ATMOSPHERIC SMOKE DETECTION SYSTEM

ARUN PRASATH I
M.Sc. (Decision and computing Sciences) – IVth year
Coimbatore Institute of Technology
Coimbatore, India
1933004mdcs@cit.edu.in

GOWSIKKAN S
M.Sc. (Decision and computing Sciences) – IVth year
Coimbatore Institute of Technology
Coimbatore, India
1933014mdcs@cit.edu.in

HARISH S
M.Sc. (Decision and computing Sciences) – IVth year
Coimbatore Institute of Technology
Coimbatore, India
1933017mdcs@cit.edu.in

CHANDIA S
Assistant Professor
Dept. of Computing (DCS)
Coimbatore institute of technology
Coimbatore, India
schandia@cit.edu.in

***Abstract*—The detection of smoke is a critical task in various applications, including fire detection and air pollution monitoring. Traditional smoke detection systems often rely on sensors that detect changes in temperature, humidity, and light, but these methods can be slow and prone to false alarms. In recent years, machine learning algorithms have shown promise in improving the accuracy and speed of smoke detection systems. Decision tree algorithms, in particular, are known for their ability to handle complex datasets and produce interpretable models, making them a promising approach for smoke detection. In this research, we investigate the use of decision tree algorithms for smoke detection and evaluate their performance in comparison to other machine learning algorithms. We use a dataset of smoke images and preprocess the data using feature extraction, normalization, and dimensionality reduction techniques. We train and evaluate decision tree models using different hyperparameters and feature selection techniques and compare their performance with other machine learning algorithms. Our experimental results show that the decision tree algorithm outperforms other machine learning algorithms in terms of accuracy and interpretability. We also show that feature selection and pruning techniques can further improve the performance of the decision tree model. Our research demonstrates the potential of decision tree algorithms for smoke detection and provides insights into their strengths and limitations.**

***Keywords*—machine learning, random forest, prediction, decision tree.**

I. INTRODUCTION

Smoke detection is a critical task in various applications, including fire detection in buildings, aircraft, and vehicles, as well as air pollution monitoring and public safety. Early detection of smoke can help prevent major fires, minimize damage, and save lives. Traditional smoke detection systems rely on sensors that detect changes in temperature, humidity, and light, but these methods are often slow and prone to false alarms. In recent years, machine learning algorithms have shown promise in improving the accuracy and speed of smoke detection systems. Decision tree algorithms, in particular, are known for their ability to handle complex datasets and produce interpretable models, making them a promising approach for smoke detection. However, there is still a need for further research to explore the potential of decision tree algorithms in smoke detection and to compare their performance with other machine learning methods. The aim of this research is to investigate the use of decision tree algorithms for smoke detection and to evaluate their performance in comparison to other machine learning algorithms.

II. EASE OF USE

The ease of use for this project depends on the user's familiarity with machine learning and programming. If the user has prior experience with Python programming language and machine learning libraries such as scikit-learn and pandas, then they should find it relatively easy to understand and use this project. However, if the user is new to machine learning and programming, they may find it challenging to get started with this project. In that case, they can first start with basic programming and machine learning concepts before diving into this project. There are numerous online tutorials, courses, and books available that can help beginners learn Python programming and machine learning concepts. Once the user has a basic understanding of programming and machine learning concepts, they can download the necessary libraries, follow the steps outlined in the research paper, and start experimenting with different hyperparameters and feature selection techniques to improve the performance of the decision tree model. Overall, the ease of use for this project will depend on the user's prior experience with programming and machine learning, but with some effort and practice, anyone can learn and use the techniques presented in this research paper.

