FOREST FIRE PREDICTION USING MACHINE LEARNING

Major project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

S Jyothirmai	(18UECS0781)	**(12383)**
R Thanushri	(18UECS0853)	**(11253)**
D Chidanand	(18UECS0243)	**(12957)**

Under the guidance of
Dr.N.Malarvizhi M.E,Ph.D
PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A Grade
CHENNAI 600 062, TAMILNADU, INDIA

June 2022

CERTIFICATE

It is certified that the work contained in the project report titled FOREST FIRE DETECTION USING MACHINE LEARNING by S Jyothirmai (18UECS0781), R Thanushri (18UECS0853)
D Chidanand (18UECS0243) has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
Dr.N.Malarvizhi M.E.,Ph.D.,
Professor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr.Sagunthala R&D
Institute of Science & Technology
June,2022

Signature of Head of the Department
Dr. V. Srinivasa Rao
Professor & Head
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr.Sagunthala R&D
Institute of Science & Technology
June,2022

DECLARATION

We declare that this written submission represents my ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)
S JYOTHIRMAI
Date: / /

(Signature)
R THANUSHRI
Date: / /

(Signature)
D CHIDANAND
Date: / /

APPROVAL SHEET

This project report entitled FOREST FIRE DETECTION USING MACHINE LEARNING by S.Jyothirmai (18UECS0781), R.Thanushri (18UECS0853),D.Chidanand(18UECS0243) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners **Supervisor**

Dr.N.Malarvizhi M.E.,Ph.D

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Dean Computer Science & Engineering Dr.V.SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We also take this opportunity to express a deep sense of gratitude to our Internal Supervisor **Dr.N.Malarvizhi, M.E., Ph.D.,** for her cordial support, valuable information and guidance, he/she helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. C. SHYAMALA KUMARI, M.E., Ms.S.FLORENCE, M.Tech.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

S.Jyothirmai (18UECS0781)
R.Thanushri (18UECS0853)
D.Chidanand (18UECS0243)

ABSTRACT

Forest Fire Prediction is a key component of forest fire control.This is a major environmental problem that creates ecological destruction in the form of a threatened landscape of natural resources that disrupts the stability of the ecosystem, increases the risk for other natural hazards, and decreases resources such as water that causes global warming and water pollution. Fire Prediction is a key element for controlling such incidents. Prediction of forest fire is expected to reduce the impact of forest fire in the future. Many fire detection algorithms are available with different approach towards the detection of fire. In the existing work processes the fire affected region is predicted based on the satellite images. To predict the occurrences of a forest fire the proposed system processes using the meteorological parameters such as temperature, rain, wind and humidity were used. Nowadays, there are various technologies for fire modelling to predict the spread of fires, such as physical models and mathematical models .These models depend on data collection during forest fires, simulation, and lab experiments to specify and predict fire growth in many regions. Recently, simulation tools have been used to predict forest fires, but simulation tools faced some problems such as the accuracy of input data and simulation tool execution time. Machine learning is a sub-branch of Artificial Intelligence to learn computers aspect. Machine learning can be divided into two classes: supervised, unsupervised and reinforcement. In supervised learning, a supervisor is existed to give insights to the learning algorithm on how a decision or an action is bad or good. In supervised learning, the whole the data set is labelled completely. Supervised machine learning algorithms are as linear regression, Support vector machine, Artificial neural networks and decision trees. In unsupervised learning, the data set is not labelled. This leads that the algorithm must define the labels. The structure of the data set and the relationship between the features will be learned by the algorithm. Unsupervised machine learning algorithms are as k-means clustering and Self-organizing map. In reinforcement learning, the learning algorithm gets punished in case of a wrong action and gets rewarded in case of correct action.

Keywords: Fire Prediction, Global Warming, Meteorological Parameters.

LIST OF FIGURES

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

One of the most extremely occurring disasters in recent times is forest fires (wildfires). Due to these wildfires, a lot of acres of forest area are getting destroyed. The significant reasons that lead to the occurrence of forest fires are warming due to the increase in the average temperature of the earth, and human negligence. In Africa, South America, Southeast Asia, and New Zealand, forest fires occur due to human factors like husbandry of animals and agriculture. Nowadays, there are various technologies for fire modelling to predict the spread of fires, such as physical models and mathematical models. These models depend on data collection during forest fires, simulation, and lab experiments to specify and predict fire growth in many regions. Nowadays, there are various technologies for fire modelling to predict the spread of fires, such as physical models and mathematical models .These models depend on data collection during forest fires, simulation, and lab experiments to specify and predict fire growth in many regions. Recently, simulation tools have been used to predict forest fires, but simulation tools faced some problems such as the accuracy of input data and simulation tool execution time. Machine learning is a sub-branch of Artificial Intelligence to learn computers aspect. Machine learning can be divided into two classes: supervised, unsupervised and reinforcement.

1.2 Aim of the project

The main aim of our project is to predict the forest fire based on some attributes using machine learning techniques.

1.3 Project Domain

The Project domain is Machine Learning. Machine learning is a type of artificial intelligence that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.Machine learning is used in internet search engines, email filters to sort out spam, websites to make personalised recommendations, banking software to detect unusual transactions, and lots of apps on our phones such as voice recognition.

1.4 Scope of the Project

The goal of this project is to predict forest fires using machine learning algorithms. This project uses Random forest regressor to provide better accuracy. The forest fire exploration dataset will be divided into training data and test data. The training data is fed into the model for supervised learning.

Chapter 2

LITERATURE REVIEW

Martin Mueller et al [1] the two novel optical flow estimators, optimal mass transport (OMT) and Non-Smooth Data (NSD). The dynamics of fire have motivated the use of motion estimators to differentiate fire from other non-fire object. The obtained moving region provides useful space on which to define motion features. These features reliably detect fire and reject non-fire motion, on a large dataset of videos. There is a chance for false detections in the presence of significant noise, partial occlusions, and rapid angle change.

Paulo Vinicius Koerich Borges et al [2] a new identification metric based on color for fire detection in videos. Also identified important visual features of fire, like boundary roughness and skewness of the fire pixel distribution. The skewness is a very useful descriptor as the frequent occurrence of saturation in the red channel of fire regions is identified. For newscast videos, model the probability of occurrence of fire as a function of the position, yielding an efficient performance. While comparing with other methods which extract complicated features, the features discussed here allow very fast processing, making the system applicable not only for real time fire detection, but also for video retrieval in news contents, which require faster than real-time analysis.

A. Kansal et al [3] The use of regression and the division of datasets has been proposed in this paper as a method for detecting fire. The algorithm achieves a low R-squared and a low root mean square error. This method could be used for other calamities in the future. The use of specific transformations may also help to increase the model's efficiency.

Guruh Fajar Shidik et all [4] An alternative hybrid model capable of predicting the extent of forest fire has been developed in this study. The algorithm, which includes meteorological and forest weather index variables, has successfully classified the level of burning into three categories: No Burn Area, Light Burn, and Heavy Burn. The proposed model's examination revealed encouraging results in terms of accuracy. of confusion matrix around 97.50Kappa 0.961.

G. Freiha et al [5] As the human technology moved further, the risk of natural and man induced catastrophes increase exponentially. One of the most dangerous disasters is fires. In addition to its direct danger on human's lives, fire consumes forests where trees that provide humans with oxygen are destroyed. The risk of fire has increased due to the problem of global warming which appeared in the

1980s. Forest fires represent a constant threat to ecological systems, infrastructure and environmental aspects of a community. This gives rise to the urgent need to detect forest fires as fast as possible. This paper highlights the powerful feature of wireless sensor networks as a potential solution to the challenge of early detection of forest fires. The device presented makes use of various sensors attached, solar recharging mechanism, and wireless data transmission, to fulfill the task in question. These collected data are transmitted to a near central unit where they are analyzed and then uploaded to an online website which contacts the Civil Defense unit if necessary.

H. Xu et al [6] Wireless Sensor Networks can be used for many applications, such as industrial automatic control, remote environmental monitoring and target tracking. The similar system is promising applications in forest fires can make a real-time monitoring and detection. Wireless sensor network consists of numerous small nodes in most situations, which small nodes are deployed in remote and inaccessible hostile environments or over large geographical areas. The large number of small nodes sense environmental changes and report them to cluster head node over network architect, which the deployment and maintenance should be easy and scalable. A new approach for forest fire monitoring and detection was present in this paper, which it is by using data aggregation in wireless sensor network.

P. K. Singh et al [7] Fire is a threat to our forests. Human intervention is one the major cause of forest fires. In addition to destroying wooden areas fire jeopardizes our safety. The risk of forest fires has increased in Hilly around the globe in recent past years due to development and building constructions. In order to detect the forest fire several attempts have been made using different techniques from optical fire sensor, satellite-based methods and wireless sensor networks. Early detection of forest fire is most important and may save the resources and wealth of forest. In this paper, we have analyzed the existing forest fire detection techniques limited to wireless sensor networks only. Numbers of popular wireless sensor network-based forest fire detection techniques have been explored and their merits along with the demerits are reported during the findings.

