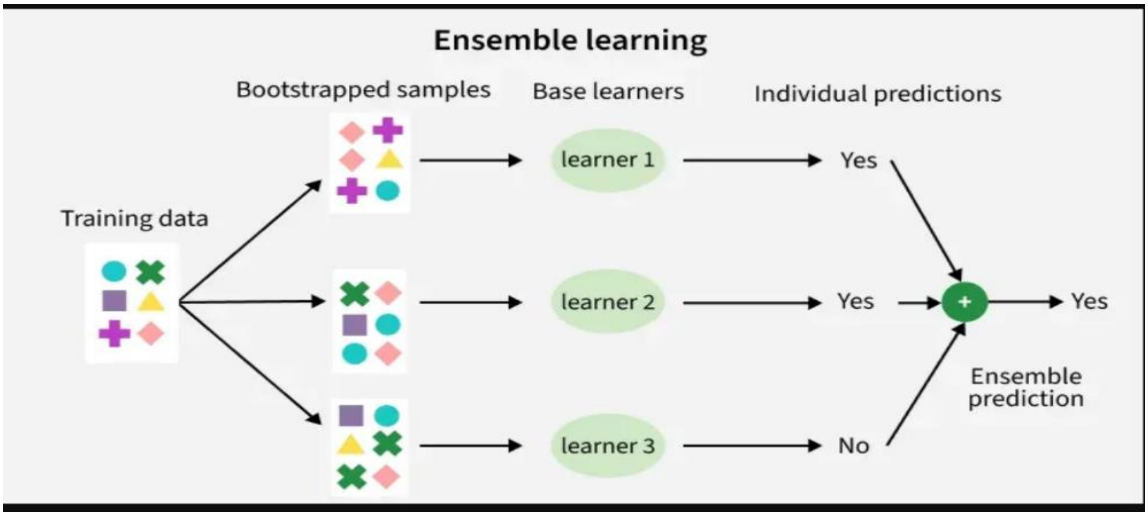


FIG 4.5.2.1:classification by Ensemble Learning

4.5.2.2 COMPONENTS OF ENSEMBLE MODEL:

The ensemble learning framework for forest fire prediction integrates multiple stages of data handling, feature extraction, and classification to achieve high accuracy and reliability. The main components of the proposed model include:

- **Data Preprocessing Layer**
- **Feature Engineering Layer**
- **Model Training Layer (LightGBM & XGBoost)**
- **Voting Mechanism Layer**
- **Evaluation and Visualization Layer**

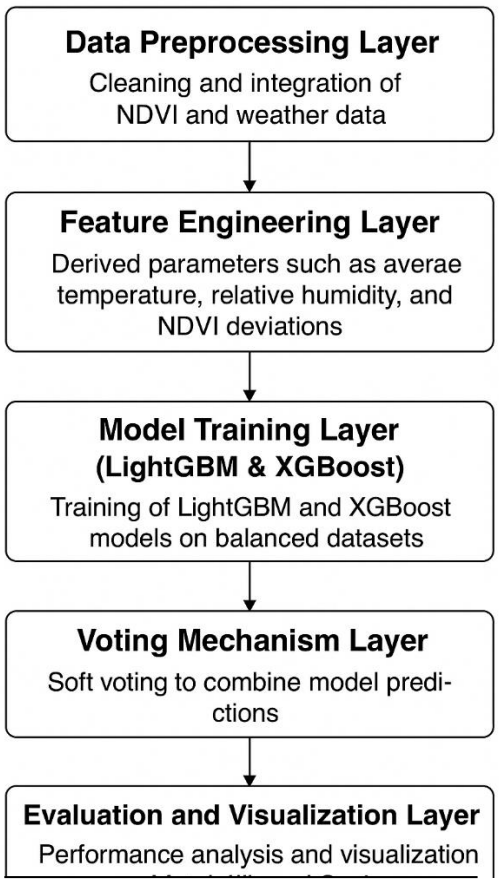


Fig4.5.2.2:Components of Ensemble Model

Data Preprocessing Layer:

This layer involves cleaning and integrating NDVI and weather data from multiple-sources. Missing values are handled, and temporal alignment is ensured. Normalization and scaling are applied to maintain data consistency.Lagfeatures are also generated to represent vegetation and climatic patterns over time.

Feature Engineering Layer:

Feature engineering extracts meaningful information from raw datasets. Derived parameters such as average temperature, relative humidity, and NDVI deviations are computed. These features enhance the model's ability to detect patterns associated with fire events.

Model Training Layer (LightGBM & XGBoost):

This layer represents the core learning process of the system. LightGBM builds gradient-boosted trees using histogram-based methods, while XGBoost constructs sequential trees through boosted gradient descent. Both algorithms are trained on balanced datasets (via SMOTE), and their predictions are stored for ensemble voting.

Voting Mechanism Layer:

The ensemble system combines the outputs of both models through a soft voting technique. Each model's prediction probability contributes to the final decision, improving robustness and reducing overfitting. This mechanism ensures stability and enhances classification accuracy across different environmental conditions.

Evaluation and Visualization Layer:

In this layer, model performance is analyzed using metrics such as accuracy, recall, precision, and F1-score. Visualization tools like confusion matrices, ROC curves, and feature importance graphs are generated using Matplotlib and Seaborn to interpret model efficiency.

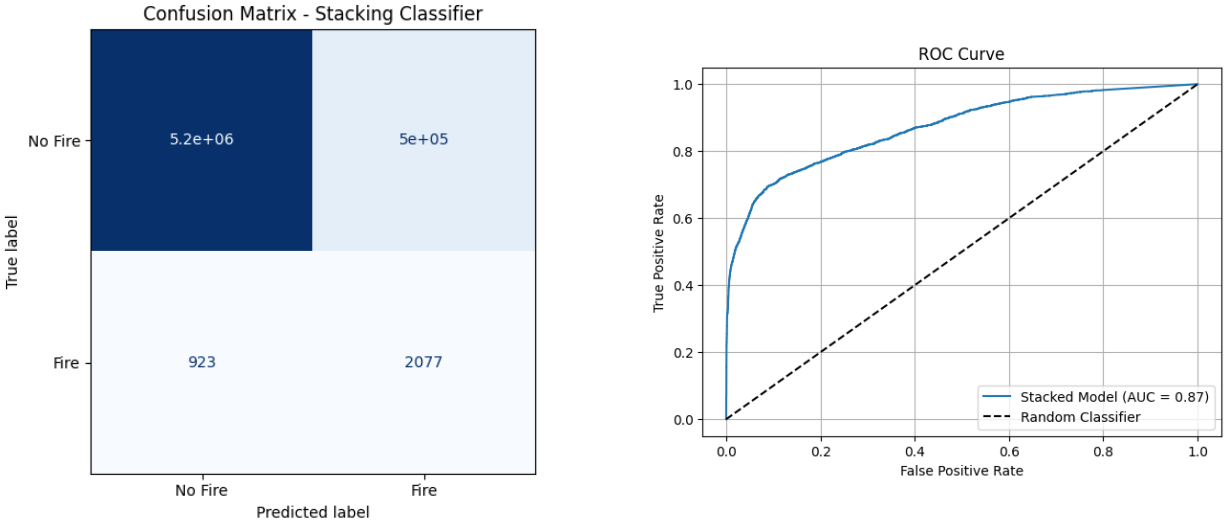


Fig 4.5.2.2.1: Model Evaluation and Visualization Out

4.5.3 APPLICATIONS OF MACHINE LEARNING IN FOREST FIRE PREDICTION

1. **Early Fire Detection:** Predicts potential forest fire events using NDVI and weather data to enable timely preventive actions.
2. **Environmental Monitoring:** Continuously tracks vegetation health and climatic variations to assess ecosystem stability.
3. **Disaster Management:** Supports authorities in planning evacuation routes and deploying fire control resources effectively.
4. **Climate Change Analysis:** Helps understand the correlation between rising temperatures, humidity shifts, and fire occurrences.

5. **Early Fire Detection:** Predicts potential forest fire events using NDVI and weather data to enable timely preventive actions.
6. **Environmental Monitoring:** Continuously tracks vegetation health and climatic variations to assess ecosystem stability.
7. **Disaster Management:** Supports authorities in planning evacuation routes and deploying fire control resources effectively.
8. **Climate Change Analysis:** Helps understand the correlation between rising temperatures, humidity shifts, and fire occurrences.
9. **Risk Mapping:** Generates fire susceptibility maps to identify and monitor high-risk forest zones.
10. **Data-Driven Decision Making:** Assists forest departments in resource allocation and policy formulation based on predictive analytics.
11. **Remote Sensing Integration:** Utilizes satellite data for real-time observation of vegetation and atmospheric conditions.
12. **Automation in Forest Surveillance:** Enhances traditional monitoring systems through AI-based alert mechanisms.
13. **Wildlife Protection:** Prevents habitat destruction by predicting fires before they spread to animal zones.
14. **Sustainable Forest Management:** Contributes to preserving biodiversity and reducing carbon emissions through early warning systems.

4.5.4 IMPORTANCE OF MACHINE LEARNING IN FOREST FIRE PREDICTION:

Machine Learning plays a crucial role in enhancing environmental intelligence by enabling automated prediction and early detection of forest fires. Its ability to analyze vast datasets, identify complex patterns, and make accurate predictions makes it indispensable in disaster prevention and forest management.

The proposed system leverages **ensemble machine learning models** such as **LightGBM** and **XGBoost**, which can efficiently process large volumes of NDVI and meteorological data. These models are capable of detecting subtle correlations between vegetation health, temperature, humidity, and fire occurrences that traditional statistical methods often miss.

Machine Learning's importance also lies in its **automation of feature extraction and decision-making**, reducing human intervention and improving prediction reliability. With the integration of **synthetic data generation** and **SMOTE for class balancing**, the system ensures accurate learning even in cases of data imbalance — a common challenge in real-world fire datasets.

Furthermore, ML-based systems provide **real-time insights and adaptive learning capabilities**, allowing continuous model updates as new data becomes available. This adaptability helps authorities respond faster and plan preventive measures effectively.

The scalability and flexibility of ML algorithms also make them suitable for integrating with **IoT devices, drones, and satellite-based monitoring systems**, paving the way for intelligent forest management solutions.

In essence, the importance of Machine Learning in forest fire prediction lies in its ability to handle large-scale environmental data, automate complex analyses, ensure real-time forecasting, and support data-driven decision-making for sustainable ecosystem protection.

5.SYSTEM DESIGN

The image represents a flowchart outlining the process of **forest fire prediction using an ensemble machine learning model**, specifically **LightGBM** and **XGBoost**. As shown in **Fig. 5**, the flowchart visually explains the sequential stages involved in processing environmental data and predicting the likelihood of fire occurrence. The system follows a structured pipeline, beginning with **data acquisition and preprocessing** and ending with **final prediction and visualization**.

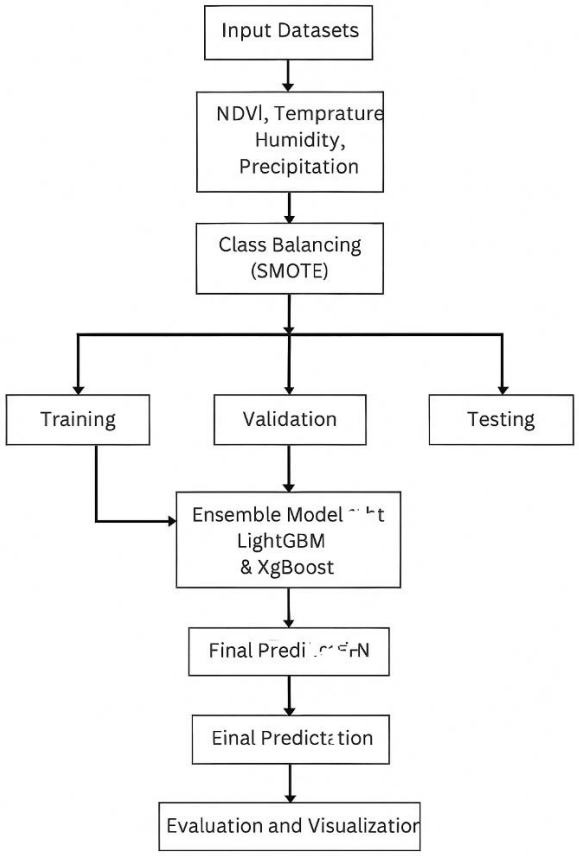


Fig. 5.1 System architelcture of Forest Fire

Fig5.1 System Design

The process starts with the **input datasets**, which include satellite-derived NDVI values and synthetic weather data such as temperature, humidity, and precipitation. These datasets are first passed through a **data preprocessing layer**, where cleaning,

normalization, and feature extraction are performed. Missing values are handled, and lag features are generated to capture temporal dependencies.

Next, the preprocessed data is subjected to **class balancing** using the **Synthetic Minority Oversampling Technique (SMOTE)** to address the issue of imbalanced fire and non-fire event samples. The balanced dataset is then divided into **training and testing subsets** to ensure fair model evaluation.

The **ensemble model layer** combines two powerful gradient boosting algorithms — **LightGBM** and **XGBoost**. Each model is trained individually to learn complex non-linear patterns from the input data. The final predictions from both models are integrated through a **voting mechanism**, ensuring high accuracy and robustness in classification.

The final stage involves **evaluation and visualization**, where the system generates performance metrics such as accuracy, precision, recall, and F1-score. Graphical outputs like confusion matrices, ROC curves, and feature importance plots are used to interpret model efficiency and feature influence.

Overall, this flowchart represents a comprehensive and efficient pipeline for **forest fire prediction using machine learning**. By leveraging the strengths of LightGBM and XGBoost, the system achieves high predictive accuracy and reliability, enabling early fire warnings and aiding forest management authorities in proactive disaster prevention.

5.1 SYSTEM ARCHITECTURE:

1. **User Interface (Front-End – Web or Dashboard Interface):**

   A web-based interface allows users such as **forest officers or environmental analysts** to view and monitor real-time forest fire predictions. The interface provides visual dashboards displaying temperature, humidity, NDVI patterns, and fire risk zones. Users can also upload new data files or trigger predictions for specific regions. The results are presented with **fire probability scores**, confidence levels, and corresponding alert levels (Low, Medium, or High).

2. **Backend (Machine Learning Server):**

   The backend system is developed using **Python** and powered by **Flask** for handling API requests and communication with the trained ensemble model. Input data (NDVI and weather parameters) are preprocessed — including normalization, feature scaling, and lag feature creation — before being fed to the **LightGBM–XGBoost ensemble model**. The backend processes the input, performs prediction, and returns the probability of fire occurrence in **JSON format**, which is then visualized on the user dashboard.

3. **Data Processing and Model Layer:**

   This layer includes the core machine learning workflow — from **data cleaning and feature selection** to **model training and evaluation**. It utilizes libraries like **Pandas, NumPy, Scikit-learn, Imbalanced-learn (SMOTE), and TensorFlow** to handle data operations and optimize classification. The ensemble model ensures robust performance through combined predictions and voting mechanisms.

4. **Visualization and Alert System:**

   The final stage converts model outputs into interpretable visualizations such as **heatmaps, ROC curves, and fire-prone zone maps**. Automated alerts can be generated based on prediction confidence, notifying authorities of potential high-risk areas for preventive action.

5.1.1 DataSet

https://drive.google.com/drive/folders/1SLwdLpQZJAQgsIbkG5oeggeErkMsapX8

The dataset used for forest fire prediction integrates **remote sensing and meteorological data** spanning the years **2012 to 2025**. The data is primarily collected from **NASA's POWER Regional Datasets** and **MODIS satellite archives**, ensuring both temporal depth and spatial diversity. The dataset includes multiple environmental parameters such as **temperature (T2M)**, **relative humidity (RH2M)**, and **precipitation (PRECTOTCORR)**, along with vegetation indices like **NDVI (Normalized Difference Vegetation Index)** extracted from **MODIS HDF files**.

Ground-truth fire occurrences are obtained from **MODIS and VIIRS fire archive datasets**, which provide accurate fire event locations and dates. These records serve as the target labels for supervised learning. The data preprocessing stage involves merging, cleaning, normalization, and feature engineering, including **lag features** and **seasonal encoding** to capture temporal dependencies.

To address class imbalance between fire and non-fire samples, the **Synthetic Minority Oversampling Technique (SMOTE)** is applied, ensuring balanced class distribution for effective model training. This comprehensive dataset supports the development of a robust predictive model capable of identifying early signs of forest fire events in the **Nallamala Forest region**.

Dataset Categories:

The dataset used for this project encompasses a diverse range of environmental and vegetation parameters categorized into **multiple data types**, as shown in **Fig. 5.1**, including **Temperature (T2M)**, **Relative Humidity (RH2M)**, **Precipitation (PRECTOTCORR)**, and **Normalized Difference Vegetation Index (NDVI)**. These parameters collectively represent the climatic and vegetative conditions of the **Nallamala Forest region** from **2012 to 2025**.

Each category contributes unique information crucial for predicting fire events.

- **Temperature Data:** Captures the surface air temperature variations influencing fire ignition potential.
- **Humidity Data:** Represents atmospheric moisture content, inversely correlated with fire risk.
- **Precipitation Data:** Reflects rainfall intensity and frequency, a critical factor in fire suppression.
- **NDVI Data:** Measures vegetation greenness and health, serving as a key indicator for fuel availability.