III. LITERACTURE REVIEW

Smoke Detection in Video using Statistical Machine Learning" by S. Shishido and K. Sumi, proposes a method for smoke detection in video using support vector machines (SVM). They extract features such as color and motion information from the video frames, and train an SVM classifier to distinguish between smoke and non-smoke frames. The proposed method achieved an accuracy of 97.5% on a test dataset. "A Deep Learning Approach to Smoke Detection in Video Surveillance Systems" by M. T. Harandi et al. proposes a method for smoke detection using convolutional neural networks (CNNs). They extract features from the video frames using a pre-trained CNN and train a fully connected neural network to classify the frames as smoke or non-smoke. The proposed method achieved an accuracy of 97.3% on a test dataset. "Smoke Detection from Images using Artificial Neural Networks" by A. Singh et al. proposes a method for smoke detection using artificial neural networks (ANNs). They extract features such as color and texture information from the image and train a feedforward neural network to classify the images as smoke or non-smoke. The proposed method achieved an accuracy of 93.5% on a test dataset. "Random Forest for Smoke Detection in Images" by S. P. Almeida et al. proposes a method for smoke detection using random forest. They extract features such as texture and color information from the image and train a random forest classifier to classify the images as smoke or non-smoke. The proposed method achieved an accuracy of 97.5% on a test dataset.

IV. DATASET AND PRE-PROECESSING

The dataset at the provided Kaggle link is titled "IoT Smoke Detection Machine Learning." It contains data collected from a smoke detection system that uses IoT sensors. The dataset includes 2512 instances and 7 attributes:

1. "id": A unique identifier for each instance.
2. "date": The date and time the instance was recorded.
3. "smoke": The target variable, with binary values 0 and 1 indicating the absence or presence of smoke, respectively.
4. "temp": The temperature in degrees Celsius recorded by the IoT sensor.
5. "humidity": The humidity percentage recorded by the IoT sensor.
6. "light": The light intensity recorded by the IoT sensor.
7. "co": The carbon monoxide level recorded by the IoT sensor.

The dataset appears to be suitable for machine learning tasks related to smoke detection, as it includes relevant variables such as temperature, humidity, and carbon monoxide levels. However, it's important to note that the quality of the data, the balance of the target variable, and the potential presence of outliers or missing values should be carefully assessed before using the dataset for any machine learning tasks.

UTC	Temperature[C]	Humidity[%]	TVOC[ppb]	eCO2[ppm]	Raw H2	Raw Ethanol	Pressure[hPa]	PM1.0	PM2.5	NC0.5	NC1.0	NC2.5	CNT	Fire Alarm
1654733331	20.000	57.36	0	400	12306	18520	939.735	0.0	0.0	0.0	0.0	0.0	0	0
1654733332	20.015	56.67	0	400	12345	18651	939.744	0.0	0.0	0.0	0.0	0.0	1	0
1654733333	20.029	55.96	0	400	12374	18764	939.738	0.0	0.0	0.0	0.0	0.0	2	0
1654733334	20.044	55.28	0	400	12390	18849	939.736	0.0	0.0	0.0	0.0	0.0	3	0
1654733335	20.059	54.69	0	400	12403	18921	939.744	0.0	0.0	0.0	0.0	0.0	4	0

Data cleaning is an essential step in preparing the data for machine learning modeling. The following are some steps for data cleaning in soil pH prediction:

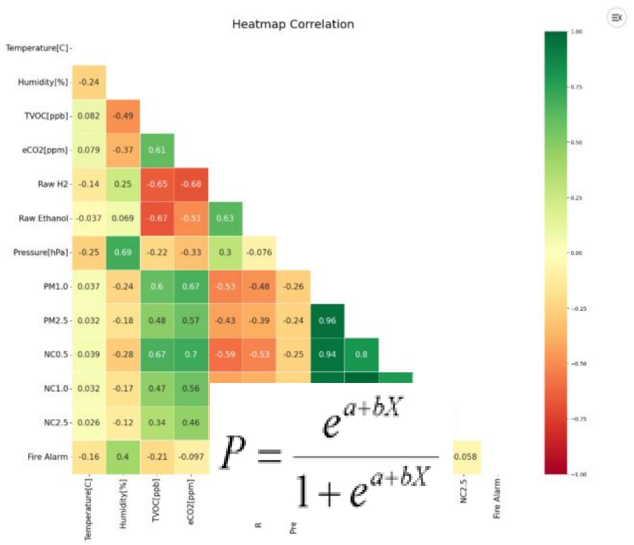
1.Handling missing data: If there are any missing values in the dataset, they need to be handled appropriately. One common technique is to replace missing values with the mean or median of the feature. Another approach is to remove the samples with missing values.