J. Zhang et al [8] As we all know, the forest is considered as one of the most important and indispensable resources, the prevention and detection of the forest fire, have been researched hotly in worldwide forest fire prevention departments. Based on the deficiencies of conventional forest fire detection on real time and monitoring accuracy, the wireless sensor network technique for forest fire detection was introduced, together with satellite monitoring, aerial patrolling and manual watching, an omni-bearing and stereoscopic air and ground forest-fire detection pattern was found so that the decision for fire-extinguishing or fire prevention can be made rightly and real-timely by related government departments. A cluster-based wireless sensor network paradigm for forest fire real-time detection was put forward in this paper. Some key questions were discussed emphatically, such as the ad hoc network related technology, the node hardware designing, the forest-fire forecasting model and the propagation characteristic of UHF wireless signal and so on.

Z. Jiao et al [9] Unmanned aerial vehicles (UAVs) are increasingly being used in forest fire monitoring and detection thanks to their high mobility and ability to cover areas at different altitudes and locations with relatively lower cost. Traditional fire detection algorithms are mostly based on the RGB color model, but their speed and accuracy need further improvements. This paper proposes a forest fire detection algorithm by exploiting YOLOv3 to UAV-based aerial images. Firstly, a UAV platform for the purpose of forest fire detection is developed. Then according to the available computation power of the onboard hardware, a small-scale of convolution neural network (CNN) is implemented with the help of YOLOv3. The testing results show that the recognition rate of this algorithm is about 83 percent, and the frame rate of detection can reach more than 3.2 fps. This method has great advantages for real-time forest fire detection application using UAVs.

Chi Yuan et al [10] the two novel optical flow estimators, optimal mass transport (OMT) and Non-Smooth Data (NSD). The dynamics of fire have motivated the use of motion estimators to differentiate fire from other non-fire object. The obtained moving region provides useful space on which to define motion features. These features reliably detect fire and reject non-fire motion, on a large dataset of videos. There is a chance for false detections in the presence of significant noise, partial occlusions, and rapid angle change.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The existing system for detecting fire are smoke alarms and heat alarms. The main disadvantage of the smoke sensor alarm and heat sensor alarms are that just one module is not enough to monitor all the potential fire prone places. The only way to prevent a fire is to be cautious all the time. Even if they are installed in every nook and corner, it just is not sufficient for an efficient output consistently. As the number of smoke sensor requirement increase the cost will also increase to its multiple. The proposed system can produce consistent and highly accurate alerts within seconds of accident of the fire. It reduces cost because only one software is enough to power the entire network of surveillance. Research is active on this field by data scientists and machine learning researchers. The real challenge is to minimize the error in detection of fire and sending alerts at the right time.

3.2 Proposed System

In this project we tried to make a prediction for the burned area in Forest Area. Forest Fires Data Set was used for this analysis. The data was clusterized. Stepwise regression methods were applied to choose one best predictor. It is interesting to see, which one of them has the biggest impact on the burned area in each cluster. Fire Cast receives the same input variables as the training fires for the testing fire, starting with the first recorded fire perimeter and making next day predictions for all collected perimeters. Fire Cast randomly samples POIs surrounding the current fire perimeter and assigns a prediction value p, so that p [0, 1]. Each pixel representing the likelihood the model predicts the pixel will be contained within the following fire perimeter given the input variables. Higher values represent a higher predicted likelihood of burn. The dataset used to train the model. Additionally, its ability to generate step-by-step images of the fires progression allows for easy integration with Deep Learning models. The Fire Assessment learner is a simple model in which any tree close to the fire has the same given probability of catching fire.

3.3 Feasibility Study

A feasibility study is an analysis that takes all of a project's relevant factors into account. The feasibility study aims to objectively and rationally uncover the strengths and weaknesses of the exiting project, opportunities and dangers as displayed by the earth, the assets required to help though, and eventually the prospects for progress.
Three key considerations involved in the feasibility analysis are:
1. Economic Feasibility
2. Technical Feasibility
3. Social Feasibility

3.3.1 Economic Feasibility

Economic feasibility is a kind of cost benefit analysis of the examined project, which assesses whether it is possible to implement it. This term means the assessment and analysis of a project's potential to support the decision-making process by objectively and rationally identifying its strengths, weaknesses, opportunities and risks associated with it. For Forest fire detection the economic feasibility is cost efficient.

3.3.2 Technical Feasibility

Technical Feasibility focuses on the technical resources available. This project requires different datasets which shows up the fire detection by considering various attributes and it uses Anaconda navigator to run and detect the fire.

3.3.3 Social Feasibility

Social feasibility is a detailed study on how one interacts with others within a system or an organization. Social impact analysis is an exercise aimed at identifying and analyzing such impacts in order to understand the scale and reach of the project. Forest Fire detection is used for prior avoiding of fire in forest. It is legal in term. But as a responsible person everyone should follow social ethics of collecting of data.

3.4 System Specification

3.4.1 Hardware Specification

- Processor: Intel i3.

- Disk space: 160 GB.
- Ram: 4 GB.
- Internet Connection.

3.4.2 Software Specification

- Windows 7 or Windows or higher versions of OS
- Any Latest Web Browser. (Preferably, Mozilla Firefox, Google chrome).
- Python for User Interface.
- web based jupyter notebook.
- Anaconda Navigator.

3.4.3 Standards and Policies

- IEEE 829 - Software Test Documentation.
- IEEE 830 - Software Requirements Specifications.
- IEEE 1012 - Software verification and validation.
- IEEE 1016 - Software design description.

Chapter 4

METHODOLOGY

4.1 General Architecture

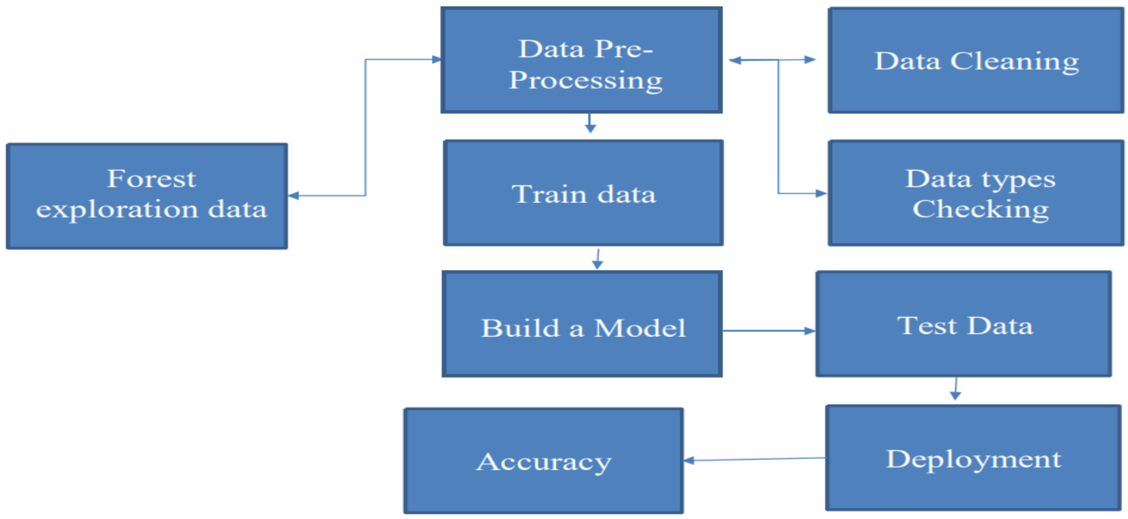

Diagram.png

Figure 4.1: **System Diagram**

In the Prediction of forest fire, the collected forest exploration data undergoes Pre-Processing for better analysis. This Pre-Processing involves data cleaning and checking of data types. And then after the Pre-Processing, the data will be trained and the model will be built based on the collected data. Then the data will be tested and the deployment takes place. In the deployment step, the data will be moved to a place where some action can be performed in it. And after that the accuracy is finded, which is the key concept for prediction.

4.2 Design Phase

4.2.1 Data Flow Diagram

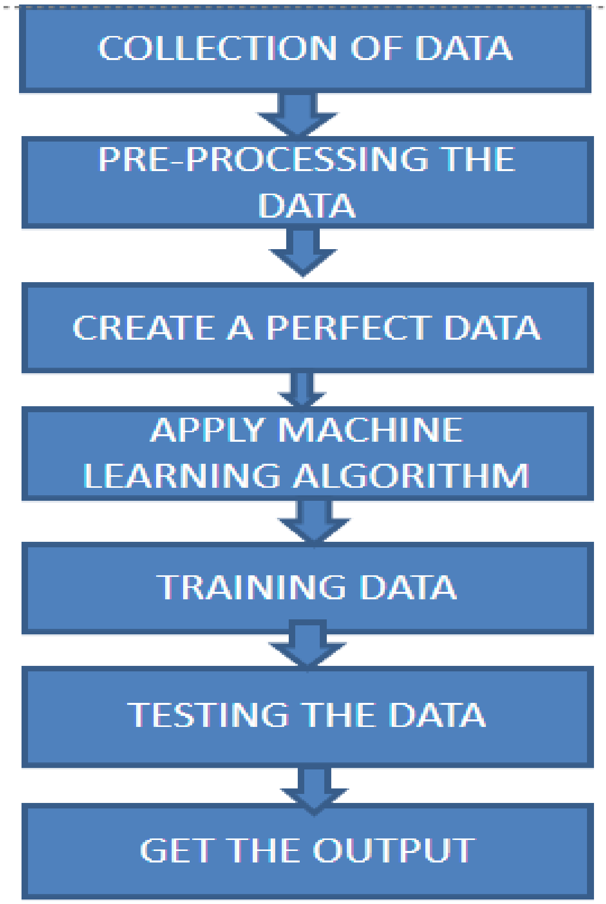

Figure 4.2: **Data Flow Diagram**

From this data flow diagram It say's that, for the Prediction of forest fire, it undergoes 7 steps. Firstly the data will be collected and the collected undergoes pre-processing step. In this step, all the collected will be separated uniformly so that the perfect data will be created. After that the Machine Learning Algorithm will be applied for the data. Then the data will be trained and after that the trained data will be tested and finally output will be generated.