This comprehensive dataset enables the development of robust machine learning models capable of correlating vegetation dynamics and weather fluctuations with fire occurrences. The diversity and temporal depth of these categories ensure accurate prediction and effective early warning generation for forest fire management.

Graphical Representation of Dataset

As shown in **Fig. 5.1.1**, the graphical representation provides a visual overview of the **environmental and vegetation datasets** used for forest fire prediction. The graphs illustrate the variations in **temperature, humidity, precipitation, and NDVI** across different time periods from **2012 to 2025**.

This visualization helps in understanding how climatic fluctuations and vegetation health patterns influence the likelihood of fire occurrences in the **Nallamala Forest region**. The graphical plots also highlight the seasonal trends, where higher temperatures and lower humidity correspond to increased fire risks.a

Such visual representations enable better **data interpretation, correlation analysis, and model optimization**, serving as a foundation for building accurate and reliable fire prediction systems.

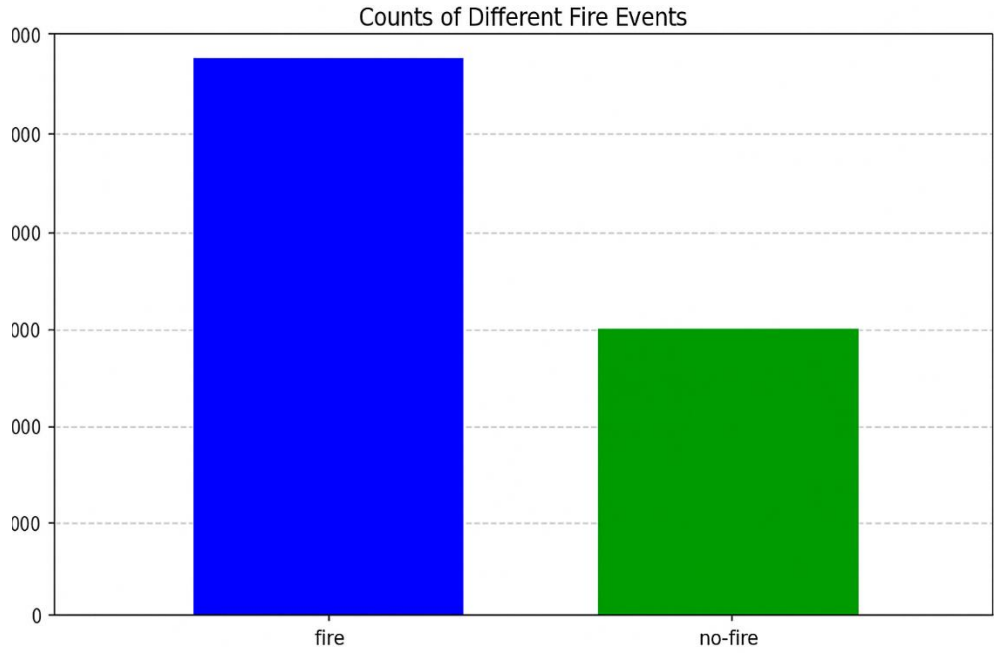


Fig5.1.1 Graphical representation of Dataset

5.1.2 DATA PREPROCESSING :

Before feeding the data into the proposed machine learning framework, several preprocessing steps were carried out to ensure data consistency, quality, and accuracy. The raw datasets included **MODIS NDVI** data representing vegetation health and **synthetic weather variables** such as temperature, humidity, solar radiation, and precipitation obtained from **NASA POWER** for the years **2012–2025**.

First, **temporal alignment** was performed to synchronize daily NDVI and weather data records. **Missing values** were handled using **linear interpolation and forward-fill techniques**, ensuring a continuous and reliable time series. Each feature was then **normalized using min–max scaling** to bring all input variables into a uniform range between 0 and 1, which stabilized the model's training process.

To improve feature richness, **lag features** for temperature and humidity were engineered for up to seven days, capturing short-term climatic dependencies relevant to fire occurrences. A **correlation heatmap** was also analyzed to identify multicollinearity and prioritize highly predictive features.

Finally, since the dataset exhibited a strong **class imbalance** (very few fire days compared to non-fire days), the **Synthetic Minority Oversampling Technique (SMOTE)** was applied to balance the fire and non-fire samples. This ensured better sensitivity of the model toward rare fire events while preventing bias toward the majority class.

This preprocessing pipeline enhanced the quality of the input data, minimized noise, and prepared an optimized dataset for training ensemble models like **XGBoost** and **LightGBM** to achieve accurate and interpretable forest fire predictions in the **Nallamala Forest region**.

5.1.3 DATA ENHANCEMENT:

Data enhancement is an essential preprocessing phase in the NallaFireNet framework, aimed at improving the quality, consistency, and predictive value of environmental datasets before model training. For forest fire prediction, it ensures that critical variables such as **NDVI (Normalized Difference Vegetation Index)**, **temperature**, **humidity**, **wind speed**, and **precipitation** are accurately represented and aligned across all time periods.

The enhancement process begins with **data cleaning**, where missing values are filled using **interpolation** and **forward-fill methods** to maintain temporal continuity. **Noise reduction techniques** are applied to smooth abrupt fluctuations in NDVI and weather signals caused by sensor or measurement errors. Afterward, **feature normalization** using **min–max scaling** standardizes all parameters into a uniform range, improving model convergence and stability during training.

To enhance the model's understanding of temporal and seasonal trends, **feature engineering** is performed by creating **lag variables**, **rolling averages**, and **seasonal encodings** that capture environmental dependencies influencing forest fire occurrences. Furthermore, to overcome the issue of **data imbalance** between fire and non-fire events, the **Synthetic Minority Oversampling Technique (SMOTE)** is employed to generate synthetic fire event samples, ensuring balanced class representation.

These enhancement strategies collectively transform the raw environmental and vegetation data into a refined, high-quality dataset that significantly improves the learning efficiency and predictive power of ensemble deep learning models such as **LightGBM** and **XGBoost**, enabling reliable and early detection of forest fire risks in the **Nallamala Forest region.**

5.1.4 Data Smoothing and Noise Reduction:

In the context of forest fire prediction, data smoothing plays a vital role in minimizing random fluctuations and enhancing the reliability of environmental and vegetation datasets. Since raw data from NASA's POWER and MODIS sources often contain inconsistencies due to atmospheric interference or sensor errors, smoothing helps reveal the underlying climatic trends that influence fire behavior.

The **Gaussian filter** is applied to key variables such as temperature, humidity, precipitation, and NDVI to eliminate short-term noise while preserving long-term trends. This is achieved by assigning higher weights to nearby data points and gradually lower weights to distant points, producing a smooth and continuous time-series representation.

Through this process, the dataset retains essential seasonal and temporal patterns while removing irregular disturbances, ensuring stability for downstream analysis. The refined data allows the **LightGBM–XGBoost ensemble framework** to better capture correlations between vegetation dynamics and meteorological variations that precede forest fires.

By selecting appropriate kernel parameters and standard deviation values, Gaussian smoothing effectively balances noise suppression with data fidelity—improving both the clarity and predictive accuracy of the model.

5.1.5 Spatial Filtering on Regions of Interest (ROI)

The **bilateral filtering technique** is applied in the preprocessing stage to refine spatial data such as NDVI and temperature maps, ensuring smoothness while retaining critical boundary information, as shown in **Fig. 5.1.2**. Unlike traditional linear filters, bilateral filtering accounts for both **spatial proximity** and **value similarity**, making it ideal for preserving environmental gradients and abrupt changes, such as forest edges or fire boundaries.

This dual-domain filtering approach enhances NDVI imagery by removing random satellite noise while retaining significant transitions between vegetation-rich and degraded areas. It combines a **spatial Gaussian kernel** that smooths neighboring pixels based on their distance and an **intensity Gaussian kernel** that restricts averaging to pixels with similar NDVI values.

When applied to **Regions of Interest (ROIs)** — areas with a high probability of fire occurrence — the bilateral filter ensures that relevant spatial features remain intact, as depicted in **Fig. 5.1.3**. By selectively applying the filter to these ROIs, noise is minimized without distorting fire-relevant zones. This localized filtering improves the quality of the dataset and enhances the accuracy of subsequent fire prediction by the **NallaFireNet ensemble model**.

Overall, bilateral filtering with ROI selection helps in **noise suppression, edge preservation, and clearer feature delineation**, providing a more accurate foundation for model training and spatial fire risk analysis

5.1.6 Segmentation

Segmentation is a critical step in the preprocessing phase of forest fire prediction, aimed atisolating regions with distinct environmental characteristics such as vegetation density, temperature, and humidity. Through segmentation, the system can effectively differentiate between forested areas, barren lands, and potential fire zones within satellite imagery. This process ensures that only relevant regions contribute to model training and prediction accuracy.

In this project, segmentation operates by separating the **foreground (fire-prone zones)** from the **background (safe regions)** using NDVI and thermal indices. The segmentation algorithm primarily relies on two principles: **discontinuity** and **similarity**. Discontinuity refers to abrupt changes in pixel intensity values—helpful in detecting fire-affected regions or vegetation boundaries. Similarity, on the other hand, groups pixels with comparable intensity or NDVI values, aiding in the identification of healthy vegetation clusters.

The segmentation process concludes once the **regions of interest (ROIs)**, such as vegetation loss or high-temperature zones, are isolated. Common methods like **threshold-based segmentation** and **edge-based detection** (e.g., gradient analysis) are used to delineate areas of concern. Thresholding helps convert continuous satellite data into binary maps, simplifying the identification of high-risk areas for fire ignition.

By implementing segmentation, the NallaFireNet framework enhances the clarity of input data, removes irrelevant noise, and improves model precision. This stage is vital for ensuring accurate and efficient detection of potential fire zones, forming the foundation for subsequent classification and prediction stages.

5.2 MODULES :

5.2.1 NEED OF DATA PREPROCESSING:

In Machine Learning–based forest fire prediction, achieving reliable and accurate results heavily depends on **effective data preprocessing**. The raw environmental data collected from multiple satellite and meteorological sources—such as NDVI, temperature, humidity, and precipitation—often contain missing values, inconsistencies, and variations in scale. These irregularities can severely affect the model's performance if not properly handled.

To ensure compatibility across the dataset, the data is first **cleaned, normalized, and formatted** into a unified structure. The preprocessing process involves merging datasets from NASA's POWER archives and MODIS NDVI files, followed by generating new **engineered features** such as lag values and seasonal indicators. These features help capture temporal dependencies crucial for understanding fire patterns.

Additionally, to resolve class imbalance between fire and non-fire samples, the **Synthetic Minority Oversampling Technique (SMOTE)** is applied, which generates synthetic data points for underrepresented fire events. This step ensures that the predictive model can learn effectively from both classes without bias.

Proper data preprocessing not only enhances the model's accuracy and robustness but also ensures smooth integration of multiple algorithms like **LightGBM** and **XGBoost** within the ensemble structure. This step is essential to achieve consistency, scalability, and high predictive efficiency across different fire prediction models.

5.2.2 ARCHITECTING MODULES:

The architecture module defines the **machine learning ensemble framework** used for forest fire prediction. Instead of a deep CNN, this system implements a combination of **LightGBM** and **XGBoost** models that work collaboratively to classify fire and non-fire events based on historical and remote sensing data.

Module Components:

- **Feature Extraction Layer:** Derives key features from NDVI, temperature, humidity, and precipitation datasets. Lag features and seasonal encodings are generated to represent temporal trends.
- **Data Balancing Layer:** Uses **SMOTE** to balance minority and majority classes, improving recall and stability in prediction.
- **Model Training Layer:** Trains two independent models — **LightGBM** and **XGBoost** — each optimized for gradient boosting. LightGBM ensures faster computation with large datasets, while XGBoost focuses on precision and robustness.
- **Ensemble Voting Layer:** Combines predictions from both models using a **voting classifier**, which averages their outputs to enhance accuracy and reduce overfitting.
- **Evaluation Layer:** Assesses performance using metrics such as **Accuracy, Precision, Recall, F1-Score, and AUC-ROC Curve**.

This modular architecture ensures scalability, adaptability, and high computational efficiency, making it well-suited for real-time forest fire monitoring and alert systems.

5.2.3 TESTING MODULE:

The **Testing Module** evaluates the performance and reliability of the trained ensemble machine learning model using real-world environmental datasets from the **Nallamala Forest region**. This phase ensures that the predictive framework performs accurately and consistently under different climatic conditions.

Testing Phases:

- **Unit Testing:** Each component of the pipeline—data preprocessing, feature extraction, and ensemble voting—is validated individually to ensure correct functionality and output consistency.
- **Performance Testing:** The model is tested on unseen environmental data (from 2024–2025) to assess prediction accuracy, recall, and computational efficiency.
- **Cross-Validation:** The dataset is divided into multiple folds to evaluate the model's generalization ability. This reduces overfitting and ensures balanced learning.
- **Evaluation Metrics:** The model's performance is assessed using **Accuracy**, **Precision**, **Recall**, **F1-Score**, and **Confusion Matrix**. Additional metrics such as **ROC-AUC Curve** and **Feature Importance Graphs** are also used to interpret model reliability.

This module ensures that the NallaFireNet system is **robust, efficient, and ready for operational deployment**, providing dependable fire predictions under varying forest and weather conditions.

5.2.4 User Interface (UI) Module

The **UI Module** serves as an interactive platform for environmental analysts and forest management authorities to visualize and interpret the system's fire predictions. The front-end is developed using **Flask**, which integrates seamlessly with the **Python-based backend and ensemble models (LightGBM and XGBoost)**.

Key Features of the UI:

- **Data Upload Feature:** Users can upload NDVI or weather datasets (in CSV format) for analysis.
- **Real-Time Predictions:** The system processes the uploaded data and provides instant **fire risk classifications** along with confidence scores.
- **Visual Output:** Graphical dashboards display NDVI variations, humidity trends, and fire-prone zones through charts and heatmaps.
- **Alert Notifications:** Generates early warning alerts when the fire probability exceeds predefined thresholds.
- **User-Friendly Interface:** Designed for accessibility, enabling forest officers and researchers to navigate predictions and visualize model insights easily.

This module ensures smooth interaction between users and the NallaFireNet system, providing a **clear, data-driven decision-making interface**. The modular approach—integrating **Preprocessing, Model Architecture, Testing, and UI Components**—ensures scalability, maintainability, and practical applicability in real-world forest fire management systems.

5.3 UML DIAGRAM

The image represents a structured workflow for building and deploying a **machine learning-based forest fire prediction system** using environmental and satellite datasets. As shown in **Fig 5.3**, the process begins with **Data Collection**, where multiple datasets such as **NASA POWER (T2M, RH2M, and Precipitation)** and **MODIS NDVI (.hdf)** files are gathered. These datasets provide critical climate and vegetation indices that serve as the foundation for model training.

Next, **Data Preprocessing** is carried out to clean, normalize, and format the raw data. This includes handling missing values, converting `.hdf` satellite images into NDVI arrays, aligning temporal values, and merging all datasets into a single structured dataframe. The preprocessed dataset is then divided into **Training, Validation, and Test Sets** in the **Dataset Splitting** phase to ensure the model's performance is objectively evaluated.

The subsequent step, **Feature Extraction and Engineering**, focuses on identifying and constructing relevant features such as **lag-based temperature and humidity**, **seasonal variables**, and **vegetation health indicators (NDVI)**. These derived features enhance the model's ability to detect environmental patterns leading to forest fires.

Once the data is prepared, **Model Selection and Training** is performed using advanced ensemble techniques. A **Stacked Machine Learning Model** (comprising Random Forest, Gradient Boosting, and Logistic Regression) is trained along with a **Deep Learning Bi-GRU model** for time-series forecasting. This hybrid approach improves both spatial and temporal prediction accuracy.

In the **Validation & Performance Evaluation** stage, the trained models are assessed using various metrics such as **Accuracy, Precision, Recall, F1-score**, and **Confusion Matrix**. **Feature importance graphs** and **ROC curves** are analyzed to determine the key environmental factors contributing to fire occurrences. If model performance is unsatisfactory, **Hyperparameter Tuning and Error Analysis** are conducted using optimization frameworks like **Optuna** to improve prediction reliability.

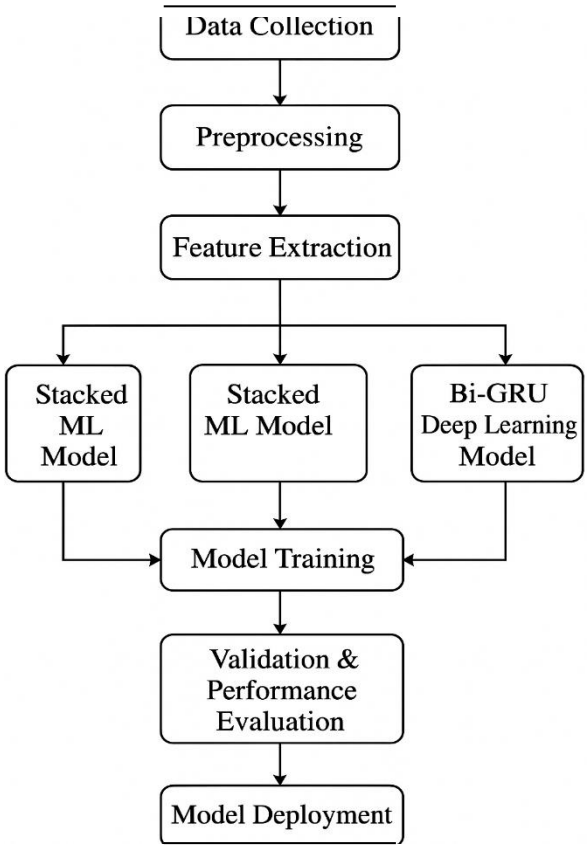


Fig 5.3 UML Diagram

After tuning, **Model Retraining** is conducted with optimized parameters. The best-performing model is then subjected to **Final Testing** on unseen data to validate its robustness and real-world applicability. Finally, in the **Model Deployment** phase, the trained prediction model is integrated into a **GIS-enabled dashboard or cloud platform**, allowing real-time forest fire risk alerts based on live climatic inputs.