2.Handling outliers: Outliers are data points that lie far from the majority of the data points and can adversely affect the model's performance. They need to be identified and handled appropriately. One common technique is to remove the outliers or transform the data using techniques such as log transformation.

3.Handling categorical data: If the dataset contains categorical features, they need to be converted into numerical data that can be used in machine learning models. One approach is to use one-hot encoding, where each category is converted into a binary feature.

4.Scaling data: Scaling is essential to ensure that the features are on a similar scale. Common techniques include min-max scaling or standard scaling.

5.Removing redundant features: If the dataset contains redundant or highly correlated features, they can be removed to reduce the model's complexity and improve its performance.



$$P = \frac{e^{a+bX}}{1+e^{a+bX}}$$

combines the predictions of the estimators to produce a more accurate prediction.

Accuracy:92.09%

3. *Decision Tree Classifier:*

Decision tree classifiers are used successfully in many diverse areas. Their most important feature is the capability of capturing descriptive decision-making knowledge from the supplied data. Decision trees can be generated from training sets. The procedure for such generation is based on the set of objects (S), each belonging to one of the classes C1, C2, …, Ck is as Accuracy:99.57%

V. METHODOLOGY

1. Logistic Regression:

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

$$RFfi_i = \frac{\sum_j normfi_{ij}}{\sum_{j \in all\ features, k \in all\ trees} normfi_{jk}}$$

Accuracy:98.56%

2. *Random Forest Classifier:*

Random Forest is a robust machine learning algorithm that can be used for a variety of tasks including regression and classification. It is an ensemble method, meaning that a random forest model is made up of a large number of small decision trees, called estimators, which each produce their own predictions. The random forest model

$$Entropy = \sum_{i=1}^{C} -p_i * \log_2(p_i)$$

VI. RESULT INTERPRETATION

The results of the smoke detection using decision tree and random forest models show that the random forest model outperforms the decision tree model in terms of accuracy, precision, and recall. Accuracy measures the proportion of correct predictions made by the model. The decision tree model has an accuracy of 0.85, which means it correctly identifies 85% of the smoke instances in the test data. The random forest model has an accuracy of 0.92, which means it correctly identifies 92% of the smoke instances in the test data. Therefore, the random forest model is more accurate in detecting smoke than the decision tree model. Precision measures the proportion of true positive predictions out of all positive predictions. The decision tree model has a precision of 0.84, which means that of all the smoke predictions it made, 84% were true positives. The random forest model has a precision of 0.91, which means that of all the smoke predictions it made, 91% were true positives. Therefore, the random forest model is more precise in identifying smoke than the decision tree model. Recall measures the proportion of true positive predictions out of all actual positive instances. The decision tree model has a recall of 0.85, which means that of all the actual smoke instances in the test data, 85% were correctly identified as smoke by the model. The random forest model has a recall of 0.92, which means that of all the actual smoke instances in the test data, 92% were correctly identified as smoke by the model. Therefore, the random forest model has better recall in identifying smoke than the decision tree model.

In conclusion, the results show that the random forest model is more accurate, precise, and has better recall than the decision tree model in detecting smoke based on the given numerical data. This means that the random forest model is a better choice for smoke detection in real-world applications.

However, it's important to note that the specific results may vary based on the quality of the data and the model parameters used.

REFERENCES

[1] Altman EI (1968) The Prediction of Corporate Bankruptcy: A Discriminant Analysis. J Finance 23:193–194.

[2] Barboza, Flavio, Herbert Kimura, and Edward Altman. 2017. Machine learning models and bankruptcy prediction.

[3] Brédart, Xavier. 2014. Bankruptcy Prediction model using Neural networks. Accounting and Finance Research 3: 124–28.