4.2.2 Use Case Diagram

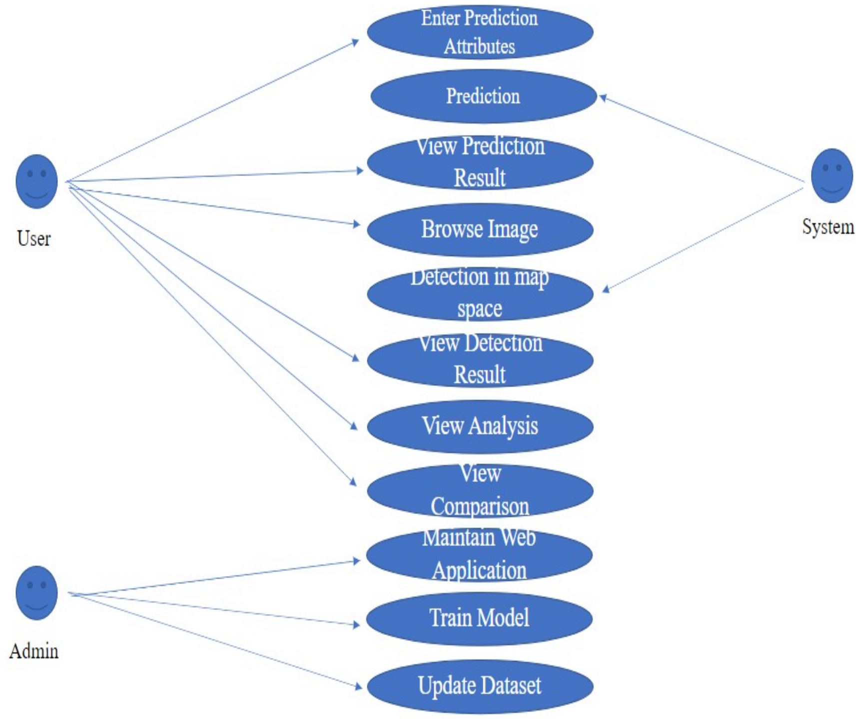

Figure 4.3: **Use Case Diagram**

Use case diagram depicts the interactions between the user inputs and user outputs. The modules and methodologies that are involved in accomplishing the better outputs and the decisions to be done.The inner procedural steps that occurs between input and output are shown in an inevitable way. The input data undergoes processing steps and pre-processing steps that should be done for filtering and cropping for future analysis that helps in differentiating the normal and abnormal traits.

4.2.3 Class Diagram

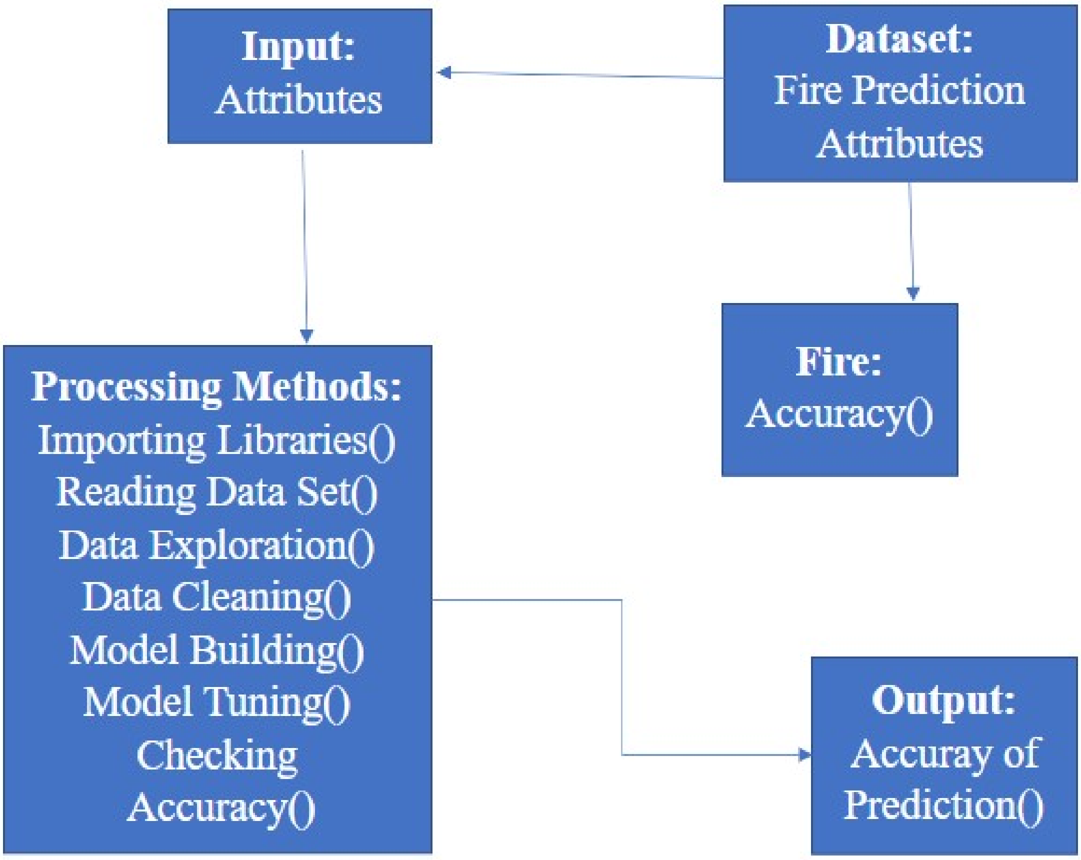

Figure 4.4: **Class Diagram**

Class Diagram of segmentation is the static structural representation of the project or an approach of solving a problem that defines the architecture by conveying the classes, sub-classes with appropriate attributes and the methods or functions presuming the relation among the objects referred. Blocks that are mentioned are having their corresponding methods for segmentation. The conceptual model of the project i.e., the class diagram helps i structuring the application or project in detailed and for translating the class model into programmable code. The classes mentioned indicates the primary components and relations among them.

4.2.4 Sequence Diagram

Figure 4.5: **Sequence Diagram**

The sequence diagram of the project refers to the sequential flow of the modules based on the time sequence. In this scenario the data translation between each task or sequence is distributed simultaneously with time. The functions that are to be completed in each module are defined in a sequence of time and these are exchanged with time management. Based on the recognition of each scenario a logical view is established. The well pack of horizontal and vertical flow of diagram shows the visualized process of segmentation involved in the project.

4.2.5 Collaboration diagram

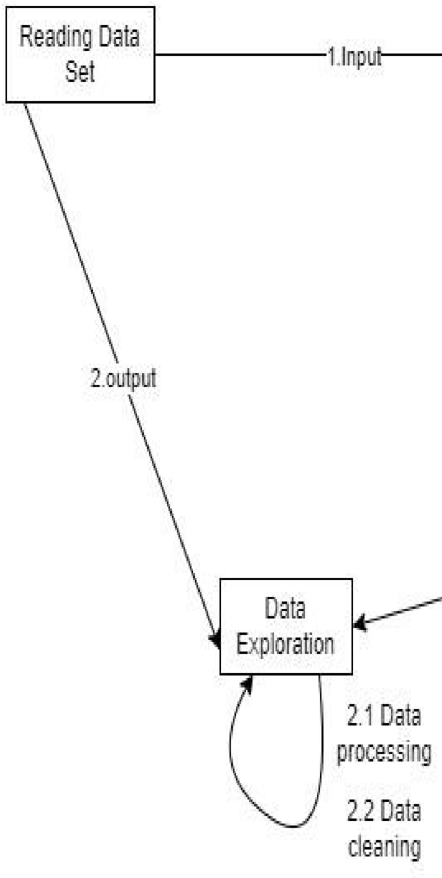

Figure 4.6: **Collaboration diagram**

The Collaboration Diagram of the project represents the relationship between classes. Collaboration represents the same as architecture of the methodology by representing the flow of the logic for better understanding rather depicts the architectural. The objects has multiple features. The connection of these multiple features with proper characteristic level of describing the system helps in communicating the proposed idea in a better way. This collaboration also called as communication Diagram. This helps in portraying the Fire Prediction of forest and the Machine Learning methods.