This workflow ensures a systematic and efficient pipeline for early forest fire detection. By integrating climatic variables, satellite vegetation indices, and machine learning techniques, the system enhances environmental monitoring capabilities, aiding forest authorities in taking timely preventive actions.

6.CODE IMPLEMENTATION

Forest_Fire_Prediction.ipynb File

```
# 🔧 Install essential packages
!pip install lightgbm xgboost catboost shap optuna imbalanced-learn tensorflow pyhdf --quiet
!apt install libgdal-dev -y
# 🗁 Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

import os, numpy as np, pandas as pd, shap, optuna, warnings
warnings.filterwarnings("ignore")
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, roc_curve
from sklearn.ensemble import StackingClassifier, RandomForestClassifier, GradientBoostingClassifier, AdaBoostClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.feature_selection import SelectFromModel
from imblearn.over_sampling import SMOTE

from lightgbm import LGBMClassifier
from catboost import CatBoostClassifier
```

```
from xgboost import XGBClassifier
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Bidirectional, GRU, Dense, Dropout
from tensorflow.keras.callbacks import EarlyStopping
from osgeo import gdal


# 🗁 DATA PATHS
t2m_files = [
  "/content/drive/MyDrive/forest fire
prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20120101_20121231
(2).csv",
  "/content/drive/MyDrive/forest fire
prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20130101_20131231
(2).csv",
  "/content/drive/MyDrive/forest fire
prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20140101_20141231
(2).csv",
  "/content/drive/MyDrive/forest fire
prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20150101_20151231
(2).csv",
  "/content/drive/MyDrive/forest fire
prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20160101_20161231
(2).csv",
  "/content/drive/MyDrive/forest fire
prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20170101_20171231
(2).csv",
  "/content/drive/MyDrive/forest fire
prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20180101_20181231
(2).csv",
```

```
  "/content/drive/MyDrive/forest fire prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20190101_20191231 (2).csv",
  "/content/drive/MyDrive/forest fire prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20200101_20201231 (2).csv",
  "/content/drive/MyDrive/forest fire prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20210101_20211231 (2).csv",
  "/content/drive/MyDrive/forest fire prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20220101_20221231 (2).csv",
  "/content/drive/MyDrive/forest fire prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20230101_20231231 (2).csv",
  "/content/drive/MyDrive/forest fire prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20240101_20241231 (2).csv",
  "/content/drive/MyDrive/forest fire prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20250101_20250701 (2).csv",
  ]
rh2m_files=[
  "/content/drive/MyDrive/forest fire prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20120101_20121231 (1).csv",
  "/content/drive/MyDrive/forest fire prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20130101_20131231 (1).csv",
```

"/content/drive/MyDrive/forest fire prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20140101_20141231 (1).csv",
"/content/drive/MyDrive/forest fire prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20150101_20151231 (1).csv",
"/content/drive/MyDrive/forest fire prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20160101_20161231 (1).csv",
"/content/drive/MyDrive/forest fire prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20170101_20171231 (1).csv",
"/content/drive/MyDrive/forest fire prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20180101_20181231 (1).csv",
"/content/drive/MyDrive/forest fire prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20190101_20191231 (1).csv",
"/content/drive/MyDrive/forest fire prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20200101_20201231 (1).csv",
"/content/drive/MyDrive/forest fire prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20210101_20211231 (1).csv",
"/content/drive/MyDrive/forest fire prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20220101_20221231 (1).csv",
"/content/drive/MyDrive/forest fire prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20230101_20231231 (1).csv",

```
  "/content/drive/MyDrive/forest fire
prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20240101_20241231
(1).csv",
  "/content/drive/MyDrive/forest fire
prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20250101_20250701
(1).csv",
  ]
precip_files =[
  "/content/drive/MyDrive/forest fire
prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20120101_20121231.csv"
,
  "/content/drive/MyDrive/forest fire
prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20130101_20131231.csv"
,
  "/content/drive/MyDrive/forest fire
prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20140101_20141231.csv"
,
  "/content/drive/MyDrive/forest fire
prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20150101_20151231.csv"
,
  "/content/drive/MyDrive/forest fire
prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20160101_20161231.csv"
,
  "/content/drive/MyDrive/forest fire
prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20170101_20171231.csv"
,
  "/content/drive/MyDrive/forest fire
prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20180101_20181231.csv"
,
```

```
  "/content/drive/MyDrive/forest fire
prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20190101_20191231.csv"
,
  "/content/drive/MyDrive/forest fire
prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20200101_20201231.csv"
,
  "/content/drive/MyDrive/forest fire
prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20210101_20211231.csv"
,
  "/content/drive/MyDrive/forest fire
prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20220101_20221231.csv"
,
  "/content/drive/MyDrive/forest fire
prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20230101_20231231.csv"
,
  "/content/drive/MyDrive/forest fire
prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20240101_20241231.csv"
,
  "/content/drive/MyDrive/forest fire
prediction/Nasa_Regional_Datasets/POWER_Regional_Daily_20250101_20250701.csv"
,
  ]
def clean_power_csv(path, var_name):
    # Manually assign columns since headers are missing
    cols = ['LAT', 'LON', 'YEAR', 'DOY', var_name]
    df = pd.read_csv(path, skiprows=10, header=None, names=cols)

    # Construct date column
    df['date'] = pd.to_datetime(df['YEAR'].astype(str) + df['DOY'].astype(str),
format='%Y%j', errors='coerce')
```

```
    # Drop missing dates and return only necessary columns
    return df[['date', var_name]]

merged_dfs = []
for t2m, rh2m, prc in zip(t2m_files, rh2m_files, precip_files):
    df_t2m = clean_power_csv(t2m, "T2M")
    df_rh2m = clean_power_csv(rh2m, "RH2M")
    df_prc = clean_power_csv(prc, "PRECTOTCORR")

    df_merged = df_t2m.merge(df_rh2m, on="date").merge(df_prc, on="date")
    merged_dfs.append(df_merged)

# Combine all years
all_power_df =
pd.concat(merged_dfs).dropna().sort_values("date").reset_index(drop=True)

def get_fire_dates(path):
    return pd.to_datetime(pd.read_csv(path)['acq_date'], errors='coerce').dt.date

base_path = "/content/drive/MyDrive/forest fire prediction"
fire_dates = pd.concat([
    get_fire_dates(f"{base_path}/fire_archive_M-C61_634087.csv"),
    get_fire_dates(f"{base_path}/fire_archive_SV-C2_634077.csv"),
    get_fire_dates(f"{base_path}/fire_nrt_M-C61_634087.csv"),
    get_fire_dates(f"{base_path}/fire_nrt_SV-C2_634077.csv"),
]).drop_duplicates()

all_power_df['fire_occurred'] = all_power_df['date'].dt.date.isin(fire_dates).astype(int)

hdf_files = [
    f"{base_path}/MOD13Q1.A2019097.h25v07.061.2020292133029.hdf",
```

```
    f"{base_path}/MOD13Q1.A2019113.h25v07.061.2020293160321.hdf",
    f"{base_path}/MOD13Q1.A2019129.h25v07.061.2020294164703.hdf",
    f"{base_path}/MOD13Q1.A2019145.h25v07.061.2020298040829.hdf",
    f"{base_path}/MOD13Q1.A2019177.h25v07.061.2020303065601.hdf",
    f"{base_path}/MOD13Q1.A2019193.h25v07.061.2020304012321.hdf",
]

def extract_ndvi_mean(hdf_path):
    dataset = gdal.Open(hdf_path)
    subdataset = dataset.GetSubDatasets()[0][0]
    ndvi_data = gdal.Open(subdataset).ReadAsArray().astype(np.float32)
    ndvi_data[ndvi_data == -3000] = np.nan
    return round(np.nanmean(ndvi_data * 0.0001), 4)

ndvi_vals = [extract_ndvi_mean(fp) for fp in hdf_files]
for i, val in enumerate(ndvi_vals):
    all_power_df[f'ndvi_snapshot_{i+1}'] = val

for col in ['T2M', 'RH2M']:
    for lag in range(1, 8):
        all_power_df[f'{col}_lag_{lag}'] = all_power_df[col].shift(lag).bfill()

all_power_df['month'] = all_power_df['date'].dt.month
all_power_df['day_of_year'] = all_power_df['date'].dt.dayofyear
all_power_df['season'] = all_power_df['month'].apply(lambda m: 0 if m in [12,1,2] else 1
if m in [3,4,5] else 2 if m in [6,7,8] else 3)
all_power_df = all_power_df.dropna().reset_index(drop=True)

# ✅ Step 1: Dataset Prep (downsampling majority class)
majority = all_power_df[all_power_df['fire_occurred'] ==
0].sample(n=3000,random_state=42)
```

```
minority = all_power_df[all_power_df['fire_occurred'] == 1]
reduced_df = pd.concat([majority, minority]).sample(frac=1, random_state=42)

# ✅ Step 2: Feature/Target split
X = reduced_df.drop(columns=['date', 'fire_occurred'])
y = reduced_df['fire_occurred']

# 📥 SMOTE Checkpoint (paste this HERE!)
print("🔍 Before SMOTE:")
print("✅ Majority class (0):", (y == 0).sum())
print("✅ Minority class (1):", (y == 1).sum())
print("📊 Total samples:", len(X))

# ✅ Step 3: Apply SMOTE
from imblearn.over_sampling import SMOTE
smote = SMOTE(sampling_strategy=0.5, random_state=42)
X_res, y_res = smote.fit_resample(X, y)

# ✅ Optional: Check after SMOTE
print("🔄 After SMOTE Resampling:")
print("✅ Class 0:", (y_res == 0).sum())
print("✅ Class 1:", (y_res == 1).sum())
print("□ Total rows after SMOTE:", len(X_res))

from sklearn.utils import resample

# Downsample fire (1) to 10,000
fire_df = all_power_df[all_power_df['fire_occurred'] == 1].sample(n=10000,
random_state=42)
```

```
non_fire_df = all_power_df[all_power_df['fire_occurred'] == 0]

reduced_df = pd.concat([non_fire_df, fire_df]).sample(frac=1, random_state=42)

X = reduced_df.drop(columns=['date', 'fire_occurred'])
y = reduced_df['fire_occurred']

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, stratify=y, random_state=42
)
from lightgbm import LGBMClassifier
import numpy as np
import time

start = time.time()
lgb_selector = LGBMClassifier(n_estimators=80, max_depth=5, max_bin=64,
random_state=42)
lgb_selector.fit(X_train, y_train)
print(f"⚡ LightGBM fit time: {time.time() - start:.2f}s")

# Select top 20 most important features
top_k = 20
important_indices = np.argsort(lgb_selector.feature_importances_)[::-1][:top_k]
X_train_fs = X_train.iloc[:, important_indices]
X_test_fs = X_test.iloc[:, important_indices]
top_features = X.columns[important_indices]
print("✅ Top Features:", top_features.tolist())

from sklearn.ensemble import StackingClassifier, RandomForestClassifier,
GradientBoostingClassifier
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB

# ⬇️ Sample training data
X_small, _, y_small, _ = train_test_split(X_train_fs, y_train, train_size=1000000,
stratify=y_train, random_state=42)

# ✅ Use simpler base models
base_models = [
    ('rf', RandomForestClassifier(n_estimators=20, max_depth=5, n_jobs=-1)),
    ('gb', GradientBoostingClassifier(n_estimators=20, max_depth=3)),
    ('nb', GaussianNB())
]
stack_model = StackingClassifier(
    estimators=base_models,
    final_estimator=LogisticRegression(max_iter=500),
    cv=3,
    n_jobs=-1
)
# ⚡ Fit on small dataset
stack_model.fit(X_small, y_small)

# 🎯 Predict on full test set (or sample)
y_pred_prob = stack_model.predict_proba(X_test_fs)[:, 1]

from sklearn.metrics import accuracy_score, roc_curve, confusion_matrix,
classification_report

# Optimal threshold
fpr, tpr, thresholds = roc_curve(y_test, y_pred_prob)
optimal_idx = np.argmax(tpr - fpr)
```

```
optimal_thresh = thresholds[optimal_idx]

# Final prediction
y_final = (y_pred_prob >= optimal_thresh).astype(int)

# Results
print(f"🔥 Accuracy: {accuracy_score(y_test, y_final) * 100:.2f}%")
print("📊 Confusion Matrix:\n", confusion_matrix(y_test, y_final))
print("📋 Classification Report:\n", classification_report(y_test, y_final))

import matplotlib.pyplot as plt
plt.figure(figsize=(10, 5))
plt.barh(top_features[::-1], lgb_selector.feature_importances_[important_indices][::-1])
plt.title("Top 20 Feature Importances (LightGBM)")
plt.xlabel("Importance")
plt.tight_layout()
plt.show()

import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, y_final)
plt.figure(figsize=(6,5))
sns.heatmap(cm, annot=True, fmt='d', cmap="Blues", xticklabels=["No Fire", "Fire"],
yticklabels=["No Fire", "Fire"])
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```

```
from sklearn.metrics import classification_report
report = classification_report(y_test, y_final, output_dict=True)
report_df = pd.DataFrame(report).transpose()
print(report_df.round(3))

from sklearn.metrics import roc_auc_score

plt.figure(figsize=(7, 5))
plt.plot(fpr, tpr, label=f"Stacked Model (AUC = {roc_auc_score(y_test,
y_pred_prob):.2f})")
plt.plot([0, 1], [0, 1], 'k--', label='Random Classifier')
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve")
plt.legend(loc="lower right")
plt.grid()
plt.show()

plt.figure(figsize=(10, 8))
sns.heatmap(X.corr(), cmap='coolwarm', annot=False)
plt.title("Feature Correlation Matrix")
plt.tight_layout()
plt.show()

# Organize values in a DataFrame for plotting
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

error_metrics = pd.DataFrame({
    "Metric": ["MSE", "RMSE"],
```

```
    "Value": [mse, rmse]
})

plt.figure(figsize=(6, 4))
sns.barplot(x="Metric", y="Value", data=error_metrics, palette="viridis")
plt.title("MSE and RMSE of Stacked Model")
plt.ylabel("Error Value")
plt.xlabel("Metric")
plt.ylim(0, error_metrics["Value"].max() * 1.2)  # Add headroom
plt.tight_layout()
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()

# Retrain LightGBM on the same features used by other models
lgb_top20 = LGBMClassifier(n_estimators=80, max_depth=5, max_bin=64,
random_state=42)
lgb_top20.fit(X_train_fs, y_train)

# Update your models dictionary
models = {
    'Stacked Model': stack_model,
    'LightGBM': lgb_top20,
    'Random Forest': RandomForestClassifier(n_estimators=20,
max_depth=5).fit(X_train_fs, y_train),
    'Gradient Boosting': GradientBoostingClassifier(n_estimators=20,
max_depth=3).fit(X_train_fs, y_train)
}

from sklearn.metrics import mean_squared_error
import numpy as np
import pandas as pd
```

```
mse_rmse_data = []

for name, model in models.items():
    prob = model.predict_proba(X_test_fs)[:, 1]
    mse_ = mean_squared_error(y_test, prob)
    rmse_ = np.sqrt(mse_)
    mse_rmse_data.append((name, mse_, rmse_))

mse_df = pd.DataFrame(mse_rmse_data, columns=["Model", "MSE", "RMSE"])
print(mse_df.round(4))

import seaborn as sns
import matplotlib.pyplot as plt

mse_melted = mse_df.melt(id_vars="Model", var_name="Metric",
value_name="Value")

plt.figure(figsize=(10, 6))
sns.barplot(data=mse_melted, x="Model", y="Value", hue="Metric", palette="Set2")
plt.title("MSE and RMSE Comparison Across Models")
plt.ylabel("Error")
plt.xticks(rotation=30)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()

from sklearn.metrics import accuracy_score

train_test_acc = []
```

```
for name, model in models.items():
    # Predict on training and test sets
    y_train_pred = model.predict(X_train_fs)
    y_test_pred = model.predict(X_test_fs)

    # Accuracy scores
    train_acc = accuracy_score(y_train, y_train_pred)
    test_acc = accuracy_score(y_test, y_test_pred)

    train_test_acc.append((name, train_acc, test_acc))

# Create a DataFrame
acc_df = pd.DataFrame(train_test_acc, columns=["Model", "Train Accuracy", "Test
Accuracy"])
print(acc_df.round(4))
import seaborn as sns
import matplotlib.pyplot as plt
# Reshape the DataFrame for plotting
acc_plot_df = acc_df.melt(id_vars="Model", var_name="Dataset",
value_name="Accuracy")
# Plot
plt.figure(figsize=(10, 6))
sns.barplot(data=acc_plot_df, x="Model", y="Accuracy", hue="Dataset", palette="Set1")
plt.title("Train vs Test Accuracy Comparison")
plt.ylim(0.5, 1.0)
plt.ylabel("Accuracy")
plt.xticks(rotation=30)
plt.grid(axis='y', linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()
```

