**SMOKE DETECTION**

A Course Project report submitted

in partial fulfillment of requirement for the award of degree

**BACHELOR OF TECHNOLOGY**

in

**ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING**

by

**BALA VARSHITHA (2103A52012)**

**CHAITHRA SRI (2103A52027)**

**REETHU VARMA (2103A52037)**

Under the guidance of

**Mr. Ramesh Dadi**

Assistant Professor, Department of CSE.

**

**Department of Computer Science and Artificial Intelligence**



**Department of Computer Science and Artificial Intelligence**

**CERTIFICATE**

This is to certify that project entitled **“SMOKE DETECTION”** is the bonafied work carried out by **Bala varshitha, Chaithra sri, Reethu Varma** as a Course Project for the partial fulfillment to award the degree **BACHELOR OF TECHNOLOGY** in **ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING** during the academic year 2022-2023 under our guidance and Supervision.

**Mr. D.Ramesh Dr. M.Sheshikala**

Asst. Professor, Assoc. Prof. & HOD (CSE), S R University, S R University,

Ananthasagar, Warangal. Ananthasagar, Warangal.

**ACKNOWLEDGEMENT**

We express our thanks to Course co-coordinator **Mr.** **D.Ramesh, Asst. Prof.** for guiding us from the beginning through the end of the Course Project. We express our gratitude toHead of the department CS&AI, **Dr. M.Sheshikala, Associate Professor** for encouragement, support and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved Dean, School of Computer Science and Artificial Intelligence, **Dr C. V. Guru Rao**, for his continuous support and guidance to complete this project in the institute.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

**ABSTRACT**

Fire disaster throughout the globe causes social, environmental, and economical damage, making its early detection and instant reporting essential for saving human lives and properties. Smoke detection plays a key role in early fire detection but majority of the existing methods are limited to either indoor or outdoor surveillance environments, with poor performance for hazy scenarios.

In this paper, we present a K-Nearest Neighbor (KNN)-based smoke detection and segmentation framework for both clear and hazy environments. Unlike existing methods, we employ an efficient KNN architecture, for smoke detection with better accuracy. Our smoke detection results evince a noticeable gain up to 3% in accuracy and a decrease of 0.46% in False Alarm Rate (FAR), while segmentation reports a significant increase of 2% and 1% in global accuracy respectively.

This makes our method a best fit for smoke detection and segmentation in real-world surveillance settings.

**Table of Contents**

**Chapter No. Title Page No.**

1. Introduction
   1. Overview 1
   2. Problem Statement 1-2
   3. Existing system 2
   4. Proposed system 2
   5. Objectives 2
   6. Architecture 3
2. Literature survey 4
3. Data pre-processing
   1. Dataset description 5
   2. Data cleaning 9
   3. Data Visualization 9
4. Methodology
   1. Procedure to solve the given problem 10
   2. Model architecture 11
   3. Software description 11
5. Results and discussion 12
6. Conclusion and future scope 16
7. References 17

**List of figures**

**Topics Page No.**

**1. Smoke detector 3**

**2.Graphs between X&Y 7-8**

**3.KNN confusion matrix 15**

**4.SVM confusion matrix 15**

**List of Tables**

**1.Data set 6**

**2.output 15**

**INTRODUCTION**

**1.1 OVERVIEW**

A smoke detector is a device that senses smoke, typically as an indicator of fire. Smoke detectors are usually housed in plastic enclosures, typically shaped like a disk about 150 millimetres (6 in) in diameter and 25 millimetres(1 in) thick, but shape and size vary. Smoke can be detected either optically (photoelectric) or by physical process (ionization). Detectors may use one or both sensing methods. Sensitive alarms can be used to detect and deter smoking in banned areas. Smoke detectors in large commercial and industrial buildings are usually connected to a central fire alarm system.

Household smoke detectors, also known as smoke alarms, generally issue an audible or visual alarm from the detector itself or several detectors if there are multiple devices interlinked. Household smoke detectors range from individual battery-powered units to several interlinked units with battery backup. With interlinked units, if any unit detects smoke, alarms will trigger at all of the units. This happens even if household power has gone out.

Commercial smoke detectors issue a signal to a fire alarm control panel as part of a fire alarm system. Usually, an individual commercial smoke detector unit does not issue an alarm; some, however, do have built-in sounders.

* 1. **PROBLEM STATEMENT**

Fire incidents are a critical issue, especially when it comes to hospitals, industrial and residential areas. According to the National Crime Records Bureau (NCRB), 11,037 fire incidents were reported in India, which caused 10,915 deaths in the year 2019. Another analysis shows that emergency fire services have a 90% shortage of staff, a 78% shortage of firefighting equipment or vehicles, and a 61% shortage of fire stations, as against a 2012 requirement according to the home minister's reply to the Rajya Sabha (2018).In a general scenario of a fire accident, a person reports a fire incident, the fire department confirms the location and severity, and then the fire department responds to the emergency, which results in even more delay.

This study focuses on building an IoT based advanced fire alarm that could directly alert the nearest fire station of a possible fire accident without any human intervention needed, considering the considering the challenges mentioned above.