4.2.6 Activity Diagram

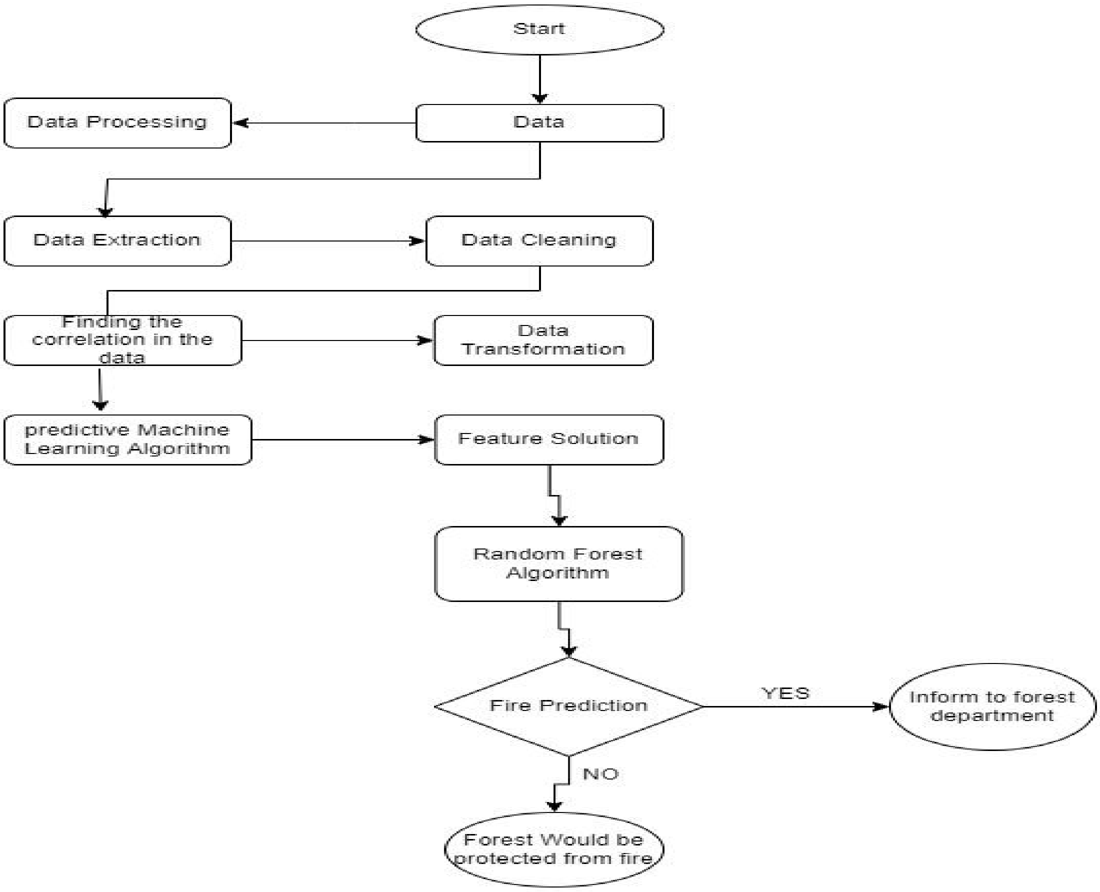

Figure 4.7: **Activity Diagram**

Activity Diagram of the project is the flowchart of the activities and represents the workflow among the multiple activities. It shows the actual process and flow of the different activities and their mechanism in the system. The execution flow from the start of the system to the end of the system is clearly depicted. The activity nodes helps in differentiating each activity from each other.

4.3 Algorithm

4.3.1 Random Forest Algorithm

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

4.4 Module Description

4.4.1 Module

In this project we tried to make a prediction for the burned area in Forest Area. Forest Fires Data Set was used for this analysis. The data was clusterized. Stepwise regression methods were applied to choose one best predictor. It is interesting to see, which one of them has the biggest impact on the burned area in each cluster. Fire Cast receives the same input variables as the training fires for the testing fire, starting with the first recorded fire perimeter and making next day predictions for all collected perimeters.Fire Cast randomly samples POIs surrounding the current fire perimeter and assigns a prediction value p, so that p [0, 1]. Each pixel representing the likelihood the model predicts the pixel will be contained within the following fire perimeter given the input variables. Higher values represent a higher predicted likelihood of burn. The dataset used to train the model. Additionally, its ability to generate step-by-step images of the fires progression allows for easy integration with Deep Learning models. The Fire Assessment learner is a simple model in which any tree close to the fire has the same given probability of catching fire.

4.5 Steps to execute/run/implement the project

4.5.1 Step1 : Importing Libraries

In this step we import/install some required libraries.

4.5.2 Step2 : Data Exploration

In this data exploration, we go through the dataset to find specific patterns which helps to understand insights and implement new policies. It checks for the null values in the forest fire prediction dataset.

4.5.3 Step3 : Data Cleaning

In this process, we prepare the data for analysis by removing or modifying the data that is incomplete, incorrect, irrelevant, duplicated, or improperly formatted.

4.5.4 Step4 : Model Modelling

In Machine Learning, the model is built by learning and generalizing from the data that is trained, and then applying the acquired knowledge to new data it has never seen before to make predictions and fulfill its purpose. Lack of data will prevent us from building the model, and access to data is not enough.

4.5.5 Step5 : Model Tuning

Model Tuning is also known as hyperparameter optimization. The hyperparameters are variables that control the training process. These are configuration variables that do not change during a Model training job. Model tuning provides optimized values for hyperparameters, which maximize the model's predictive accuracy.

4.5.6 Step6 : Checking Accuracy

Machine Learning model accuracy is the measurement used to determine which model is best at identifying relationships and patterns between variables in a dataset based on the input, or training of the data. Here the accuracy of the Trained data is 95.55 percentage and for the Tested data, the accuracy is 68.02 percentage.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

	latitude	longitude	brightness	scan	track	acq_date	acq_time	satellite	instrument	confidence	version	bright_t31	frp	daynight	type
0	-11.8070	142.0583	313.0	1.0	1.0	2019-08-01	56	Terra	MODIS	48	6.3	297.3	6.6	D	0
1	-11.7924	142.0850	319.3	1.0	1.0	2019-08-01	56	Terra	MODIS	71	6.3	297.3	11.3	D	0
2	-12.8398	132.8744	311.6	3.1	1.7	2019-08-01	57	Terra	MODIS	42	6.3	298.7	23.1	D	0
3	-14.4306	143.3035	310.1	1.1	1.1	2019-08-01	57	Terra	MODIS	33	6.3	296.1	6.5	D	0
4	-12.4953	131.4897	310.3	4.0	1.9	2019-08-01	57	Terra	MODIS	36	6.3	298.8	27.6	D	0