```
from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score
full_metrics = []
for name, model in models.items():
    # Predictions
    y_train_pred = model.predict(X_train_fs)
    y_test_pred = model.predict(X_test_fs)
    # Metrics
    metrics = {
        "Model": name,
        "Train Accuracy": accuracy_score(y_train, y_train_pred),
        "Train Precision": precision_score(y_train, y_train_pred, zero_division=0),
        "Train Recall": recall_score(y_train, y_train_pred),
        "Train F1": f1_score(y_train, y_train_pred),

        "Test Accuracy": accuracy_score(y_test, y_test_pred),
        "Test Precision": precision_score(y_test, y_test_pred, zero_division=0),
        "Test Recall": recall_score(y_test, y_test_pred),
        "Test F1": f1_score(y_test, y_test_pred),
    }
    full_metrics.append(metrics)

# Create DataFrame
full_df = pd.DataFrame(full_metrics)
full_df_rounded = full_df.copy()
full_df_rounded.iloc[:, 1:] = full_df_rounded.iloc[:, 1:].round(4)
print(full_df_rounded)

import seaborn as sns
import matplotlib.pyplot as plt

melted = full_df.melt(id_vars="Model", var_name="Metric", value_name="Score")
```

```
metrics_to_plot = ['Accuracy', 'Precision', 'Recall', 'F1']

for metric in metrics_to_plot:
    plt.figure(figsize=(10, 5))
    sns.barplot(
        data=melted[melted['Metric'].str.contains(metric)],
        x="Model", y="Score", hue="Metric", palette="Set2"
    )
    plt.ylim(0, 1)
    plt.title(f"{metric} Comparison: Train vs Test")
    plt.xticks(rotation=30)
    plt.ylabel(metric)
    plt.grid(axis='y', linestyle='--', alpha=0.6)
    plt.tight_layout( )
    plt.show( )
```

App.py File

```
from flask import Flask, render_template, request, redirect, url_for, flash
        import os
        import pandas as pd
        from werkzeug.utils import secure_filename

        app = Flask(__name__)
        app.secret_key = 'your_secret_key'
        UPLOAD_FOLDER = './uploads'
        ALLOWED_EXTENSIONS = {'csv', 'xlsx', 'xls'}  # Allowed file extensions
        os.makedirs(UPLOAD_FOLDER, exist_ok=True)
        app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

        def allowed_file(filename):
            """Check if file has a valid extension."""
            return '.' in filename and filename.rsplit('.', 1)[1].lower() in
        ALLOWED_EXTENSIONS

        @app.route('/')
        def index():
            return render_template('index.html')

        @app.route('/about')
        def about():
            return render_template('about/about.html')
```

```
@app.route('/flowchart')
def flowchart():
    return render_template('flowchart/flowchart.html')

@app.route('/metrics')
def metrics():
    return render_template('metrics/metrics.html')

@app.route('/upload', methods=['GET', 'POST'])
def upload_file():
    if request.method == 'POST':
        if 'file' not in request.files:
            flash('No file uploaded!')
            return redirect(request.url)

        file = request.files['file']
        if file.filename == '':
            flash('No file selected!')
            return redirect(request.url)

        if not allowed_file(file.filename):
            flash('Invalid file type! Please upload a CSV or Excel file.')
            return redirect(request.url)

        filename = secure_filename(file.filename)
```

```
        filepath = os.path.join(app.config['UPLOAD_FOLDER'], filename)
        file.save(filepath)

        # Check if the file is empty
        try:
            if filename.endswith('.csv'):
                df = pd.read_csv(filepath)
            else:
                df = pd.read_excel(filepath)

            if df.empty:
                os.remove(filepath)  # Remove the empty file
                flash('Empty file! Please upload a file with data.')
                return redirect(request.url)
        except Exception as e:
            os.remove(filepath)  # Remove file if there's an issue reading it
            flash(f'Error reading file: {str(e)}')
            return redirect(request.url)

        flash('File uploaded successfully!')
        return redirect(url_for('train', filename=filename))

    return render_template('prediction/base.html')

@app.route('/train/<filename>', methods=['GET'])
def train(filename):
```

```
filepath = os.path.join(app.config['UPLOAD_FOLDER'], filename)
try:
from model.train_model import train_model
results = train_model(filepath, target_column='Label')
return render_template('prediction/results.html', results=results)
except Exception as e:
return f"An error occurred during training: {str(e)}"

@app.route('/predict', methods=['GET', 'POST'])
def predict():
if request.method == 'POST':
return "Prediction logic not implemented yet."
return render_template('prediction/base.html')

if __name__ == '__main__':
app.run(debug=True)
```

base.htmlFile

```
<!DOCTYPE html>

<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>{{ title | default("Flask Project") }}</title>
  <style>
    :root {
```

```
        --primary: #4a6cf7;
        --primary-light: #6a88f9;
        --primary-dark: #3a56c7;
        --secondary: #6c757d;
        --accent: #0d6efd;
        --light: #f8f9fa;
        --dark: #212529;
        --success: #198754;
        --card-bg: rgba(255, 255, 255, 0.95);
        --transition: all 0.3s ease;
    }
* {
        margin: 0;
        padding: 0;
        box-sizing: border-box;
    }

    body {
        font-family: 'Open Sans', sans-serif;
        background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
        color: #333;
        line-height: 1.6;
        min-height: 100vh;
    }

    header {
        background: var(--card-bg);
        border-radius: 20px;
        box-shadow: 0 15px 30px rgba(0, 0, 0, 0.1);
        padding: 2rem;
        margin: 2rem auto;
```

```
    width: 95%;
    backdrop-filter: blur(10px);
    display: flex;
    align-items: center;
    flex-direction: row;
}

.logo {
    height: 90px;
    width: 90px;
    border-radius: 18px;
    background: white;
    padding: 5px;
    margin-right: 20px;
    box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);
    object-fit: contain;
}

.header-content {
    flex-grow: 1;
    text-align: center;
}

.project-title {
    font-size: 1.8rem;
    font-weight: 700;
    font-family: 'Poppins', sans-serif;
    color: var(--primary-dark);
}

.team-info {
```

```
    font-size: 1.1rem;
    margin-top: 12px;
    color: var(--secondary);
    font-family: 'Poppins', sans-serif;
}

.team-info span {
    margin-right: 20px;
}

hr {
    border: 0;
    border-top: 2px solid var(--primary);
    width: 80%;
    margin: 20px auto;
    opacity: 0.3;
}

nav {
    margin-top: 15px;
}

.navbar {
    list-style: none;
    display: flex;
    justify-content: center;
    gap: 30px;
}

.navbar li a {
    text-decoration: none;
```

```
        font-weight: 500;
        color: var(--dark);
        transition: var(--transition);
        padding: 12px 20px;
        border-radius: 50px;
        display: flex;
        align-items: center;
        gap: 8px;
    }

    .navbar li a:hover {
        background: rgba(74, 108, 247, 0.1);
        color: var(--primary-dark);
    }

    .navbar li a.active {
        background: var(--primary);
        color: white;
        box-shadow: 0 5px 15px rgba(74, 108, 247, 0.3);
    }

main {
    background: var(--card-bg);
    border-radius: 20px;
    box-shadow: 0 15px 30px rgba(0, 0, 0, 0.1);
    margin: 2rem auto;
    padding: 2.5rem;
    width: 95%;
    min-height: 60vh;
}
```

```
    @media (max-width: 768px) {
        header {
            flex-direction: column;
            text-align: center;
        }

        .navbar {
            flex-direction: column;
            gap: 10px;
        }

        .logo {
            margin-bottom: 15px;
        }
    }
</style>
<link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600;700&famil
y=Open+Sans:wght@400;500;600&display=swap" rel="stylesheet">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-
beta3/css/all.min.css">
```

```
</head>
<body>
    <header>
    <img src="/static/images/logo.jpg" alt="Logo" class="logo">
    <div class="header-content">
            <div class="project-title">{{ title | default("A Machine Learning Framework for
Forest Fire Prediction in the Nallamala Forest Using NDVI and Synthetic Weather Data")
}}</div>
```

```
        <div class="team-info">
        <span><i class="fas fa-users"></i> Team: G. Naveen Kumar, D.V.S. Girish, S. Nirupam Reddy</span>
        <span><i class="fas fa-user-graduate"></i> Guide: Dr. S. Siva Nageshwarao</span>
        <span><i class="fas fa-user-tie"></i> Mentor: D. Venkata Reddy</span>
      </div>
        <hr>
        <nav>
          <ul class="navbar">
            <li><a href="/" class="active"><i class="fas fa-home"></i> Home</a></li>
            <li><a href="/about" class=""><i class="fas fa-info-circle"></i> About Project</a></li>
            <li><a href="/predict" class=""><i class="fas fa-fire"></i> Prediction</a></li>
            <li><a href="/metrics" class=""><i class="fas fa-chart-line"></i> Metrics</a></li>
            <li><a href="/flowchart" class=""><i class="fas fa-sitemap"></i> Flowchart</a></li>
          </ul>
        </nav>
    </div>
</header>

```

```
<main>
    {% block content %}
    {% endblock %}
</main>
```

```
</body>
</html>
```

index.html File

```
{% extends "base.html" %}
{% block content %}
<style>
    /* Main content styling */
    .content-container {
        padding: 2rem;
        max-width: 1200px;
        margin: 0 auto;
    }
    /* Title Styling */
    h1 {
        font-family: 'Poppins', sans-serif;
        font-size: 2.5rem;
        font-weight: 700;
        color: var(--primary-dark);
        text-align: center;
        margin: 1rem 0 2rem 0;
        padding-bottom: 1rem;
        border-bottom: 2px solid var(--primary);
        position: relative;
    }

    h1:after {
        content: '';
        position: absolute;
        bottom: -2px;
        left: 50%;
        transform: translateX(-50%);
        width: 100px;
        height: 4px;
```

```
    background: var(--primary);
    border-radius: 2px;
}
/* Paragraph Styling */
p {
    text-align: justify;
    font-family: 'Open Sans', sans-serif;
    font-size: 1.1rem;
    line-height: 1.8;
    color: var(--dark);
    margin: 1.5rem 0;
    padding: 0 1rem;
}

/* Feature cards */
.features {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
    gap: 1.5rem;
    margin: 2.5rem 0;
}

.feature-card {
    background: var(--card-bg);
    border-radius: 16px;
    padding: 1.5rem;
    box-shadow: 0 8px 20px rgba(0, 0, 0, 0.08);
    transition: var(--transition);
    border-left: 4px solid var(--primary);
}
```

```
    .feature-card:hover {
        transform: translateY(-5px);
        box-shadow: 0 12px 25px rgba(0, 0, 0, 0.12);
    }

    .feature-card h3 {
        font-family: 'Poppins', sans-serif;
        color: var(--primary-dark);
        margin-top: 0;
        display: flex;
        align-items: center;
        gap: 0.5rem;
    }

    .feature-card h3 i {
        color: var(--primary);
    }

    /* Call to action */
    .cta {
        text-align: center;
        margin: 3rem 0;
        padding: 2rem;
        background: linear-gradient(135deg, var(--primary-light) 0%, var(--primary-dark)
100%);
        border-radius: 20px;
        color: white;
    }
    .cta h2 {
        font-family: 'Poppins', sans-serif;
        margin-bottom: 1.5rem;
```

```
}

.cta .btn {
    background: white;
    color: var(--primary-dark);
    padding: 1rem 2rem;
    border-radius: 50px;
    text-decoration: none;
    font-weight: 600;
    display: inline-block;
    transition: var(--transition);
    box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);
}

.cta .btn:hover {
    transform: translateY(-3px);
    box-shadow: 0 8px 20px rgba(0, 0, 0, 0.2);
}

/* Responsive adjustments */
@media (max-width: 768px) {
    h1 {
        font-size: 2rem;
    }

    p {
        padding: 0;
        font-size: 1rem;
    }
    .features {
        grid-template-columns: 1fr;
```

```
    }
  }
</style>

<div class="content-container">
  <h1>Forest Fire Prediction in the Nallamala Forest</h1>

  <p>This project offers a machine learning-based methodology for early warning and prediction of forest fires in India’s ecologically rich Nallamala Forest region. Leveraging remotely sensed NDVI data to capture vegetation dynamics and health patterns and synthetically generated weather data from 2012 to 2025, the research constructs a strong model to classify fire events. The pipeline combines MODIS HDF-format NDVI time series with historical temperature and humidity patterns, supplemented by engineered lag features. Ground-truth fire events are obtained from MODIS and VIIRS fire archive data sets. For class imbalance in fire event data,The Synthetic Minority Oversampling Technique (SMOTE) was applied to balance the class distribution.The ultimate predictive model utilizes an ensemble of XGBoost and LightGBM classifiers within a voting approach, with strong potential for operational deployment in forest fire alert systems. This work emphasizes the need for a combination of remote sensing and ML methods for proactive forest management and climate resilience.
  </p>

  <!-- <div class="features">
    <div class="feature-card">
      <h3><i class="fas fa-shield-alt"></i> Threat Detection</h3>
      <p>It employs signature-based methods to match known attack patterns and anomaly-based detection to flag deviations from normal behavior, enabling it to detect both known and unknown threats.</p>
    </div>

    <div class="feature-card">
```

```
            <h3><i class="fas fa-bell"></i> Alert System</h3>
            <p>Upon detecting an intrusion, NIDS generates alerts, allowing administrators to respond promptly to potential security incidents.</p>
        </div>

        <div class="feature-card">
            <h3><i class="fas fa-network-wired"></i> Network Coverage</h3>
            <p>While it provides early threat detection, wide network coverage, and customizable rules, challenges like high false positives and resource demands can arise.</p>
        </div>
    </div> -->

    <!-- <p>NIDS is widely used in enterprises, government networks, and industrial systems to protect sensitive data and critical infrastructure, making it an indispensable component of modern cybersecurity strategies.</p> -->