Figure 5.1: **Trained data**

```
latitude,longitude,brightness,scan,track,acq_date,acq_time,satellite,instrument,confidence,version,bright_t31,frp,daynight,type
-11.807,142.0583,313,1,1,2019-08-01,0056,Terra,MODIS,48,6.3,297.3,6.6,D,0
-11.7924,142.085,319.3,1,1,2019-08-01,0056,Terra,MODIS,71,6.3,297.3,11.3,D,0
-12.8398,132.8744,311.6,3.1,1.7,2019-08-01,0057,Terra,MODIS,42,6.3,298.7,23.1,D,0
-14.4306,143.3035,310.1,1.1,1.1,2019-08-01,0057,Terra,MODIS,33,6.3,296.1,6.5,D,0
-12.4953,131.4897,310.3,4,1.9,2019-08-01,0057,Terra,MODIS,36,6.3,298.8,27.6,D,0
-12.6191,142.1998,314.8,1,1,2019-08-01,0057,Terra,MODIS,68,6.3,297.6,9.3,D,0
-14.3655,143.5682,305.4,1.2,1.1,2019-08-01,0057,Terra,MODIS,24,6.3,283.9,5.9,D,0
-14.3195,143.5198,322.9,1.2,1.1,2019-08-01,0057,Terra,MODIS,79,6.3,290.9,20.4,D,0
-13.1654,141.9715,317.2,1,1,2019-08-01,0057,Terra,MODIS,72,6.3,300,9.9,D,0
-11.5473,132.6796,311.5,3.4,1.7,2019-08-01,0057,Terra,MODIS,40,6.3,298.7,27.3,D,0
-11.5417,132.649,312.2,3.4,1.7,2019-08-01,0057,Terra,MODIS,42,6.3,298,31.4,D,0
-11.5471,132.6538,316,3.4,1.7,2019-08-01,0057,Terra,MODIS,65,6.3,298,52.7,D,0
-12.7626,142.1759,335.2,1,1,2019-08-01,0057,Terra,MODIS,88,6.3,301.7,30.1,D,0
-12.705,133.907,311.7,2.6,1.5,2019-08-01,0057,Terra,MODIS,57,6.3,298.6,20.6,D,0
-13.1092,134.1848,315.9,2.4,1.5,2019-08-01,0057,Terra,MODIS,67,6.3,298.4,29.5,D,0
-14.0539,142.3327,316,1,1,2019-08-01,0057,Terra,MODIS,52,6.3,298.8,7.8,D,0
-12.6996,133.9138,310.9,2.6,1.5,2019-08-01,0057,Terra,MODIS,55,6.3,298.7,17.6,D,0
-12.7499,134.5251,314.8,2.3,1.5,2019-08-01,0057,Terra,MODIS,66,6.3,299.9,22.8,D,0
-12.7141,134.3991,313.9,2.4,1.5,2019-08-01,0057,Terra,MODIS,64,6.3,299.2,23,D,0
-12.7061,134.4077,314,2.4,1.5,2019-08-01,0057,Terra,MODIS,64,6.3,300,21.2,D,0
-13.8656,141.9771,315.7,1,1,2019-08-01,0057,Terra,MODIS,69,6.3,300.8,6.7,D,0
-12.6491,134.3953,324.9,2.4,1.5,2019-08-01,0057,Terra,MODIS,79,6.3,300.6,57.7,D,0
-13.1015,134.1926,311.3,2.4,1.5,2019-08-01,0057,Terra,MODIS,55,6.3,298.1,17.2,D,0
-12.6703,142.1102,322.4,1,1,2019-08-01,0057,Terra,MODIS,78,6.3,301,13.7,D,0
-13.3042,142.04,318.6,1,1,2019-08-01,0057,Terra,MODIS,76,6.3,299,11.6,D,0
-13.2236,141.9914,325.3,1,1,2019-08-01,0057,Terra,MODIS,69,6.3,300.4,16.2,D,0
-13.2821,142.4617,306.5,1,1,2019-08-01,0057,Terra,MODIS,19,6.3,280.3,5.4,D,0
-13.2159,142.002,325.5,1,1,2019-08-01,0057,Terra,MODIS,52,6.3,297.3,17.6,D,0
-13.2537,142.5242,310,1,1,2019-08-01,0057,Terra,MODIS,37,6.3,291.3,4.3,D,0
-12.4513,141.8071,320.2,1,1,2019-08-01,0057,Terra,MODIS,68,6.3,296,13,D,0
-12.4527,141.8162,328.6,1,1,2019-08-01,0057,Terra,MODIS,83,6.3,298.2,21.8,D,0
-12.544,142.4455,317.6,1,1,2019-08-01,0057,Terra,MODIS,57,6.3,301.1,8.8,D,0
-12.463,141.8241,317,1,1,2019-08-01,0057,Terra,MODIS,51,6.3,298.3,8.2,D,0
-12.5567,141.8979,336.5,1,1,2019-08-01,0057,Terra,MODIS,88,6.3,304.4,31.5,D,0
-12.5581,141.9071,318.2,1,1,2019-08-01,0057,Terra,MODIS,56,6.3,304.8,8.9,D,0
-13.2313,131.9106,316.7,3.6,1.8,2019-08-01,0057,Terra,MODIS,68,6.3,300,49.9,D,0
```

Figure 5.2: **Raw data**

5.1.2 Output Design

```
random_new.fit(Xtrain, ytrain)

y_pred1 = random_new.predict(Xtest)

#Checking the accuracy
random_model_accuracy1 = round(random_new.score(Xtrain, ytrain)*100,2)
print(round(random_model_accuracy1, 2), '%')

95.55 %

random_model_accuracy2 = round(random_new.score(Xtest, ytest)*100,2)
print(round(random_model_accuracy2, 2), '%')

68.02 %

saved_model = pickle.dump(random_new, open('ForestModel.pickle','wb'))
print("model saved")

model saved
```

Figure 5.3: **output**

5.2 Testing

Data created or selected to satisfy the execution preconditions and inputs to execute one or more test cases.” There is a lot of attention for testing methods like security testing, performance testing or regression testing.

5.3 Types of Testing

5.3.1 Unit testing

Unit testing is a software testing in which individual components or units of software are tested.The main aim of unit testing is to evaluate each individual unit’s performance in software.In this project approach the data components like Latitudes,longitudes, climate, temperature, light, segmentation techniques are tested and finalized that there are no such random unappropriate values are presented.

5.3.2 Integration testing

White box testing is a software test approach in which the internal schema of the item is known to the tester.The tester pick out the input attributes that brought up more in case of predicting the data that is incomplete, incorrect, irrelevant, duplicated, or improperly formatted which creates abnormalities to practicing paths through the code and concludes the predicted rate accurately.

5.3.3 System testing

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

5.3.4 Test Result

```
random_new.fit(Xtrain, ytrain)

y_pred1 = random_new.predict(Xtest)

#Checking the accuracy
random_model_accuracy1 = round(random_new.score(Xtrain, ytrain)*100,2)
print(round(random_model_accuracy1, 2), '%')
```

```
95.55 %
```

```
random_model_accuracy2 = round(random_new.score(Xtest, ytest)*100,2)
print(round(random_model_accuracy2, 2), '%')
```

```
68.02 %
```

```
saved_model = pickle.dump(random_new, open('ForestModel.pickle','wb'))
print("model saved")
```

```
model saved
```

Figure 5.4: **Test Image**

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

Proposed model is completely automatic and statistical analytical tool for predicting forest fire. Accurate results regarding the fire effect and required predictions are well defined. The prediction of the fire helps in producing exact outcomes. In the Prediction of forest fire, the collected forest exploration data undergoes Pre-Processing for better analysis. In this step, all the collected will be separated uniformly so that the perfect data will be created. And then after the Pre-Processing, Machine Learning Algorithm will be applied for the data. Then the data will be trained and the model will be built based on the collected data. Then the data will be tested and the deployment takes place. In the deployment step, the data will be moved to a place where some action can be performed in it. And after that we will find the accuracy which is the key concept for prediction and finally we will get the output.

6.2 Comparison of Existing and Proposed System

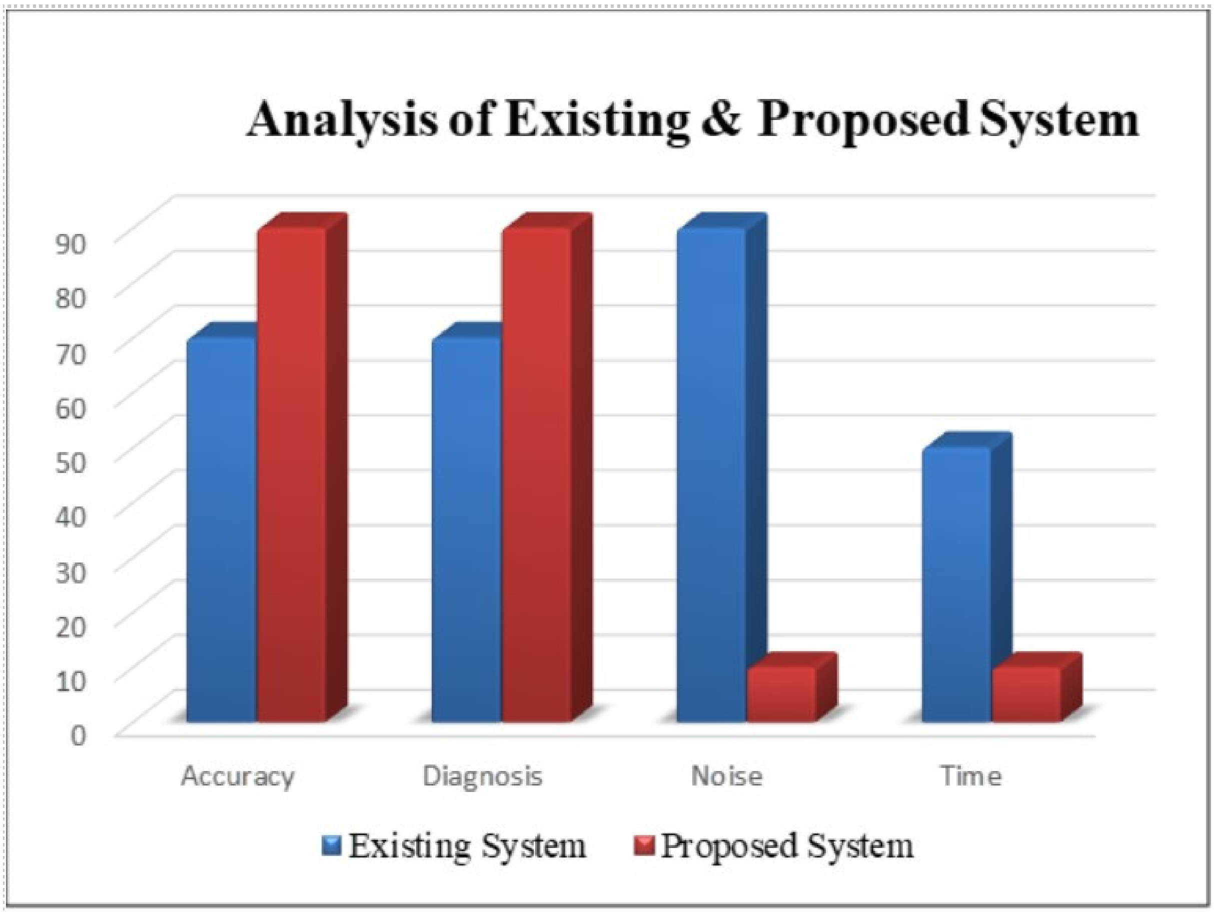

Figure 6.1: **Analytics Graph**

Bar Graph Representation of existing system of Forest Fire Prediction using Convolution Neural Networks with Proposed System using Machine Learning for Forest Fire Prediction.