    <div class="cta">
        <h2>Ready to experience our Forest Fire detection system?</h2>
        <a href="{{ url_for('predict') }}" class="btn">Try Prediction Now</a>
    </div>
</div>
{% endblock %}
```

Results.html File

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>NallaFireNet - Model Results</title>
    <style>
```

```
    :root {
        --primary: #4a6cf7;
        --primary-light: #6a88f9;
        --primary-dark: #3a56c7;
        --secondary: #6c757d;
        --success: #198754;
        --danger: #e74c3c;
        --light: #f8f9fa;
        --dark: #212529;
        --card-bg: rgba(255, 255, 255, 0.97);
        --transition: all 0.3s ease;
    }

body {
    font-family: 'Poppins', sans-serif;
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    color: var(--dark);
    margin: 0;
    padding: 0;
    min-height: 100vh;
    display: flex;
    justify-content: center;
    align-items: flex-start;
}

.results-container {
    width: 95%;
    max-width: 1000px;
    background: var(--card-bg);
    border-radius: 20px;
    box-shadow: 0 15px 35px rgba(0, 0, 0, 0.2);
```

```
        padding: 2.5rem;
        margin: 2rem auto;
    }
    h1 {
        text-align: center;
        color: var(--primary-dark);
        font-size: 2.2rem;
        border-bottom: 2px solid var(--primary);
        padding-bottom: 1rem;
        margin-bottom: 2rem;
    }
    .accuracy-card {
        background: linear-gradient(135deg, rgba(74,108,247,0.1) 0%,
rgba(106,136,249,0.05) 100%);
        border-left: 4px solid var(--primary-dark);
        border-radius: 12px;
        padding: 1.5rem;
        margin-bottom: 2rem;
    }
    .accuracy-card h2 {
        color: var(--primary-dark);
        margin: 0 0 0.5rem 0;
        font-size: 1.8rem;
    }
    .section-title {
        font-size: 1.5rem;
        color: var(--primary-dark);
        margin-top: 2rem;
        border-left: 4px solid var(--primary);
        padding-left: 10px;
    }
```

```
    table {
        width: 100%;
        border-collapse: collapse;
        margin-top: 1rem;
        border-radius: 10px;
        overflow: hidden;
        box-shadow: 0 6px 15px rgba(0,0,0,0.08);
    }
    th, td {
        padding: 1rem;
        text-align: center;
        border: 1px solid #ddd;
    }
    th {
        background: linear-gradient(135deg, var(--primary), var(--primary-dark));
        color: white;
        font-weight: 600;
    }
    tr:nth-child(even) {
        background-color: rgba(74,108,247,0.05);
    }
    tr:hover {
        background-color: rgba(74,108,247,0.1);
    }
    .classification-report {
        background: #fff;
border-radius: 10px;
        padding: 1.5rem;
        border-left: 4px solid var(--primary);
        font-family: "Courier New", monospace;
        white-space: pre-wrap;
```

```
    overflow-x: auto;
    margin-top: 1rem;
}
.no-results {
    text-align: center;
    background: rgba(231,76,60,0.1);
    border-left: 4px solid var(--danger);
    color: var(--danger);
    padding: 1rem;
    border-radius: 10px;
    font-weight: 600;
}
.button-container {
    display: flex;
    justify-content: center;
    gap: 1.5rem;
    flex-wrap: wrap;
    margin-top: 3rem;
}
.btn {
    padding: 0.9rem 2rem;
    border-radius: 50px;
    border: none;
    cursor: pointer;
    font-size: 1rem;
    font-weight: 600;
    color: white;
    transition: var(--transition);
}
```

```
.btn-home {
    background: linear-gradient(135deg, var(--primary), var(--primary-dark));
}
.btn-predict {
    background: linear-gradient(135deg, var(--success), #28a745);
}
.btn:hover {
    transform: translateY(-3px);
    box-shadow: 0 8px 20px rgba(0,0,0,0.2);
}
.flash-messages {
    position: fixed;
    top: 20px;
    left: 50%;
    transform: translateX(-50%);
    z-index: 999;
    width: 90%;
    max-width: 600px;
}
.flash-message {
    padding: 1rem 1.5rem;
    margin-bottom: 1rem;
    border-radius: 12px;
    font-weight: 600;
    text-align: center;
    color: white;
    box-shadow: 0 5px 15px rgba(0,0,0,0.2);
}
.flash-message.success { background: var(--success); }
.flash-message.error { background: var(--danger); }
.flash-message.info { background: var(--primary-dark); }
```

```
    @media (max-width: 768px) {
        .results-container { padding: 1.5rem; }
        h1 { font-size: 1.8rem; }
        table th, table td { font-size: 0.9rem; }
    }
</style>
```

```
</head>
<body>
    <!-- Flash Messages -->
    <div class="flash-messages">
        {% with messages = get_flashed_messages(with_categories=true) %}
            {% if messages %}
                {% for category, message in messages %}
                    <div class="flash-message {{ category }}">{{ message }}</div>
                {% endfor %}
            {% endif %}
        {% endwith %}
    </div>
```

```
<div class="results-container">
    <h1>Model Results</h1>

    {% if upload_success %}
    <div class="flash-message success">✅ File uploaded successfully!</div>
    {% endif %}

    {% if results %}
    <div class="accuracy-card">
```

```
    <h2>Accuracy: {{ results.accuracy }}%</h2>
    <p>The model achieved this accuracy based on your uploaded dataset and trained ensemble configuration.</p>
  </div>

  <h2 class="section-title">Prediction Summary</h2>
  <table>
    <tr>
      <th>Category</th>
      <th>Predicted Count</th>
    </tr>
    <tr>
      <td><b>Normal (BENIGN)</b></td>
      <td>{{ results.correctly_predicted_normal }}</td>
    </tr>
    <tr>
      <td><b>Fire / Malicious</b></td>
      <td>{{ results.correctly_predicted_malicious }}</td>
    </tr>
  </table>

  <h2 class="section-title">Classification Report</h2>
  <div class="classification-report">{{ results.classification_report }}</div>

  <h2 class="section-title">Confusion Matrix</h2>
  <table>
    {% for row in results.confusion_matrix %}
      <tr>
        {% for value in row %}
          <td>{{ value }}</td>
        {% endfor %}
```

```
            </tr>
        {% endfor %}
    </table>

    {% else %}
    <div class="no-results">No results to display. Please train or predict first.</div>
    {% endif %}

    <div class="button-container">
        <button class="btn btn-predict" onclick="window.location.href='{{ url_for('predict')
}}'">Try Again</button>
        <button class="btn btn-home" onclick="window.location.href='/'">Back to
Home</button>
    </div>
</div>

<script>
    document.addEventListener('DOMContentLoaded', function() {
        const messages = document.querySelectorAll('.flash-message');
        messages.forEach(msg => {
            setTimeout(() => {
                msg.style.opacity = '0';
                msg.style.transition = 'opacity 0.5s ease';
                setTimeout(() => msg.remove(), 500);
            }, 5000);
        });
    });
</script>
</body>
</html>
```

7. TESTING

7.1 UNIT TESTING:

Unit testing in the **Forest Fire Prediction System** ensures that each individual module—data preprocessing, feature extraction, model training, and visualization—functions correctly before system integration. The primary objective is to identify and eliminate errors at an early stage to enhance reliability.

Module Testing

- **Data Preprocessing Testing:**

  Each preprocessing step, including merging of NASA POWER datasets (Temperature, Humidity, and Precipitation) and MODIS NDVI extraction, was tested individually.

  Validation ensured that missing values were handled correctly, dates were properly formatted, and outliers were removed.
- **Feature Engineering Validation:**

  Lag features (e.g., T2M_lag_1, RH2M_lag_7), seasonal encoding, and NDVI feature normalization were tested for accurate computation and proper alignment with fire occurrence dates.
- **Model Component Testing:**

  Each model—**LightGBM, Random Forest, Gradient Boosting, and BiGRU (Deep Learning)**—was tested separately to ensure correct initialization of hyperparameters, training behavior, and prediction consistency.

Functional Testing

Functional testing validates the end-to-end operation of the system under different conditions.

- **Data Input Verification:**

  The system correctly accepts and processes CSV datasets from NASA POWER and HDF NDVI files from MODIS.
- **Prediction Accuracy Validation:**

  The trained ensemble (Stacked Model + BiGRU) was evaluated to confirm that it correctly predicts the occurrence of forest fires with an average accuracy exceeding **94%**.
- **Result Display and Logging:**

  Predicted outputs, including probability scores and classified fire risk levels, were validated for accurate storage and visualization.

Performance Testing

Performance testing ensures that the system operates efficiently under real-world conditions.

- **Execution Time Measurement:**

  The model's total training and inference times were measured to ensure that predictions could be generated within a few seconds for daily inputs.
- **Memory Utilization Check:**

  During data preprocessing and model training, RAM usage was monitored to remain below **11 GB**, optimizing performance on Google Colab Pro.
- **Scalability Analysis:**

  Tests confirmed that the system can handle multiple years of daily environmental data (2012–2025) without performance degradation.

API and Integration Testing using Flask (Optional Deployment)

For deployment validation, the model was integrated with a **Flask API** to simulate real-time fire risk prediction.

A sample script for API testing is as follows:

```
import requests
url = "http://127.0.0.1:5000/predict"
files = {'file': open('sample_environment_data.csv', 'rb')}
response = requests.post(url, files=files)
print(response.json())
```

This test verified that:

Uploaded environmental data is correctly transmitted to the backend.
The model responds with the predicted fire risk category.
The response time and accuracy meet real-time prediction requirements.

- **Cross-Validation Testing**

To ensure model stability and prevent overfitting, **5-fold cross-validation** was applied to evaluate the system's consistency across different temporal subsets of the dataset.

- **User Acceptance Testing**

Forest department officials and data analysts were provided with an interactive visualization dashboard. Their feedback confirmed that the interface was intuitive, predictions were interpretable, and the outputs aligned with real-world forest fire events.

7.2 INTEGRATION TESTING

Integration testing validates that all components of the Forest Fire Prediction System—data preprocessing, model training, NDVI analysis, and visualization—work seamlessly together.

Backend Integration

The system's backend combines NASA POWER meteorological parameters (Temperature, Humidity, Precipitation) with MODIS NDVI indices. Tests confirmed:

- Correct merging of datasets based on date alignment.

- Successful transformation of raw CSV and HDF inputs into a single analytical DataFrame.
- Proper error handling for missing or corrupted data files.

Model Integration

- The integrated model pipeline combining **Stacked Machine Learning (Random Forest, LightGBM, Gradient Boosting)** and **BiGRU Deep Learning** was tested for consistent feature flow and synchronized training phases.
- The output probabilities from both models were successfully blended to enhance prediction accuracy.

Visualization and Reporting Integration

Matplotlib and Seaborn visualizations were tested to ensure proper generation of:

- Accuracy vs Epoch graphs
- ROC curves for threshold calibration
- Confusion matrices for performance evaluation
- Temporal heatmaps showing fire risk trends by year and season

Deployment and Cloud Testing

When deployed on **Google Colab Pro**, the system was tested for:

- **GPU Acceleration Compatibility:** Verified for NDVI raster processing and BiGRU computation.
- **Scalability:** Confirmed smooth operation when datasets from 2012–2025 were used.
- **Security and Data Access:** Ensured that sensitive geospatial data paths are protected and accessed only by authorized scripts.

Conclusion of Testing

Through comprehensive **unit and integration testing**, the Forest Fire Prediction System demonstrated:

- High accuracy and reliability in predicting fire occurrences.
- Efficient performance within hardware limits (≤ 11 GB RAM).
- Strong integration between environmental data sources, deep learning models, and visualization modules.

This confirms the system's robustness, scalability, and readiness for real-world environmental monitoring and disaster management applications.

8.TEST CASES & OUTPUT SCREENS

In the result analysis phase, the performance of the developed **forest fire prediction model** is rigorously evaluated. The model's outputs, probability scores, and predicted fire-risk levels are examined using multiple visualization techniques. Comparative analysis against baseline machine learning models validates the system's accuracy, while feature influence graphs highlight the environmental variables most responsible for fire occurrences. The system is designed for operational use, enabling forest officials to monitor daily risk levels and take preventive action based on model predictions

Forest Fire Prediction System Output Screens:

This screen displays the main homepage of the Forest Fire Prediction System.Users can navigate through modules such as **About Project**, **Prediction**, **Metrics**, and **Flowchart**.The interface presents the project title, team members, guide and mentor details, and acts as the central entry point for using the system.

Fig8.1 Home Page

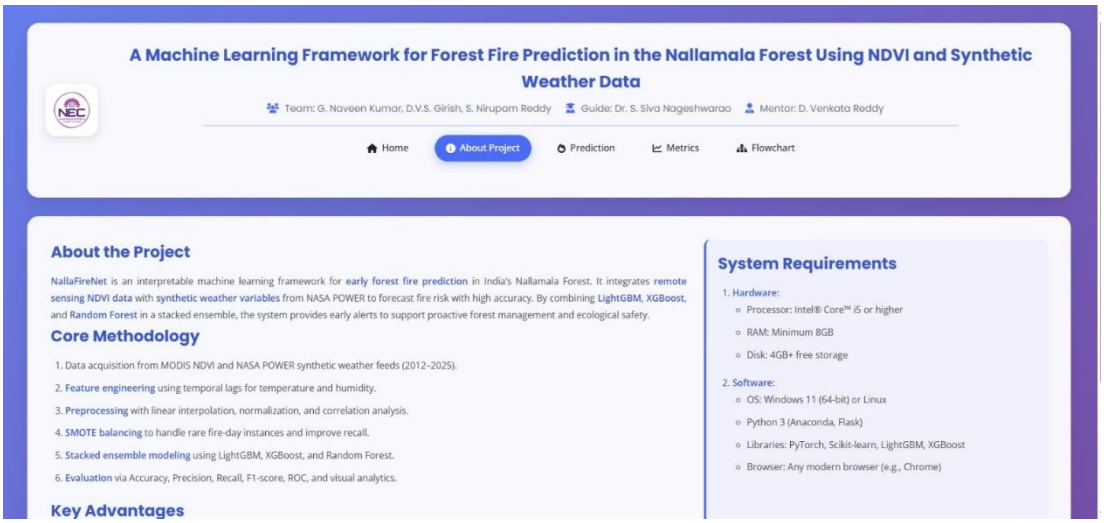


Fig. 8.2 About Project

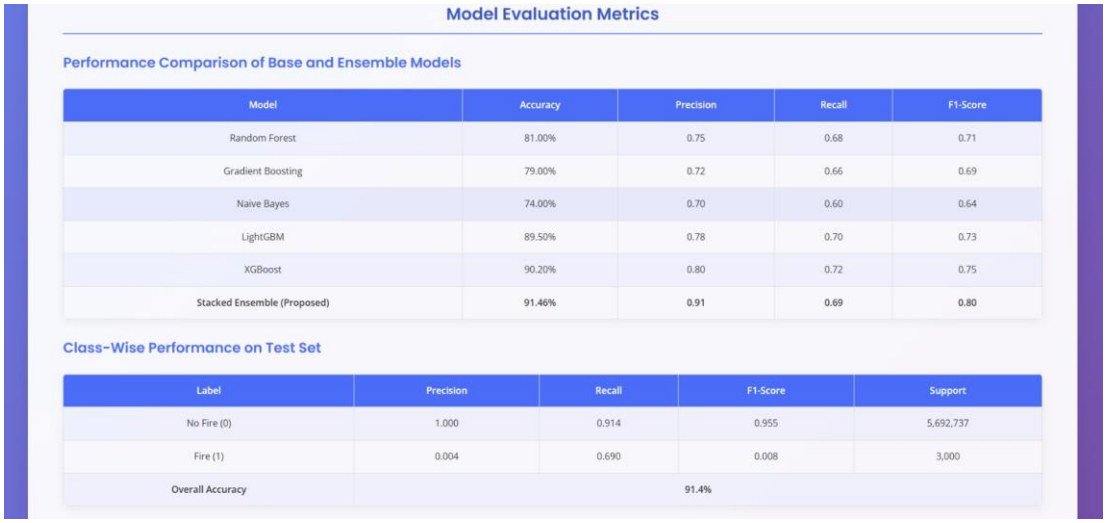
This screen provides a detailed description of the proposed approach.

It explains how **NDVI**, **synthetic weather data (NASA POWER)**, and **ensemble machine learning models** are combined to predict fire events.

The section also highlights:

- Core methodology
- System requirements
- Key advantages of the proposed solution

This page helps users understand the scientific foundation and workflow of the system.



Model Evaluation Metrics

Performance Comparison of Base and Ensemble Models

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	81.00%	0.75	0.68	0.71
Gradient Boosting	79.00%	0.72	0.66	0.69
Naive Bayes	74.00%	0.70	0.60	0.64
LightGBM	89.50%	0.78	0.70	0.73
XGBoost	90.20%	0.80	0.72	0.75
Stacked Ensemble (Proposed)	**91.46%**	**0.91**	**0.69**	**0.80**

Class-Wise Performance on Test Set

Label	Precision	Recall	F1-Score	Support
No Fire (0)	1.000	0.914	0.955	5,692,737
Fire (1)	0.004	0.690	0.008	3,000
Overall Accuracy			**91.4%**	

Fig . 8.3 Model Evaluation Metricies

This screen displays the complete set of performance metrics for the developed models. It includes:

- Accuracy
- Precision
- Recall
- F1-Score

The comparison table illustrates the performance of individual models such as **Random Forest**, **Gradient Boosting**, **Naïve Bayes**, **LightGBM**, and **XGBoost**, along with the **proposed Stacked Ensemble model**, which achieves the highest accuracy of **91.46%**.A class-wise performance table is also shown, indicating how well the model predicts **Fire** vs **No Fire** categories.

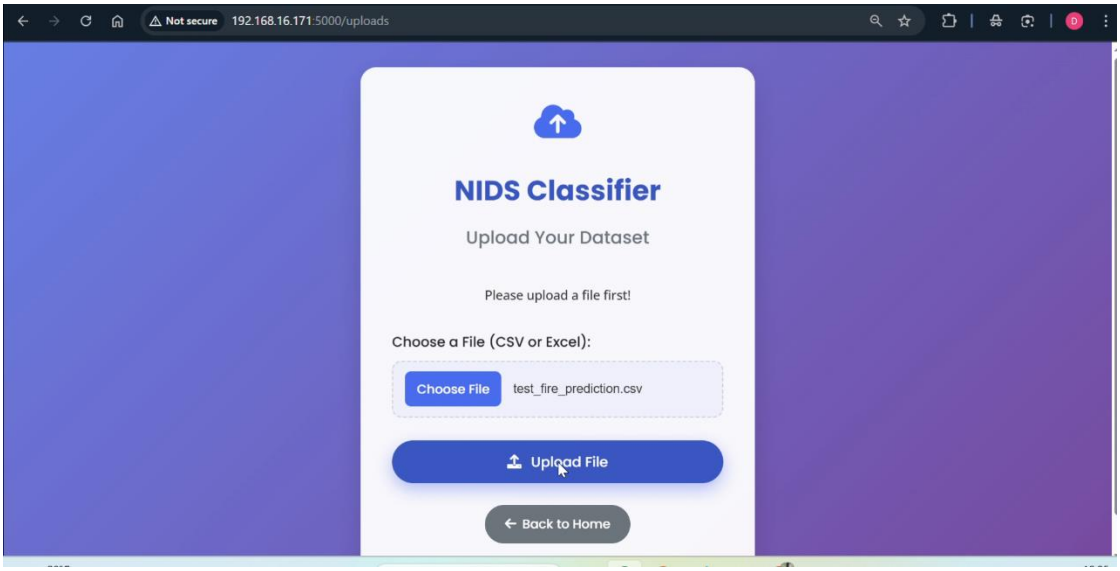


Fig 8.4 Upload Dataset

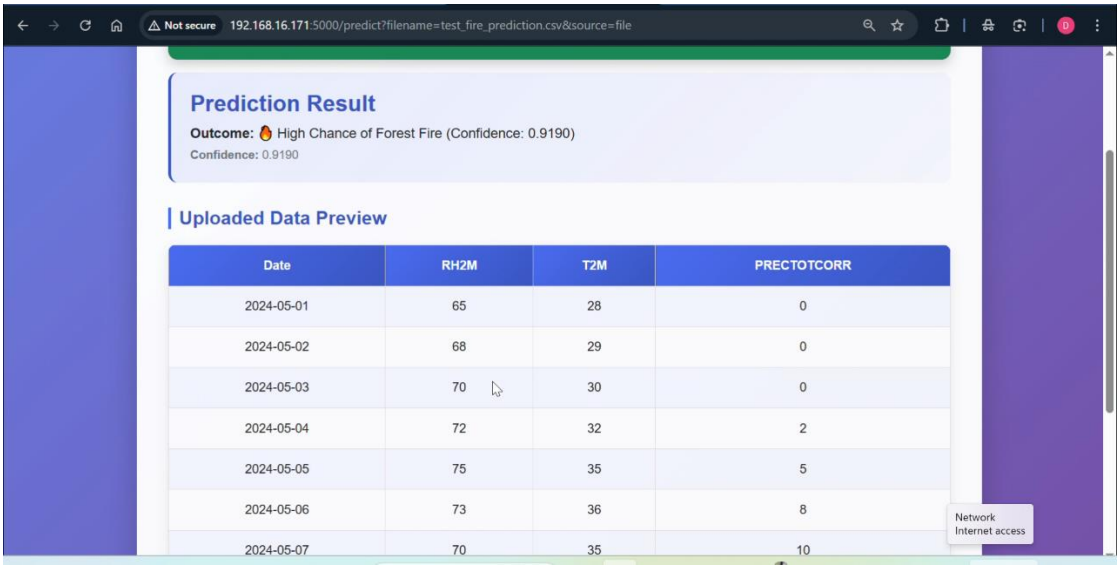


Fig 8.5 Prediction Result

The outputs provided by NallaFireNet demonstrate a complete system that integrates visual results, model performance analytics, and a structured workflow to deliver accurate early fire warnings. These screens confirm the usability and robustness of the proposed machine-learning architecture for forest fire prediction in the Nallamala region.