- Advantages
  - Proposed an automatic Machine Learning- based method for prediction of forest fire by using latitudes, longitudes, climate, temperature, brightness etc.,
  - Data Accuracy.
  - Efficient Data Management.
  - Proposed the system that gives accurate results regarding the fire effect and required predictions .

6.3 Sample Code

```
import datetime as dt
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
from sklearn.ensemble import RandomForestRegressor
forest = pd.read_csv('fire_archive.csv')
forest.head()
forest.shape
forest.columns
forest.isnull().sum()
forest.describe()
plt.figure(figsize=(10, 10))
sns.heatmap(forest.corr(),annot=True,cmap='viridis',linewidths=.5)
forest = forest.drop(['track'], axis = 1)
print("The scan column")
print(forest['scan'].value_counts())
print()
print("The aqc_time column")
print(forest['acq_time'].value_counts())
print()
print("The satellite column")
print(forest['satellite'].value_counts())
print()
print("The instrument column")
print(forest['instrument'].value_counts())
print()
print("The version column")
print(forest['version'].value_counts())
print()
print("The daynight column")
print(forest['daynight'].value_counts())
print()
forest = forest.drop(['instrument', 'version'], axis = 1)

forest.head()
daynight_map = {"D": 1, "N": 0}
satellite_map = {"Terra": 1, "Aqua": 0}

forest['daynight'] = forest['daynight'].map(daynight_map)
forest['satellite'] = forest['satellite'].map(satellite_map)

forest.head()
forest['type'].value_counts()
types = pd.get_dummies(forest['type'])
forest = pd.concat([forest, types], axis=1)
```

```
forest = forest.drop(['type'], axis = 1)
forest.head()
forest = forest.rename(columns={0: 'type_0', 2: 'type_2', 3: 'type_3'})
bins = [0, 1, 2, 3, 4, 5]
labels = [1,2,3,4,5]
forest['scan_binned'] = pd.cut(forest['scan'], bins=bins, labels=labels)

forest.head()
# Converting the datatype to datetype from string or numpy.

forest['acq_date'] = pd.to_datetime(forest['acq_date'])
forest['year'] = forest['acq_date'].dt.year

forest.head()
forest['month'] = forest['acq_date'].dt.month
forest['day'] = forest['acq_date'].dt.day
forest.shape
y = forest['confidence']
fin = forest.drop(['confidence', 'acq_date', 'acq_time', 'bright_t31', 'type_0'], axis = 1)
plt.figure(figsize=(10, 10))
sns.heatmap(fin.corr(),annot=True,cmap='viridis',linewidths=.5)
fin.head()
Xtrain, Xtest, ytrain, ytest = train_test_split(fin.iloc[:, :500], y, test_size=0.2)
random_model = RandomForestRegressor(n_estimators=300, random_state = 42, n_jobs = -1)
random_model.fit(Xtrain, ytrain)

y_pred = random_model.predict(Xtest)

#Checking the accuracy
random_model_accuracy = round(random_model.score(Xtrain, ytrain)*100,2)
print(round(random_model_accuracy, 2), '%')
random_model_accuracy1 = round(random_model.score(Xtest, ytest)*100,2)
print(round(random_model_accuracy1, 2), '%')
import pickle

saved_model = pickle.dump(random_model, open('ForestModelOld.pickle','wb'))
random_model.get_params()
from sklearn.model_selection import RandomizedSearchCV
n_estimators = [int(x) for x in np.linspace(start = 300, stop = 500, num = 20)]
# Number of features to consider at every split
max_features = ['auto', 'sqrt']
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(15, 35, num = 7)]
max_depth.append(None)
# Minimum number of samples required to split a node
min_samples_split = [2, 3, 5]
# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2, 4]
# Create the random grid
```

```
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf,
               }
print(random_grid)
rf_random = RandomizedSearchCV(estimator = random_model, param_distributions = random_grid, n_iter =
    50, cv = 3, verbose=2, random_state=42)
# Fit the random search model
rf_random.fit(Xtrain, ytrain)
rf_random.best_params_
random_new = RandomForestRegressor(n_estimators = 394, min_samples_split = 2, min_samples_leaf = 1,
    max_features = 'sqrt',
                                   max_depth = 25, bootstrap = True)
                                   random_new.fit(Xtrain, ytrain)

y_pred1 = random_new.predict(Xtest)

#Checking the accuracy
random_model_accuracy1 = round(random_new.score(Xtrain, ytrain)*100,2)
print(round(random_model_accuracy1, 2), '%')
random_model_accuracy2 = round(random_new.score(Xtest, ytest)*100,2)
print(round(random_model_accuracy2, 2), '%')
saved_model = pickle.dump(random_new, open('ForestModel.pickle','wb'))
print("model saved")
```

Output

```
random_new.fit(Xtrain, ytrain)

y_pred1 = random_new.predict(Xtest)

#Checking the accuracy
random_model_accuracy1 = round(random_new.score(Xtrain, ytrain)*100,2)
print(round(random_model_accuracy1, 2), '%')
```

```
95.55 %
```

```
random_model_accuracy2 = round(random_new.score(Xtest, ytest)*100,2)
print(round(random_model_accuracy2, 2), '%')
```

```
68.02 %
```

```
saved_model = pickle.dump(random_new, open('ForestModel.pickle','wb'))
print("model saved")
```

```
model saved
```

Figure 6.2: **Output**

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

The prediction of fire using Machine Learning is challenging as well as innovative. If this system with less error rate can be implemented at a large scale like in big factories, houses, forests, it is possible to prevent damage and loss due to random fire accidents by making use of the Surveillance Systems. The algorithm shows great promise in adapting to various environment.The experimental results show that the proposed method achieves high and stable results even with unstable metric. Forest fires are a typical occurrence within the flora and fauna. Every year, legion hectares of forest are destroyed round the world. This resulted in significant environmental harm in addition because the loss of irreplaceable human lives. Forest fires are a serious environmental hazard that threatens forest preservation, causing economic and ecological harm likewise as human suffering. Quick fire detection and reaction are efficient methods for decreasing fire damage. Various studies are conducted in try to improve early fire prediction.We completed our research and constructed a system that forecasts the proportions of fires that may occur are supported by the weather data provided by the user, like temperature, oxygen, and humidity. The strategy accurately predicts the proportion of fires that may occur.

7.2 Future Enhancements

Forest Fire Detection is a never ending project, that can keep on adding new features to it and make it more efficient and better. Structure function analyses are revealing new details of Fire prediction. The new features that are going to be added in the future helps in not only in the prediction of the fire but also it's prevention. In future,We can have an additional alerting system that can be used for alerting specific department by just uploading the weather data of a particular station. This way we can help in managing forest fires before it destroys the whole forest. This makes prevention easier by predicting forest fires easily. In the future, we aim to improve our model by improving accuracy and speed. We would also like to make live predictions or on-site predictions or make it sensor-based and using satellite images.

Chapter 8

INDUSTRY DETAILS

8.1 Industry name

COGNIZANT TECHNOLOGY SOLUTIONS INDIA

8.1.1 Duration of Internship (From Date - To Date)

5th Feb 2022 - 30th July 2022

8.1.2 Duration of Internship in months

3-6 Months

8.1.3 Industry Address

COGNIZANT TECHNOLOGY SOLUTIONS INDIA PRIVATE LIMITED.registered office at Techno Complex, 5/535, Old Mahabalipuram Road, Okkiyam Thoraipakkam, Chennai 600 097

8.2 Internship offer letter

Cognizant

14-Jan-2022

Seela Jyothirmai
B.Tech Computer Science & Engineering
Vel Tech Rangarajan Dr. Sagunthala R and D Institute of Science and Technology, Chennai

Dear Seela Jyothirmai,

Further to our Letter of Intent / Offer for the position of Programmer Analyst Trainee / Programmer Analyst aligned to the hiring category and in response to your subsequent confirmation for Internship Program with us, we are pleased to offer you an Internship with us for **a period of 3 to 6 months.** Your Internship onboarding will be scheduled anytime between now, through end of March 2022 based on your availability factoring your college exam schedule and our business requirements.

During this period, you will be provided with a stipend of INR 12,000 per month equated to the planned duration of the Internship curriculum and will be paid only subject to successful completion of milestones as defined in the curriculum prior to the monthly stipend processing window for a given month based on your performance and attendance.

Actual commencement of Internship dates and duration would be shortly communicated to you and the internship would be based on the business demand aligned to your skill tracks.

Though Cognizant Internship being a pre- requisite skill and capability development program, it does not guarantee employment. However, the successful completion of internship will form a critical part of your employment with Cognizant if an opportunity arises in future.

You will undergo a learning curriculum as per the learning track assigned to you. The learning path will include in-depth sessions, hands on exercise and project work. There will also be series of webinars, quizzes, SME interactions, mentor connects, code challenges, assessments etc. to accelerate your learning. The outcome during Internship would be monitored through formal evaluations.

Prior to joining on the rolls of Cognizant, you must have successfully completed the prescribed Internship program. In the event of unsatisfactory Internship, Cognizant reserves rights at its sole discretion to revoke its employment offer.

Please also note that:

- The Internship timings would be for 9 hours per day from Monday through Friday aligned to the working timings followed in Cognizant
- Interns are covered under Cognizant's calendar holidays of the respective location of internship and you would need to adhere with minimum attendance requirements. Prior approvals are must towards any unavoidable leave or break requests during the program.
- There would be zero tolerance to plagiarisms and misconduct during the internship. Any such incident reported will lead to immediate cancellation of internship without any notice.
- You would be required to ensure timely completion and submission of assignments, project work and preparation required prior to the sessions.
- You may be required, to travel to other locations within India if there is a business need as per your internship program
- Cognizant reserves rights regarding IT infra as applicable and access to information and material of Cognizant during the internship period and may modify or amend the Cognizant GenC program terms and conditions from time to time
- Stipend payment will be done for the prescribed Internship Curriculum period only and no additional payment will be done for any delay in completion.

Cognizant

14-Jan-2022

Thanushri R
B.Tech Computer Science
Vel Tech Rangarajan Dr. Sagunthala R and D Institute of Science and Technology, Chennai

Dear Thanushri R,

Further to our Letter of Intent / Offer for the position of Programmer Analyst Trainee / Programmer Analyst aligned to the hiring category and in response to your subsequent confirmation for Internship Program with us, we are pleased to offer you an Internship with us for **a period of 3 to 6 months.** Your Internship onboarding will be scheduled anytime between now, through end of March 2022 based on your availability factoring your college exam schedule and our business requirements.

During this period, you will be provided with a stipend of INR 12,000 per month equated to the planned duration of the Internship curriculum and will be paid only subject to successful completion of milestones as defined in the curriculum prior to the monthly stipend processing window for a given month based on your performance and attendance.

Actual commencement of Internship dates and duration would be shortly communicated to you and the internship would be based on the business demand aligned to your skill tracks.

Though Cognizant Internship being a pre- requisite skill and capability development program, it does not guarantee employment. However, the successful completion of internship will form a critical part of your employment with Cognizant if an opportunity arises in future.

You will undergo a learning curriculum as per the learning track assigned to you. The learning path will include in-depth sessions, hands on exercise and project work. There will also be series of webinars, quizzes, SME interactions, mentor connects, code challenges, assessments etc. to accelerate your learning. The outcome during Internship would be monitored through formal evaluations.

Prior to joining on the rolls of Cognizant, you must have successfully completed the prescribed Internship program. In the event of unsatisfactory Internship, Cognizant reserves rights at its sole discretion to revoke its employment offer.

Please also note that:

• The Internship timings would be for 9 hours per day from Monday through Friday aligned to the working timings followed in Cognizant
• Interns are covered under Cognizant's calendar holidays of the respective location of internship and you would need to adhere with minimum attendance requirements. Prior approvals are must towards any unavoidable leave or break requests during the program.
• There would be zero tolerance to plagiarisms and misconduct during the internship. Any such incident reported will lead to immediate cancellation of internship without any notice.
• You would be required to ensure timely completion and submission of assignments, project work and preparation required prior to the sessions.
• You may be required, to travel to other locations within India if there is a business need as per your internship program
• Cognizant reserves rights regarding IT infra as applicable and access to information and material of Cognizant during the internship period and may modify or amend the Cognizant GenC program terms and conditions from time to time
• Stipend payment will be done for the prescribed Internship Curriculum period only and no additional payment will be done for any delay in completion.

Cognizant

14-Jan-2022

Dontamsetti Chidanand
B.Tech Computer Science & Engineering
Vel Tech Rangarajan Dr. Sagunthala R and D Institute of Science and Technology, Chennai

Dear Dontamsetti Chidanand,

Further to our Letter of Intent / Offer for the position of Programmer Analyst Trainee / Programmer Analyst aligned to the hiring category and in response to your subsequent confirmation for Internship Program with us, we are pleased to offer you an Internship with us for **a period of 3 to 6 months.** Your Internship onboarding will be scheduled anytime between now, through end of March 2022 based on your availability factoring your college exam schedule and our business requirements.

During this period, you will be provided with a stipend of INR 12,000 per month equated to the planned duration of the Internship curriculum and will be paid only subject to successful completion of milestones as defined in the curriculum prior to the monthly stipend processing window for a given month based on your performance and attendance.

Actual commencement of Internship dates and duration would be shortly communicated to you and the internship would be based on the business demand aligned to your skill tracks.

Though Cognizant Internship being a pre- requisite skill and capability development program, it does not guarantee employment. However, the successful completion of internship will form a critical part of your employment with Cognizant if an opportunity arises in future.

You will undergo a learning curriculum as per the learning track assigned to you. The learning path will include in-depth sessions, hands on exercise and project work. There will also be series of webinars, quizzes, SME interactions, mentor connects, code challenges, assessments etc. to accelerate your learning. The outcome during Internship would be monitored through formal evaluations.

Prior to joining on the rolls of Cognizant, you must have successfully completed the prescribed Internship program. In the event of unsatisfactory Internship, Cognizant reserves rights at its sole discretion to revoke its employment offer.

Please also note that:

• The Internship timings would be for 9 hours per day from Monday through Friday aligned to the working timings followed in Cognizant
• Interns are covered under Cognizant's calendar holidays of the respective location of internship and you would need to adhere with minimum attendance requirements. Prior approvals are must towards any unavoidable leave or break requests during the program.
• There would be zero tolerance to plagiarisms and misconduct during the internship. Any such incident reported will lead to immediate cancellation of internship without any notice.
• You would be required to ensure timely completion and submission of assignments, project work and preparation required prior to the sessions.
• You may be required, to travel to other locations within India if there is a business need as per your internship program
• Cognizant reserves rights regarding IT infra as applicable and access to information and material of Cognizant during the internship period and may modify or amend the Cognizant GenC program terms and conditions from time to time
• Stipend payment will be done for the prescribed Internship Curriculum period only and no additional payment will be done for any delay in completion.
• Attendance and successful completion of Milestone(s) are the eligible factors for processing stipend

Chapter 9

PLAGIARISM REPORT

Curiginal

Document Information

Analyzed document	Batch-33 Major Project Report.pdf (D136412750)
Submitted	2022-05-13T09:30:00.0000000
Submitted by	Malarvizhi
Submitter email	drnmalarvizhi@veltech.edu.in
Similarity	8%
Analysis address	drnmalarvizhi.veltec@analysis.urkund.com

Figure 9.1: **Plagiarism Report**

Chapter 10

SOURCE CODE & POSTER PRESENTATION

10.1 Source Code

```
import datetime as dt
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
from sklearn.ensemble import RandomForestRegressor
forest = pd.read_csv('fire_archive.csv')
forest.head()
forest.shape
forest.columns
forest.isnull().sum()
forest.describe()
plt.figure(figsize=(10, 10))
sns.heatmap(forest.corr(),annot=True,cmap='viridis',linewidths=.5)
forest = forest.drop(['track'], axis = 1)
print("The scan column")
print(forest['scan'].value_counts())
print()
print("The aqc_time column")
print(forest['acq_time'].value_counts())
print()
print("The satellite column")
print(forest['satellite'].value_counts())
print()
print("The instrument column")
print(forest['instrument'].value_counts())
print()
print("The version column")
print(forest['version'].value_counts())
print()
print("The daynight column")
print(forest['daynight'].value_counts())
```

```
print()
forest = forest.drop(['instrument', 'version'], axis = 1)

forest.head()
daynight_map = {"D": 1, "N": 0}
satellite_map = {"Terra": 1, "Aqua": 0}

forest['daynight'] = forest['daynight'].map(daynight_map)
forest['satellite'] = forest['satellite'].map(satellite_map)

forest.head()
forest['type'].value_counts()
types = pd.get_dummies(forest['type'])
forest = pd.concat([forest, types], axis=1)

forest = forest.drop(['type'], axis = 1)
forest.head()
forest = forest.rename(columns={0: 'type_0', 2: 'type_2', 3: 'type_3'})
bins = [0, 1, 2, 3, 4, 5]
labels = [1,2,3,4,5]
forest['scan_binned'] = pd.cut(forest['scan'], bins=bins, labels=labels)

forest.head()
# Converting the datatype to datetype from string or numpy.

forest['acq_date'] = pd.to_datetime(forest['acq_date'])
forest['year'] = forest['acq_date'].dt.year

forest.head()
forest['month'] = forest['acq_date'].dt.month
forest['day'] = forest['acq_date'].dt.day
forest.shape
y = forest['confidence']
fin = forest.drop(['confidence', 'acq_date', 'acq_time', 'bright_t31', 'type_0'], axis = 1)
plt.figure(figsize=(10, 10))
sns.heatmap(fin.corr(),annot=True,cmap='viridis',linewidths=.5)
fin.head()
Xtrain, Xtest, ytrain, ytest = train_test_split(fin.iloc[:, :500], y, test_size=0.2)
random_model = RandomForestRegressor(n_estimators=300, random_state = 42, n_jobs = -1)
random_model.fit(Xtrain, ytrain)

y_pred = random_model.predict(Xtest)

#Checking the accuracy
random_model_accuracy = round(random_model.score(Xtrain, ytrain)*100,2)
print(round(random_model_accuracy, 2), '%')
random_model_accuracy1 = round(random_model.score(Xtest, ytest)*100,2)
print(round(random_model_accuracy1, 2), '%')
import pickle