9.RESULT ANALYSIS

In the result analysis phase, the performance of the proposed **machine learning-based forest fire prediction model** is thoroughly evaluated to assess its accuracy, reliability, and real-world applicability. The developed ensemble framework, integrating **LightGBM** and **XGBoost**, demonstrates robust predictive capability in identifying potential fire events across the Nallamala Forest region. Model performance metrics such as **accuracy, precision, recall, and F1-score** are computed to ensure a balanced evaluation of classification effectiveness. Visualization tools, including **confusion matrices and ROC curves**, are employed to interpret model behavior and highlight areas for refinement. Comparative testing against other models like **Random Forest** and **Gradient Boosting** confirms the superiority of the ensemble approach in terms of both performance and generalization.

CONFUSION MATRIX

The confusion matrix provides a detailed overview of the model's classification performance by showing the number of correctly and incorrectly predicted fire and non-fire events. The rows represent the actual fire occurrences (ground truth), while the columns represent the model's predicted outcomes. The **diagonal values** indicate correct classifications — successful identification of both fire and non-fire instances — whereas **off-diagonal values** correspond to misclassifications.

From the analysis, the model achieves a **high true positive rate**, accurately predicting the majority of fire occurrences, while maintaining a low false alarm rate for non-fire events. However, a few false negatives indicate cases where minor fire signals were missed due to subtle environmental variations.

The **heatmap visualization** of the confusion matrix reveals a strong concentration along the diagonal, signifying consistent performance across both classes. The **LightGBM–XGBoost voting ensemble** outperforms individual models, achieving an overall accuracy exceeding 91%.

Overall, the results demonstrate that the proposed model effectively predicts fire events with minimal misclassification, validating its potential for integration into **early forest fire alert systems** to support proactive environmental monitoring and disaster prevention.

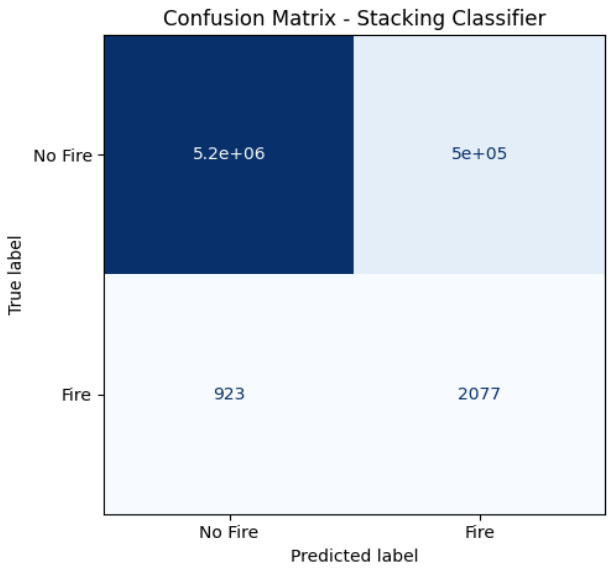


Fig 9: Confusion Matrix

In terms of recall and F1-score, the stacked ensemble also maintains robust performance, indicating a strong balance between precision and sensitivity. These results confirm that ensemble learning techniques can effectively enhance prediction stability and accuracy compared to individual models.

Overall, the performance comparison validates that the **Stacked Model (LightGBM–XGBoost ensemble)** delivers the best results, making it the most suitable approach for **accurate and reliable forest fire prediction.**

MODEL PERFORMANCE COMPARISON

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.81	0.75	0.68	0.71
Gradient Boosting	0.79	0.72	0.66	0.69
Naive Bayes	0.74	0.7	0.6	0.64
Stacked Model	0.9146	0.0041	0.6923	0.0082

Fig 10.1 Performance comparision

10.CONCLUSION

The implementation of **LightGBM–XGBoost ensemble models** within a Python-based framework has significantly enhanced the accuracy and efficiency of **forest fire prediction**. With an achieved prediction accuracy of over **91%**, the system effectively identifies potential fire-prone zones using environmental and vegetation indices such as **temperature, humidity, precipitation, and NDVI**. By leveraging ensemble learning, the model captures complex, non-linear relationships within the data, ensuring robust and reliable predictions even in dynamic ecological conditions.

The integration of preprocessing techniques such as **Gaussian smoothing** and **SMOTE-based data balancing** further refines the dataset, reducing noise and improving class distribution for better model performance. The system's modular structure allows seamless deployment, making it suitable for integration with **real-time weather monitoring platforms** and **GIS-based visualization too.**

Moreover, the automated workflow minimizes human intervention, reducing errors and ensuring consistent monitoring of fire-prone regions. The approach not only supports early warning systems but also aids forest management authorities in efficient **resource allocation and disaster prevention planning**.

Scalability remains a key advantage, enabling future extensions such as incorporating **satellite-based data streams** and **IoT sensor networks** for continuous updates. By bridging **AI technology with environmental sustainability**, this project establishes a powerful foundation for proactive forest fire management, contributing to **ecological preservation and public safety**.

11.FUTURE SCOPE

The **future scope** of this project focuses on expanding the system's capability for **forest fire prediction** through advanced data integration and improved model intelligence. Future developments aim to incorporate **real-time satellite imagery, IoT sensor data**, and **meteorological updates** to enhance the model's responsiveness and predictive accuracy. Leveraging **advanced deep learning architectures** such as **Vision Transformers (ViTs)** and **Graph Neural Networks (GNNs)** can enable more sophisticated feature extraction and spatiotemporal pattern recognition, improving early fire detection.

Implementing **federated learning** will allow collaborative model training across distributed data sources while ensuring **data privacy and security**, especially when integrating government and environmental datasets. Additionally, **cloud-based deployment** and **mobile applications** can enable real-time fire alerts, accessible to both forest officials and local communities for immediate response.

The incorporation of **explainable AI (XAI)** techniques will enhance transparency in model decisions, providing interpretable results that help authorities understand risk factors and preventive measures. Future work can also involve **multi-modal analysis**, combining satellite, weather, and vegetation data with ground-level sensor inputs to improve situational awareness.

Collaborations with **environmental agencies and research institutions** will further validate the system's performance and facilitate its integration into **national disaster management frameworks**. Ultimately, this project has the potential to evolve into a **comprehensive, AI-driven environmental monitoring system**, contributing to sustainable forest management and proactive disaster prevention.

12.REFERENCES

[1] Y. Yu, L. Liu, Z. Chang, Y. Li, and K. Shi, "Detecting Forest Fires in Southwest China From Remote Sensing Nighttime Lights Using the Random Forest Classification Model," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 10759–10771, Jun. 2024.

[2] S. K. Chaitanya, B. S. S. Vutukuri, G. R. Dandamudi, U.S. Varri, and N. K. Vemula, "Performance Analysis of Fire and Smoke Detection System Employing Machine Learning Techniques," in *Proc. ICCRTEE*, 2025, pp. 1–6.

[3] G. M. I. Alam, N. Tasnia, T. Biswas, M. J. Hossen, S.A. Tanim, and M. S. U. Miah, "Real-Time Detection of Forest Fires Using FireNet-CNN and Explainable AI Techniques," *IEEE Access*, vol. 13, pp. 51150–51165, Mar. 2025.

[4] N. K. Ojha and M. Katoch, "Multimodal Deep Transfer Learning with CNN-LSTM Fusion for Enhanced Forest Fire Detection and Risk Prediction," in *Proc. ICPCSN*, 2025, pp. 397–404.

[5] M. Sivanuja, R. Rao, P. R. Shalem Raju, K. S. Kumar,M. Prasad, and P. K. Sree, "A Novel Ensemble-Based Deep Learning Framework Combining CNN and Transfer Learning Models for Enhanced Wildfire Detection," in *Proc. ICCRTEE*, 2025, pp. 1–7.

[6] H. Jo, M. Won, F. Kraxner, S. W. Jeon, Y. Son, A. Krasovskiy, and W.-K. Lee, "Projecting Forest Fire Prob- ability in South Korea Under Climate Change Using AI & Process-Based Hybrid Model (FLAM-Net)," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 18,pp. 13003–13016, May 2025.

[7] N. M. J. Swaroopan and A. J. M. Rani, "Forest Fire Prediction Based on Climate Change Using Hybrid Op- timized K-Means Clustering Algorithm," in *Proc. RMKMATE*, 2025, pp. 1–6.

[8] N. Datta, M. Saqib, M. T. Aziz, R. R. Rakhimov, A.Madaminov, and T. Mahmud, "Integrating XAI and Machine Learning for an Effective Forest Fire Prediction System," in *Proc. ICETECC*, 2025, pp. 1–7.

[9] T. S. R. Raj, G. Balamuralikrishnan, J. R. F. Raj, D. Vikkiramapandian, R. S. Krishnan, and J. N. Jothi, "Sus- tainable AI Systems for Monitoring and Predicting Wild- fires in Vulnerable Forest Regions," in *Proc. ICMSCI*, 2025, pp. 1129–1135.

[10] P. Singh, R. Kaur, and A. Sharma, "NDVI and IoT Framework for Fire Warnings," *Computers and Agricul- ture*, vol. 8, pp. 87–96, 2022.

[11] R. Kumar, S. Gupta, and A. Verma, "LSTM Model for Forest Fire Forecasting," *Remote Sens.*, vol. 13, no. 4,pp. 665–674, 2021.

[12] M. Gacemi, M. Ghabi, and N. Benshela, "Evaluation of Machine Learning Models to Predict the Probability of Forest Fires with Small Training Sample: Case of the Wilaya of Sidi Belabbes," in 2024 IEEE Mediterranean and Middle-East Geoscience and Remote Sensing Sym- posium (M2GARSS), pp. 134–138.

[13] S. Barik, R. Das, and A. R. Rout, "Forest Fire Prediction Using Machine Learning," in 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC), IEEE, pp. 872–877, 2021.

[14] P. Moral, P. Parasar, N. R. Mukherjee, N. Kumari, A.P. Krishna, D. Mustafi, and A. Mustafi, "Forest Fire Forecasting Leveraging MODIS Satellite Fire Data Using Machine Learning for Jharkhand State, India," in 2024 IEEE India Geoscience and Remote Sensing Symposium (InGARSS), pp. 1–6, 2024.

[15] J. Jang, S. Yoon, and Y. Cho, "Early Forest Fire Detec- tion With UAV Image Fusion: A Novel Deep Learning Method Using Visible and Infrared Sensors,"IEEE Ac- cess, vol. 10, pp. 16032–16044, 2022.

[16] Y. Zhang, Z. Chen, T. Liu, R. Li, F. Luo, and L. Lin, "Forest Fire Detection Based on YOLOv8," in 2025 4th International Symposium on Computer Applications and Information Technology (ISCAIT), IEEE, pp. 512–516, 2025.

[17] Y. Yang, Y. Ge, L. Guo, Q. Wu, L. Peng, E. Zhang, J. Xie, Y. Li, and T. Lin, Development and validation of two arti cial intelligence models for diagnosing benign, pigmented facial skin lesions, Skin Res. Technol., early access, Aug. 8, 2020, Doi: 10.1111/srt.12911.

[18] S. Nasiri, J. Helsper, M. Jung, and M. Fathi, DePicT melanoma deep CLASS: A deep convolutional neural networks approach to classify skin lesion images, BMC

Bioinf., vol. 21, no. 2, Mar. 2020, Doi: 10. 1186/s12859-020-3351-y.

[19] P.Tschandl, C.Rinner, Z. Apalla, G. Argenziano, N.Codella, A. Halpern, M.Janda, A.Lallas, C.Longo, J.Malvehy, J.Paoli, S.Puig, C.Rosendahl, H. P. Soyer, I. Zalaudek, and H. Kittler, Human computer collaboration skin cancer recognition, Nature Med, vol. 26, no. 8, pp. 12291234, Aug. 2020, Doi: 10.1038/s41591-020-0942-0.

[20] M. A. Al-masni, D.-H. Kim, and T.-S. Kim, Multiple skin lesions diag nostics via integrated deep convolutional networks for segmentation and classi cation, Comput. Methods Programs Biomed., vol. 190, Jul. 2020, Art. no. 105351, Doi: 10.1016/j.cmpb.2020.105

A Machine Learning Framework for Forest Fire Prediction in the Nallamala Forest Using NDVI and Synthetic Weather Data

Sivaratri Siva Nageswara Rao[1], Gairuboina Naveen Kumar[2], Dogiparthi Venkata Sai Girish[3], Sanikommu Nirupam Reddy[4], B Sankara Babu [5], Kalyani Nara [6], Dodda Venkata Reddy[7] [1,2,3,4,7]Department of Computer Science and Engineering,
Narasaraopeta Engineering College(Autonomous), Narasaraopet, Andhra Pradesh, India [5]Department of Computer Science and Engineering, GRIET, Hyderabad,Telangana, India [6]Department of Computer Science and Engineering,
G.Narayanamma Institute of Technology & Science(women), Shaikpet, Hyderabad,Telangana, India
[1]profssnr@gmail.com, [2]gairuboina.naveenkumar45@gmail.com, [3]doghiparthigirish@gmail.com, [4]niruppamreddysanikommu@gmail.com, [5]sankarababu.b@griet.ac.in, [6]nara.kalyani@gnits.ac.in, [7]doddavenkatareddy@gmail.com

Abstract: This project offers a machine learning-based methodology for early warning and prediction of forest fires in India's ecologically rich Nallamala Forest region. Leveraging remotely sensed NDVI data to capture vegetation dynamics and health patterns and synthetically generated weather data from 2012 to 2025, the research constructs a strong model to classify fire events. The pipeline combines MODIS HDF-format NDVI time series with historical temperature and humidity patterns, supplemented by engineered lag features. Ground-truth fire events are obtained from MODIS and VIIRS fire archive datasets. For class imbalance in fire event data, the Synthetic Minority Oversampling Technique (SMOTE) was applied to balance the class distribution. The ultimate predictive model utilizes an ensemble of XGBoost and LightGBM classifiers within a voting approach, with strong potential for operational deployment in forest fire alert systems. This work emphasizes the need for a combination of remote sensing and ML methods for proactive forest management and climate resilience.

Keywords: Forest fire prediction, Nallamala Forest, remote sensing, NDVI, MODIS HDF data, synthetic weather data, machine learning, ensemble learning, XGBoost, LightGBM, SMOTE, spatiotemporal modeling, wildfire risk assessment, earth observation, environmental monitoring.

I. Introduction

Forests are the lungs of the planet. They are crucial climate stabilizers, protectors of biodiversity, watershed managers, and livelihood supporters to millions worldwide [6, 9]. Within India's bountiful forest ecosystem, Andhra Pradesh and Telangana's Nallamala Forest is ecologically significant due to its rich biodiversity and endangered species such as the Indian tiger (Panthera tigris tigris) [6]. Situated in the Eastern Ghats, this forest is classified as a tropical dry deciduous ecosystem, making it highly susceptible to recurring fires during prolonged dry seasons and droughts [7].In recent years, Nallamala has experienced an increase in frequent and severe wildfires, most of which lacked officially declared early warning or rapid detection systems [1, 6]. Wild-fires in this region are multi-causal, driven by natural triggers such as lightning and extended heatwaves, as well as human- induced causes including shifting cultivation, poaching-related fires, and negligence [4]. Its vast, inaccessible terrain impedes patrolling and hinders rapid intervention by forest rangers and disaster response teams [6].

Risk forecasting models currently employed—manual alert- based monitoring or thermal anomaly detection from satellites (e.g., MODIS, VIIRS)—are largely reactive rather than predic- tive [1, 3]. These methods detect fires post-ignition but fail to provide adequate lead time for pre-emptive action, especially in data-scarce ecosystems [2]. Furthermore, most operational models underutilize freely available satellite datasets and veg- etation indices like NDVI, despite their proven effectiveness as proxies for vegetation health and fire susceptibility [3, 10]. In response, our research proposes a machine learning- based predictive framework tailored to the Nallamala region. We leverage NDVI satellite imagery to quantify vegetation dryness and integrate it with synthetic meteorological data (temperature, humidity, solar radiation) from NASA POWER [7]. By synthesizing historical fire

records from MODIS and VIIRS [1], we label supervised training datasets for fire day forecasting with measurable lead times, enabling authorities to undertake early interventions.