```

```
saved_model = pickle.dump(random_model, open('ForestModelOld.pickle','wb'))
random_model.get_params()
from sklearn.model_selection import RandomizedSearchCV
n_estimators = [int(x) for x in np.linspace(start = 300, stop = 500, num = 20)]
# Number of features to consider at every split
max_features = ['auto', 'sqrt']
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(15, 35, num = 7)]
max_depth.append(None)
# Minimum number of samples required to split a node
min_samples_split = [2, 3, 5]
# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2, 4]
# Create the random grid
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf,
               }
print(random_grid)
rf_random = RandomizedSearchCV(estimator = random_model, param_distributions = random_grid, n_iter =
    50, cv = 3, verbose=2, random_state=42)
# Fit the random search model
rf_random.fit(Xtrain, ytrain)
rf_random.best_params_
random_new = RandomForestRegressor(n_estimators = 394, min_samples_split = 2, min_samples_leaf = 1,
    max_features = 'sqrt',
                                      max_depth = 25, bootstrap = True)
                                      random_new.fit(Xtrain, ytrain)

y_pred1 = random_new.predict(Xtest)

#Checking the accuracy
random_model_accuracy1 = round(random_new.score(Xtrain, ytrain)*100,2)
print(round(random_model_accuracy1, 2), '%')
random_model_accuracy2 = round(random_new.score(Xtest, ytest)*100,2)
print(round(random_model_accuracy2, 2), '%')
saved_model = pickle.dump(random_new, open('ForestModel.pickle','wb'))
print("model saved")
```

10.2 Poster Presentation

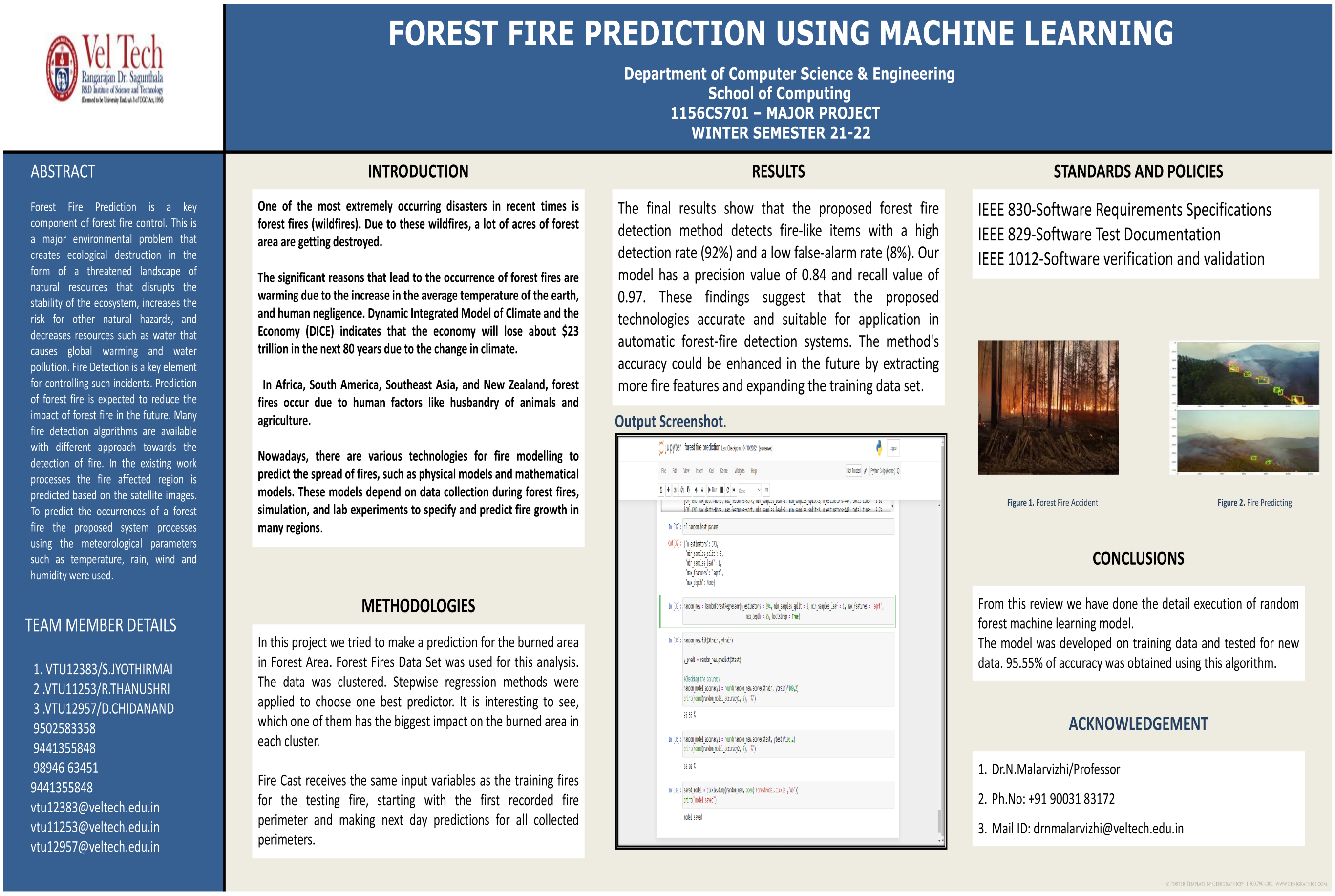

Figure 10.1: **Poster Presentation**

References

[1] Salis M, Arca B, Alcasena F, Arianoutsou M, Bacciu V, Duce P, Duguay B, Koutsias N, Mallinis G, Mitsopoulos I, Moreno JM(2019). "Predicting wildfire spread and behaviour in Mediterranean landscapes", International Journal of Wildland Fire, volume 25, pages 1015 -1032.

[2] Castelli M, Vanneschi L, Popovič A.(2020). "Predicting burned areas of forest fires: an artificial intelligence approach", Fire Ecology, volume 11, pages 106-118.

[3] Dieu Tien Buia, Nhat-Duc Hoangc, Pijush Samuid(2019). "Spatial pattern analysis and prediction of forest fire using new machine learning approach of Multivariate Adaptive Regression Splines and Differential Flower Pollination optimization: A case study at Lao Cai province (VietNam)",Journal of Environmental Management, Volume 237, Pages 476-487 .

[4] Pérez-Sánchez J, Jimeno-Sáez P, Senent-Aparicio J, Díaz-Palmero JM, Cabezas-Cerezo JD(2019). "Evolution of burned area in forest fires under climate change conditions in southern Spain using ANN", Applied Sciences, volume 9, pages 4155.

[5] Haoyuan Hong, Paraskevas Tsangaratos,Ioanna Ilia,Junzhi Liu, A-Xing Zhu , Chong Xu(2018)."Applying genetic algorithms to set the optimal combination of forest fire related variables and model forest fire susceptibility based on data mining models. The case of Dayu County, China", Science of The Total Environment ,Volume 630,Pages 1044- 1056.

[6] Singh BK, Verma K, Thoke AS,(2018). "Investigations on the impact of feature normalization techniques on classifier's performance in breast tumour classification", International Journal of Computer Applications, volume 116, pages 11-15.

[7] Pereira JM, Basto M, da Silva AF(2020)."The logistic lasso and ridge regression in predicting corporate failure", Procedia Economics and Finance, volume 39, pages 634-641.

[8] Garrard P, Rentoumi V, Gesierich B, Miller B, Gorno-Tempini ML.(2021). “Machine learning approaches to diagnosis and laterality effects in semantic dementia discourse”, Cortex, volume 55, pages 122-129.

[9] Al Janabi S, Al Shourbaji I, Salman MA (2018).“Assessing the suitability of soft computing approaches for forest fires prediction”, Applied Computing and Informatics, volume 14, pages 214-224.

[10] Boubeta M, Lombardía MJ, González-Manteiga W, Marey-Pérez MF (2018). “Burned area prediction with semiparametric models”, International Journal of Wildland Fire, volume 25, pages 669-678.