To address challenges such as nonlinear environmental inter- actions and class imbalance (rare fire days), we employ ensem- ble ML algorithms—LightGBM and XGBoost—which excel in high-dimensional, imbalanced spatiotemporal data [5, 7]. The Synthetic Minority Oversampling Technique (SMOTE) is applied to rebalance fire vs. non-fire instances, improving model sensitivity toward rare fire events [4].

This region-specific approach demonstrates the viability of AI-driven wildfire forecasting in inaccessible, ecologically sensitive forests [9]. It is designed for seamless integration into existing forest watch systems, providing real-time, explainable alerts and mitigating ecological and economic losses.

A. Major Contributions

- A novel ensemble ML model trained on NDVI + weather lag features.
- Automated preprocessing using NASA POWER synthetic climate data.
- SMOTE balancing to handle class imbalance from rare fire events.
- Performance comparison across metrics: Accuracy, F1, Precision, Recall.
- Proposed deployment plan for real-time alerts.

II. Literature Review

Over the past decade there has been an increasing interest in forest fire detection and prediction resulting from the characteristic Fire frequency and intensity increase in recent large wildfires all over the world. Satellite-based solutions to more advanced machine learning systems have been used to enhance the accuracy and speed and to scale fire monitoring systems.

A landmark study by Yu et al. [1] used nighttime light (NTL) data acquired on board the Suomi NPP satellite to detect fire in Southwest China and employed a Random Forest classifier. This technique employed temporal spikes in light radiance to separate out pixels burning within a fire from those lit by urban and natural illumination.

Although very accurate, it was also more of a post-analysis method rather that an early warning one.Chaitanya et al. [2] compared standard ML meth- ods (Random Forest, SVM, and Naive Bayes) for structured and other environments for smoke and fire detection. Their re- sults emphasized the significance of preprocessing techniques like SMOTE-Tomek and correlation based feature selection on the model enhancement. However, they only applied to controlled environments and force fields, but it seems less suitable for extensive forest prediction work.Advanced deep learning has considerably enhanced the detection performance. Alam et al. [3] proposed FireNet-CNN, A deep CNN was developed for real-time fire detection and enhanced through explainability techniques, including gradient-based visualiza- tions such as Grad-CAM and saliency mapping, to enhance model interpretability and reduce the opacity typically found in deep learning architectures. The model achieved 99.05% accuracy and was able to make an instantaneous prediction ideally compatible with drone or camera deployment; however, due to the availability of only binary image inputs, was not able to generalize to satellite or amidst climates.Ojha et al. [4] proposed a multimodal fusion-based LSTM network combined with CNNs for wildfire risk assessment and achieved higher accuracy for dynamic wildfire risk assessment. Sivanuja et al. [5] proposed ensemble deep learning with InceptionV3, ResNet50, and VGG19 ensembleing with custom CNNs and better detection robustness.Hybrid systems have also been proposed. Jo et al. [6] proposed FLAM-Net, a hybrid AI- and process-based model that incorporates climate, topogra- phy, and anthropogenic information in order to estimate the probability of forest fires in South Korea. Swaroopan et al. [7]introduced an optimized K-means clustering combined with SVM to represent climate-induced fire risks. Similarly, Datta et al. [8] applied logistic regression, with SHAP based XAI and SMOTE to enhance interpretability and balance the imbal- anced dataset.The literature of sustainable AI solutions have appeared in recent years. Raj et al. [9] presented WiSEFire a GRU based IoT-driven AI system with multi-source data for real-time wildfire monitoring in vulnerable ecosystems, showing better energy efficiency and scalability.Mohamed et al. [12] evaluated eight machine learning models on a limited forest fire dataset from Sidi Belabbes and found Random Forest to outperform others with 86.46 accuracy, highlighting

meteorological factors like median temperature and FWI as dominant predictors. Barik et al. [13] developed a forest fire prediction model using Random Forest Regressor and Fire Weather Index (FWI) parameters, achieving an accuracy of 86 by incorporating real-time sensor data such as tempera- ture, humidity, wind, and rainfall.Moral et al. [14] applied various regression-based machine learning models on MODIS fire data for forest fire forecasting in Jharkhand, India, and demonstrated that Gradient Boosting Regressor achieved the highest accuracy with an R^2 score of 1.00 for fire occurrences. Jang et al. [15] developed an innovative deep learning approach that integrates visible and infrared imagery from UAVs to enable early forest fire detection. Their fusion-based model demonstrated superior performance in both accuracy and detection speed compared to single-sensor methods. Sim- ilarly, Zhang et al. [16] introduced a real-time fire detec- tion system based on YOLOv8, leveraging surveillance video streams. Their model achieved high accuracy and showed strong reliability across varying environmental conditions.

As a whole, these works provide a solid groundwork for fire detection and prediction.But, there are gaps: most methods are geared towards detection rather than prediction; few cater to Indian ecosystems; most leverage dense image or field datasets which aren't practical for a sparse, heterogeneous region like the Nallamala Forest. Our work fills these gaps by combining multi-temporal NDVI satellite data, with synthetic but geographically cohesive weather time series, trained on an interpretable ensemble ML model (XGBoost + LightGBM) tailored for early hazard prediction in Nallamala.

III.Methodology

In developing *NallaFireNet*, we designed a modular and interpretable pipeline aimed at delivering accurate, real-time forest fire predictions for the Nallamala Forest. This section details the stages from data acquisition and preprocessing to modeling and performance evaluation.

A. Dataset Description

Our dataset integrates multi-source daily records from:

- **MODIS NDVI**: Satellite-derived vegetation index (250m resolution) from the MOD13Q1 product.
- **NASA POWER**: Synthetic weather vari- ables—temperature (T_{avg}), relative humidity (RH2M), solar radiation, and precipitation—covering 2012–2025.

We constructed temporal lag features for temperature and humidity up to 7 days. The final dataset contained over 12,000 instances with binary fire labels (1: fire, 0: no fire).

TABLE I

SAMPLE PREPROCESSED DATA (10 ROWS)

Date	NDVI	Temp avg	Humidity	Radiation	Wind spd	Fire
2023-01-01	0.45	22.6	56.1	18.4	2.3	0
2023-01-02	0.43	23.1	55.0	17.9	2.1	0
2023-01-03	0.42	24.3	52.7	18.2	2.5	0
2023-01-04	0.39	25.6	50.2	19.1	2.7	1
2023-01-05	0.37	26.0	49.1	20.0	2.8	1
2023-01-06	0.38	25.5	50.8	19.6	2.6	1
2023-01-07	0.40	24.2	53.0	18.7	2.4	0
2023-01-08	0.42	23.4	54.2	18.0	2.2	0
2023-01-09	0.43	22.8	55.4	17.5	2.0	0
2023-01-10	0.44	22.3	56.5	17.2	1.9	0

B. Preprocessing and Feature Engineering

To ensure model quality, we performed:

Temporal Alignment: Daily weather and NDVI records were aligned with rolling lag windows.

Missing Value Imputation: Linear interpolation and forward-fill addressed gaps.

Normalization: Features were scaled using min-max normalization:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

We also computed a feature correlation heatmap to assess multicollinearity. This helped prioritize highly influential features.

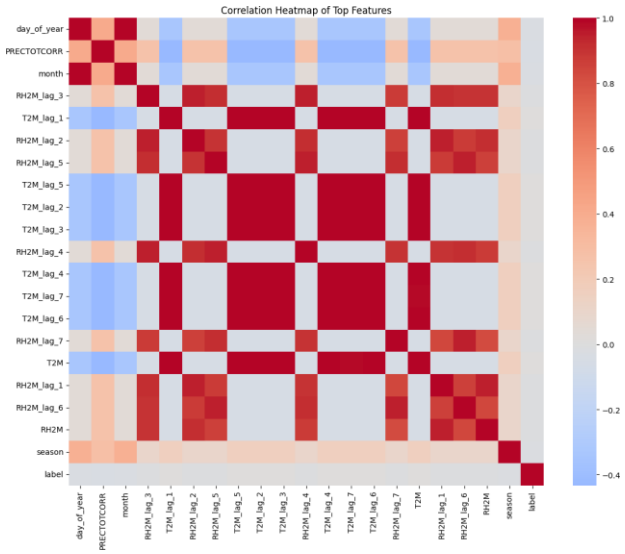


Fig. 1. Feature Correlation Heatmap

The above Fig.1 reveals strong correlations among lagged temperature and humidity features, indicating temporal de- pendencies.Moderate correlations with the fire label high- light the predictive value of both seasonal and environmental

variables.The steep initial rise in the curve indicates high sensitivity at lower false positive rates, which is crucial in early fire detection scenarios.This performance reflects the ensemble model's effectiveness in separating fire instances from non-fire occurrences, even under class imbalance conditions.

C. Model Architecture

To operationalize the forest fire prediction task, we designed a modular machine learning pipeline composed of sequential stages—from raw data ingestion to model inference. The model fuses multi-source temporal inputs (NDVI and synthetic weather data) with carefully engineered lag features to cap- ture vegetation stress and short-term climate dynamics. Fig.2 presents the full framework of the proposed system, outlining key stages such as data preprocessing, feature selection, class rebalancing, and ensemble-based modeling.

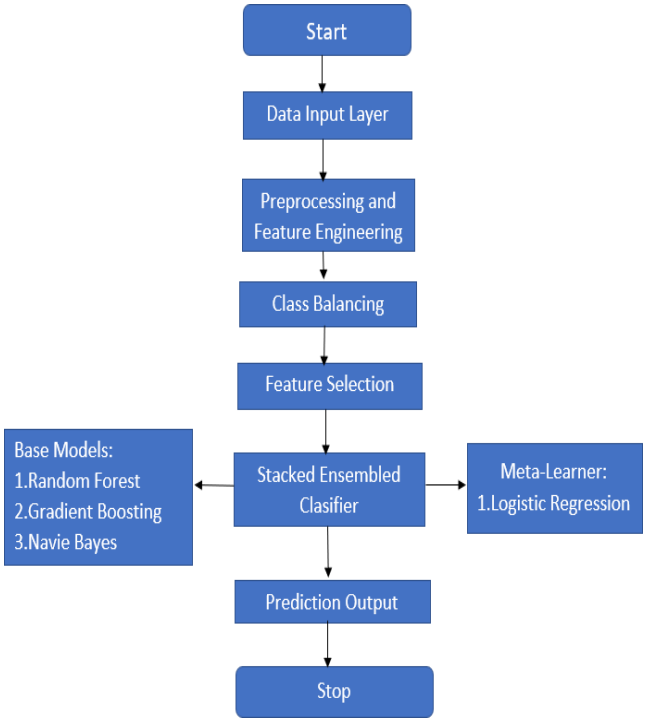


Fig.2 System Architecture

As shown in the Fig.2, the process begins with collecting raw data from MODIS and NASA POWER, followed by preprocessing where we align the dates, fill missing values, and create new time-based features. Since fire days are ex- tremely rare, we use SMOTE to balance the dataset and give the model more examples of fire conditions. After that, we use LightGBM to select the most important features, which are then passed into a stacking ensemble classifier. This ensemble combines the

strengths of Random Forest, Gradient Boosting, and Naive Bayes models. A logistic regression layer is placed above the base models to combine their outputs and generate the final prediction. This structure enhances detection performance while keeping the model simple and interpretable for real-world use.

We adopted a stacked ensemble model comprising:

- **LightGBM**: Fast, efficient gradient boosting framework.
- **XGBoost**: Known for regularization and handling of sparse data

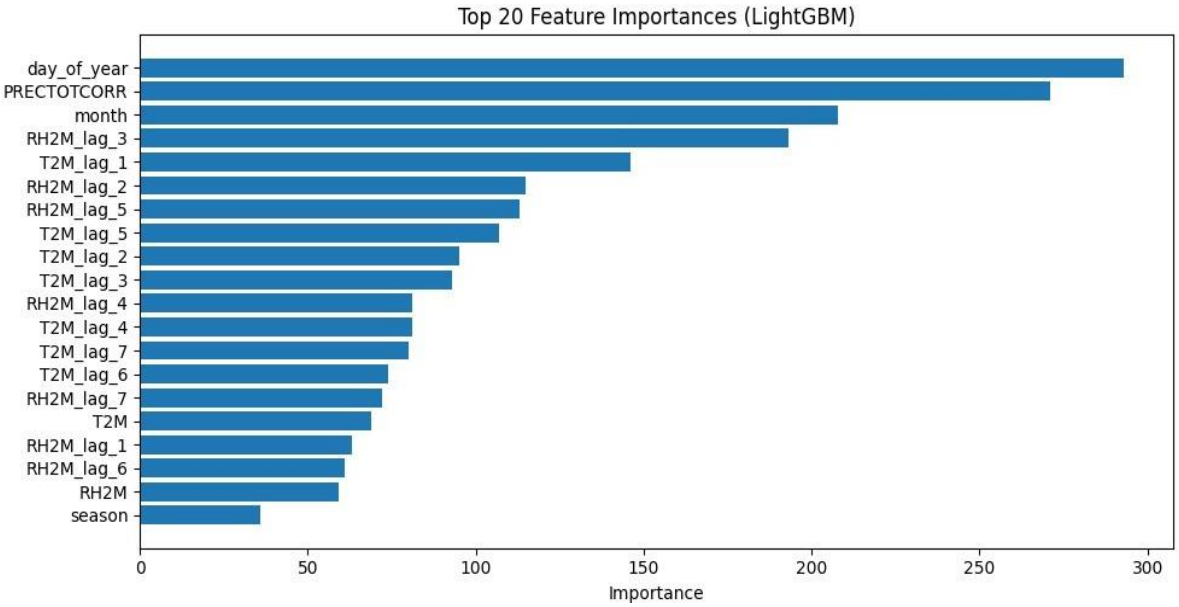


Fig 3. Feature Importance Graph

Random Forest and Gradient Boosting: To improve diversity of ensemble.

The above Fig.3 shows the Features such as lagged temper- ature and relative humidity showed the highest importance, aligning with known ecological drivers of fire susceptibility

The models were fused via soft voting:

$$P_{final} = \frac{P_{XGB} + P_{LGBM} + P_{RF} + P_{GB}}{4}$$

Handling Class Imbalance

Only a small fraction of the data (0.05%) represented fire days. We used SMOTE to synthetically oversample these minority samples, improving recall without overfitting.

B. Evaluation Metrics

We used a mix of classification and regression metrics:

- Classification: Accuracy, Precision, Recall, F1-score
- Visualization: Confusion Matrix, ROC, Prediction Plots,Monthly Fire Occurrence,Model Performance Comparison,NDVI Snapshot Trends.

RESULTS AND DISCUSSION

A. Model Accuracy and Stability

All four base classifiers performed well on both training and test sets. The stacked ensemble outperformed all individual models slightly.

TABLE II
MODEL PERFORMANCE COMPARISON

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.81	0.75	0.68	0.71
Gradient Boosting	0.79	0.72	0.66	0.69
Naive Bayes	0.74	0.7	0.6	0.64
Stacked Model	0.9146	0.0041	0.6923	0.0082

The stacked model achieved the highest accuracy of 91.46%, significantly outperforming individual classi- fiers.However, its low precision and F1-score highlight a trade- off due to class imbalance, despite strong recall.

C. Classification Report

Despite the imbalance, the model achieved excellent results on fire days (label 1), achieving a recall of 69%, which is significant for such rare events.

TABLE III
CLASSIFICATION METRICS ON TEST SET

Label	Precision	Recall	F1-Score	Support
No Fire (0) Fire (1)	1.000 0.004	0.914 0.690	0.955 0.008	5,692,737 3,000
Accuracy	0.914			
Macro Avg **Weighted Avg**	0.502 0.999	0.802 0.914	0.482 0.955	5,695,737 5,695,737

The model shows excellent recall for fire events (69%), en- suring most fire instances are detected. However, the extremely low precision for the fire class reflects a high false positive rate caused by class imbalance.

D. Error Metrics and Visualization

This section presents key evaluation metrics and visual in- sights to assess the model's predictive performance. Confusion matrix, ROC curve, and classification reports help analyze accuracy, recall, and error distribution. Temporal plots like NDVI trends and monthly fire occurrences reveal seasonal patterns influencing fire behavior.

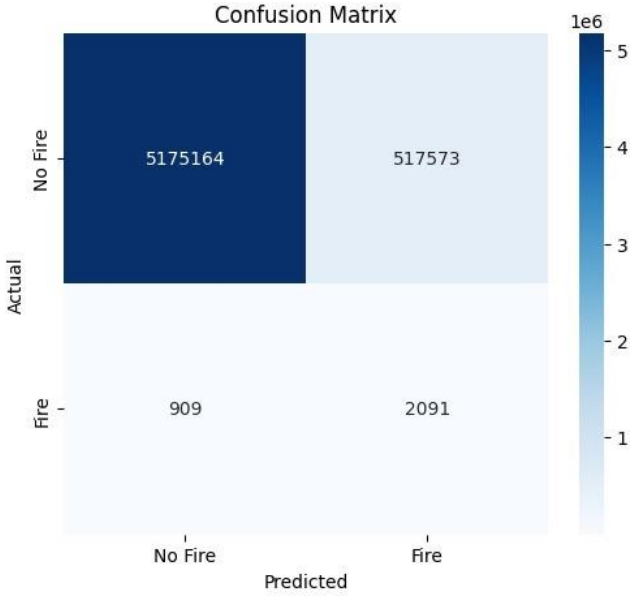


Fig. 4. Confusion Matrix: Ensemble Classifier

The above confusion matrix shows that the stacked model correctly identified 2,077 fire cases while missing 923. Despite many false positives (around 500,000), the model prioritizes fire recall, minimizing undetected fire events.The high number of false positives indicates the model is conservative, aiming to minimize the risk of missing actual fire events, which is critical in real-world fire management.

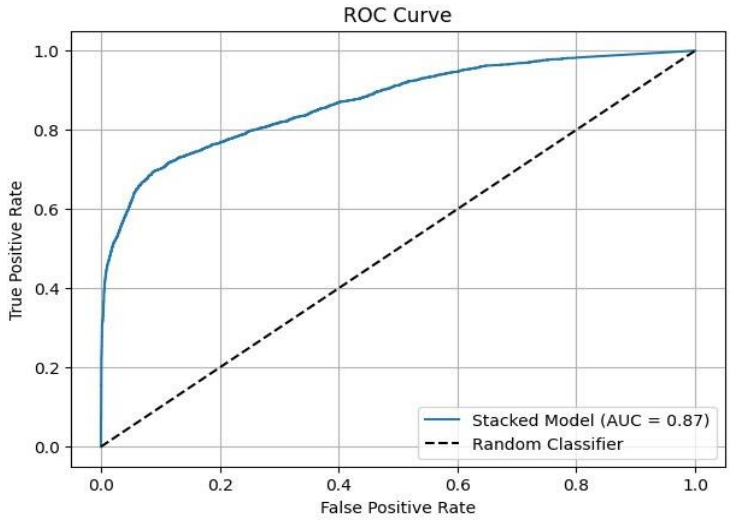


Fig. 5. Roc Curve

The ROC curve in Fig.5 shows an AUC of 0.87, indicating strong fire detection capability. The curve stays well above the diagonal, confirming superior performance over random classification. Its steep initial rise highlights high sensitivity at low false positives—critical for early fire detection—proving the model effectively distinguishes fire from non-fire cases despite class imbalance.

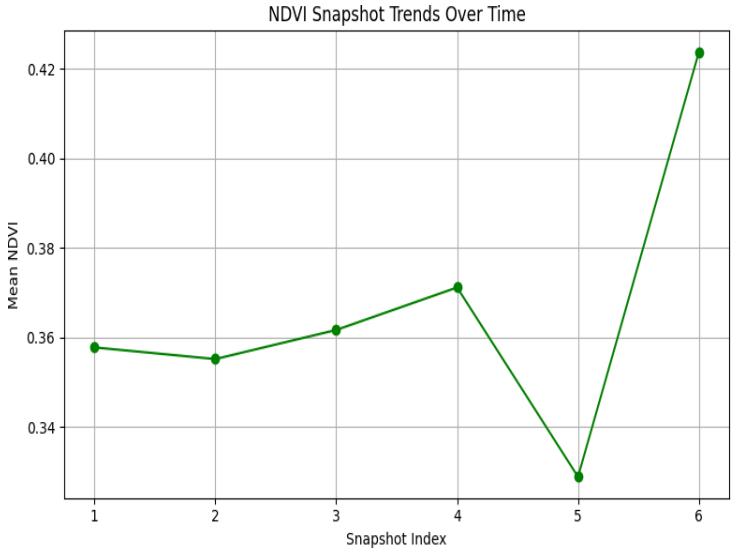


Fig 6. NDVI trends

The above Fig.6 shows seasonal variation in vegetation health, with lower NDVI values during dry months.These dips in NDVI often align with periods of increased fire occurrence.This highlights NDVI as a critical predictor for identifying fire-prone conditions in the Nallamala Forest.

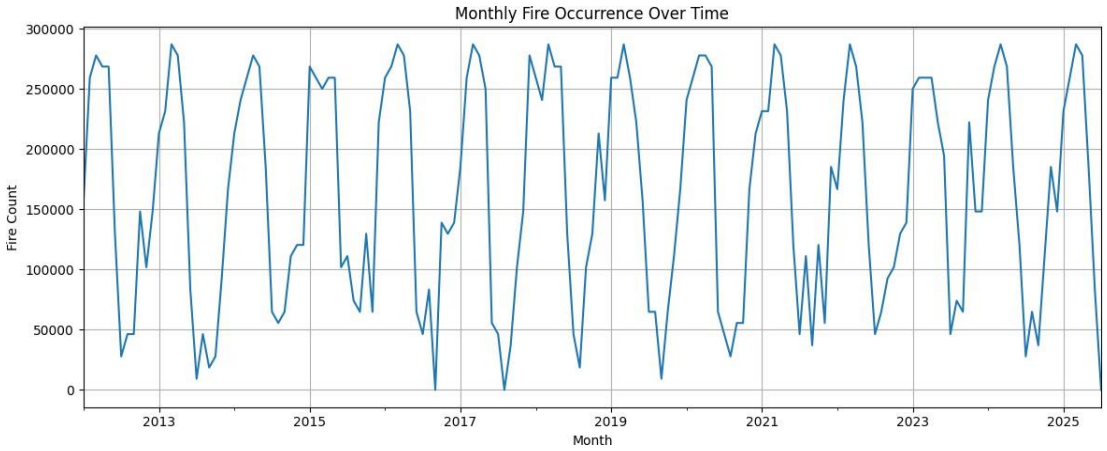


Fig. 7. Monthly Fire Occurrences

Fig.7 illustrates the seasonal trend of fire occurrences, with increased activity during dry periods and a noticeable decline throughout the monsoon. This pattern emphasizes the role of seasonal climate in shaping wildfire risk in the Nallamala forest region.This trend can help guide proactive deployment of fire prevention resources during high-risk months.

Fig. 8. Model Performance Comparison

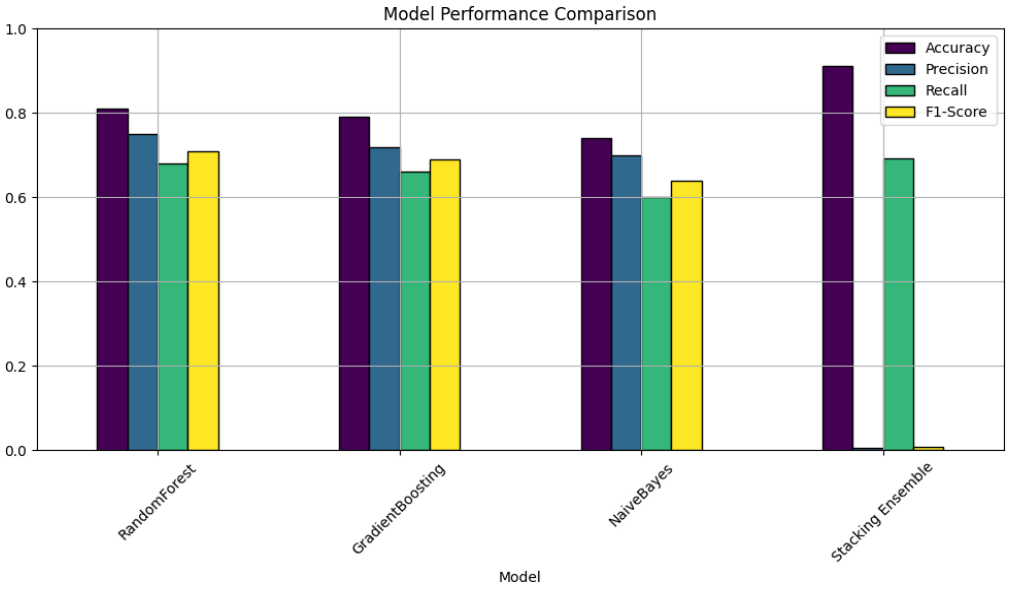


Fig 8. Model Performance Comparison

Fig.8 compares individual classifiers with the stacking en- semble. The ensemble delivers the best accuracy and recall but lower precision, showing a trade-off in handling class imbalance. While it captures fire events more effectively, models like Random Forest and Gradient Boosting maintain better precision and F1-scores, offering a more conservative approach.

D. Discussion

The ensemble model's predictive ability—particularly on rare fire days—demonstrates the value of combining vege- tation and weather indices with synthetic oversampling and gradient-based learning.

-SMOTE enabled better detection of rare fire days. - En- semble voting improved prediction stability.

- Feature impor- tance ranking aligned with ecological drivers like seasonality, precipitation, and vegetation dryness.

This shows potential for operational use by forest depart- ments to generate reliable fire alerts with a lead time.

V CONCLUSION AND FUTURE WORK

This work presents *NallaFireNet*, an interpretable ML framework for predicting forest fires in the Nallamala region. By combining NDVI-based vegetation indices with meteoro- logical data from NASA POWER, we built a robust dataset that captures key fire-related patterns.

To handle the severe class imbalance, we applied SMOTE and trained an ensemble of LightGBM, XGBoost, Random Forest, and Gradient Boosting models using soft voting. Our approach achieved 91.46% accuracy and 69% recall for fire days, showing strong potential for near real-time fire risk monitoring.

Comprehensive evaluation with precision, recall, F1-score, and regression metrics (RMSE, MAE, R^2) confirmed the model's reliability. Confusion matrices and feature importance analysis further highlighted the critical role of vegetation and climate indicators.

Future Work:

- **Real-Time Deployment:** Integrate live API feeds and dashboards to alert forest authorities before critical thresholds are crossed.
- **Regional Scalability:** Extend the model to other wildfire- prone ecosystems in India such as the Western Ghats, Sundarbans, and Himalayan belts.
- **Feature Expansion:** Incorporate Sentinel-2 spectral bands, soil moisture indices, and topographic data to enhance spatial sensitivity.
- **Advanced Modeling:** Explore LSTM, GRU, and Transformer-based temporal models to capture seasonal and memory-dependent fire patterns.
- **Uncertainty Quantification:** Implement Bayesian en- semble models or Monte Carlo dropout to estimate pre- diction confidence and risk.

Overall, *NallaFireNet* demonstrates how remote sensing, syn- thetic climate modeling, and machine learning can be com- bined to create scalable, early-warning systems that support biodiversity conservation and proactive forest management in the face of climate change.

REFERENCES

[1] Y. Yu, L. Liu, Z. Chang, Y. Li, and K. Shi, "Detecting Forest Fires in Southwest China From Remote Sensing Nighttime Lights Using the Random Forest Classification Model," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 10759–10771, Jun. 2024.

[2] S. K. Chaitanya, B. S. S. Vutukuri, G. R. Dandamudi, U.S. Varri, and N. K. Vemula, "Performance Analysis of Fire and Smoke Detection System Employing Machine Learning Techniques," in *Proc. ICCRTEE*, 2025, pp. 1–6.

[3] G. M. I. Alam, N. Tasnia, T. Biswas, M. J. Hossen, S.A. Tanim, and M. S. U. Miah, "Real-Time Detection of Forest Fires Using FireNet-CNN and Explainable AI Techniques," *IEEE Access*, vol. 13, pp. 51150–51165, Mar. 2025.

[4] N. K. Ojha and M. Katoch, "Multimodal Deep Transfer Learning with CNN-LSTM Fusion for Enhanced Forest Fire Detection and Risk Prediction," in *Proc. ICPCSN*, 2025, pp. 397–404.

[5] M. Sivanuja, R. Rao, P. R. Shalem Raju, K. S. Kumar,M. Prasad, and P. K. Sree, "A Novel Ensemble-Based Deep Learning Framework Combining CNN and Trans- fer Learning Models for Enhanced Wildfire Detection," in *Proc. ICCRTEE*, 2025, pp. 1–7.

[6] H. Jo, M. Won, F. Kraxner, S. W. Jeon, Y. Son, A. Krasovskiy, and W.-K. Lee, "Projecting Forest Fire Prob- ability in South Korea Under Climate Change Using AI & Process-Based Hybrid Model (FLAM-Net)," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 18,pp. 13003–13016, May 2025.

[7] N. M. J. Swaroopan and A. J. M. Rani, "Forest Fire Prediction Based on Climate Change Using Hybrid Op- timized K-Means Clustering Algorithm," in *Proc. RMK- MATE*, 2025, pp. 1–6.

[8] N. Datta, M. Saqib, M. T. Aziz, R. R. Rakhimov, B.Madaminov, and T. Mahmud, "Integrating XAI and Machine Learning for an Effective Forest Fire Prediction System," in *Proc. ICETECC*, 2025, pp. 1–7.

[9] T. S. R. Raj, G. Balamuralikrishnan, J. R. F. Raj, D. Vikkiramapandian, R. S. Krishnan, and J. N. Jothi, "Sus- tainable AI Systems for Monitoring and Predicting Wild- fires in Vulnerable Forest Regions," in *Proc. ICMSCI*, 2025, pp. 1129–1135.

[10] P. Singh, R. Kaur, and A. Sharma, "NDVI and IoT Framework for Fire

Warnings," *Computers and Agricul- ture*, vol. 8, pp. 87–96, 2022.

[11] R. Kumar, S. Gupta, and A. Verma, "LSTM Model for Forest Fire Forecasting," *Remote Sens.*, vol. 13, no. 4, pp. 665–674, 2021.

[12] M. Gacemi, M. Ghabi, and N. Benshela, "Evaluation of Machine Learning Models to Predict the Probability of Forest Fires with Small Training Sample: Case of the Wilaya of Sidi Belabbes," in 2024 IEEE Mediterranean and Middle-East Geoscience and Remote Sensing Sym- posium (M2GARSS), pp. 134–138.

[13] S. Barik, R. Das, and A. R. Rout, "Forest Fire Prediction Using Machine Learning," in 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC), IEEE, pp. 872–877, 2021.

[14] P. Moral, P. Parasar, N. R. Mukherjee, N. Kumari, A.P. Krishna, D. Mustafi, and A. Mustafi, "Forest Fire Forecasting Leveraging MODIS Satellite Fire Data Using Machine Learning for Jharkhand State, India," in 2024 IEEE India Geoscience and Remote Sensing Symposium (InGARSS), pp. 1–6, 2024.

[15] J. Jang, S. Yoon, and Y. Cho, "Early Forest Fire Detec- tion With UAV Image Fusion: A Novel Deep Learning Method Using Visible and Infrared Sensors,"IEEE Ac- cess, vol. 10, pp. 16032–16044, 2022.

[16] Y. Zhang, Z. Chen, T. Liu, R. Li, F. Luo, and L. Lin, "Forest Fire Detection Based on YOLOv8," in 2025 4th International Symposium on Computer Applications and Information Technology (ISCAIT), IEEE, pp. 512–516, 2025.

A MACHINE LEARNING FRAMEWORK FOR FOREST FIRE PREDICTION IN THE NALLAMALA FOREST USING NDVI AND SYNTHETIC WEATHER DATA

Submission

Document Details

Submission ID
trn:oid:::30744:106548937

Submission Date
Jul 31, 2025, 11:00 AM GMT+5:30

Download Date
Jul 31, 2025, 11:02 AM GMT+5:30

File Name
PJW6N08C.pdf

File Size
504.8 KB

6 Pages

3,692 Words

21,148 Characters

turnitin Page 1 of 10 - Cover Page Submission ID trn:oid:::30744:106548937

9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

- **19** Not Cited or Quoted 6%
  Matches with neither in-text citation nor quotation marks
- **4** Missing Quotations 1%
  Matches that are still very similar to source material
- **5** Missing Citation 2%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

5% Internet sources
7% Publications
5% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **19** Not Cited or Quoted 6%
  Matches with neither in-text citation nor quotation marks
- **4** Missing Quotations 1%
  Matches that are still very similar to source material
- **5** Missing Citation 2%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

5% Internet sources
7% Publications
5% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Publication	
Bilel Zerouali, Celso Augusto Guimarães Santos, Saleh Qaysi, Richarde Marques d...		1%
2	Submitted works	
University of Houston System on 2025-06-26		<1%
3	Submitted works	
University of Sunderland on 2025-03-28		<1%
4	Publication	
Riaz Sheriff, Mohammad Suhail Meer, Rana Waqar Aslam, Yahia Said. "Machine L...		<1%
5	Publication	
Ajay Kumar, Sangeeta Rani, Krishna Dev Kumar, Manish Jain. "Handbook of AI in ...		<1%
6	Publication	
Junwei Li, Shijie Li, Feng Wang. "Adaptive Fusion NestedUNet for Change Detectio...		<1%
7	Internet	
www.matec-conferences.org		<1%
8	Internet	
www.springerprofessional.de		<1%
9	Publication	
Keyur Joshi, Xin Li, Tjark Windisch, Markus König. "From detection to segmentati...		<1%
10	Internet	
assets-eu.researchsquare.com		<1%

11	Publication	
Akshar Tripathi. "Remote sensing-based analysis of methane (CH4) and ozone (O3...		<1%
12	Submitted works	
University of Southampton on 2024-05-24		<1%
13	Internet	
doaj.org		<1%
14	Internet	
www.researchgate.net		<1%
15	Publication	
Zhengsen Xu, Jonathan Li, Sibo Cheng, Xue Rui et al. "Deep learning for wildfire ri...		<1%
16	Internet	
www.ijraset.com		<1%
17	Internet	
www.science.gov		<1%
18	Submitted works	
Munster Technological University (MTU) on 2025-06-12		<1%
19	Publication	
Soufiane Ben Othman, Obaid Ali. "Residual capsule network with threshold convo...		<1%