**1.3 EXISTING SYSTEM**

In this system, collection of training data is performed with the help of IOT devices since the goal is to develop AI based smoke detector device. The data is collected from the fire department and used to discover the patterns with the existed machine learning models. The existing systems of predicting the smoke are K nearest neighbor’s, support vector machines (SVM) & Bayes. The results are compared for performance and accuracy with these machine learning algorithms.

**1.4 PROPOSED SYSTEM**

The proposed work predicts smoke by exploring the above mentioned four classification algorithms and does performance analysis. The objective of this study is to effectively predict if there is smoke or not. The data is fed into model which predicts the probability of ringing the fire alarm. Figure shows the entire process involved.

Training Data🡪 Classification techniques

Data Extraction🡪Classification🡪 🡪Result🡪End

Testing Data🡪 Test the model

**1.5. OBJECTIVES**

The main objective of this research is to develop a smoke detection system. The objective of this project is to build a model that can accurately detect the presence or absence of smoke. This study focuses on building an IoT based advanced fire alarm that could directly alert the nearest fire station of a possible fire accident without any human intervention needed. It helps to avoid human business and also reduce the damage and gives accurate result whether there is fire or not.

**1.6 ARCHITECTURE**

The architecture of the proposed system is as displayed below

# DATA EXTRACTION 🡪 EXPLORING DATA ANALYSIS 🡪 DATA PRE-PROCESSING 🡪 TRANING MODEL 🡪 TESTING MODEL 🡪 BUILDING THE MODELS 🡪 RESULT 🡪 FIRE OR NOT 🡪END.

**Smoke Detector:**

**LITERATURE SURVEY**

It is necessary to have a strong fire safety management system in residential as well as workplace environments. Fire safety has been overlooked from time to time, but having a simple safety arrangement can minimize and even prevent possible damage to a small risk.

Researchers [1], in their review, suggested using an MWIR camera-based airborne fire detection system to detect forest and wildland fires. An MWIR based sensor is good at detecting flames and heat through air currents when a general location of the fire is known, but it fails to achieve constant monitoring needs in urban environments.

Researchers [2] suggested optimizing the redevelopment cost of urban areas to reduce fire susceptibility. This approach leaves the traditional methods altogether and focuses on long term redesigning the settlements. This method could be highly effective as a long term solution. However, it does not meet current needs, nor is it immune to inevitable demographic and geographic changes in future.

Researchers [3] supports a fire detection system designed using MQ-135 (CO2), MQ-2 (smog), MQ-7 (CO) and DHT-11 (temperature) sensors embedded with Arduino to get the fire event information in the surrounding area. It uses K-Nearest Neighbour and decision tree algorithms to determine surrounding conditions as fire, maybe fire or no fire. It shows an impressive accuracy of 93.15%, though the sample test size of 20,000 cases is questionable.

In another research, Researchers [4] proposed the construction of a wireless fire alarm system based on 2 ZigBee (IEEE 802.15.4) to achieve low cost, low power alarm systems. The author also demonstrated the implementation of ZigBee and how to achieve low power consumption while using wireless technology.

Researchers[5] in their research mentioned a selection method in the temporal analysis phase to build a video processing based alarm system for deployment in wild-fire prone areas.

**DATA PRE-PROCESSING**

**3.1 DATASET DESCRIPTION**

***About the dataset:***

This dataset consists of 60,000 readings of temperature, humidity, pressure, particulate matter, concentrations of compounds such as Hydrogen (H2), Ethanol and Carbon Dioxide (CO2), etc. taken using a set of different types of sensors from various indoor and outdoor locations. Data collection from several different locations along with the use of various different sensors provide us with a diverse set of features and data points to predict the presence or absence of smoke.

Collection of training data is performed with the help of IOT devices since the goal is to develop AI based smoke detector device.  
 Many different environments and fire sources have to be sampled to ensure a good dataset for training. A short list of different scenarios which are captured:

* Normal indoor
* Normal outdoor
* Indoor wood fire, firefighter training area
* Indoor gas fire, firefighter training area
* Outdoor wood, coal, and gas grill
* Outdoor high humidity etc.

The dataset is nearly 60,000 readings long. The sample rate is 1Hz for all sensors. To keep track of the data, a UTC timestamp is added to every sensor reading.

The above dataset has 16 columns and 62,630 rows & the data set is considered as a classification because the target variable gives 1 if fire is present else it is 0.

***Target:***

**1.Target Variable:**Fire Alarm

**2.Fire Alarm:**(Reality) If fire was present then value is 1 else it is 0.

***Feature Description:***

**1- UTC:** The time when experiment was performed.

**2- Temperature:** Temperature of Surroundings. Measured in Celsius

**3- Humidity:** The air humidity during the experiment.

**4- TVOC:** Total Volatile Organic Compounds. Measured in ppb (parts per billion)

**5- eCo2:** CO2 equivalent concentration. Measured in ppm (parts per million)[¶](https://www.kaggle.com/code/hossamgalal68/smoke-detection-eda-score-100)

**6- Raw H2:** The amount of Raw Hydrogen present in the surroundings.

**7- Raw Ethanol:** The amount of Raw Ethanol present in the surroundings.

**8- Pressure:** Air pressure. Measured in hPa

**9- PM1.0:** Particulate matter of diameter less than 1.0 micrometer.

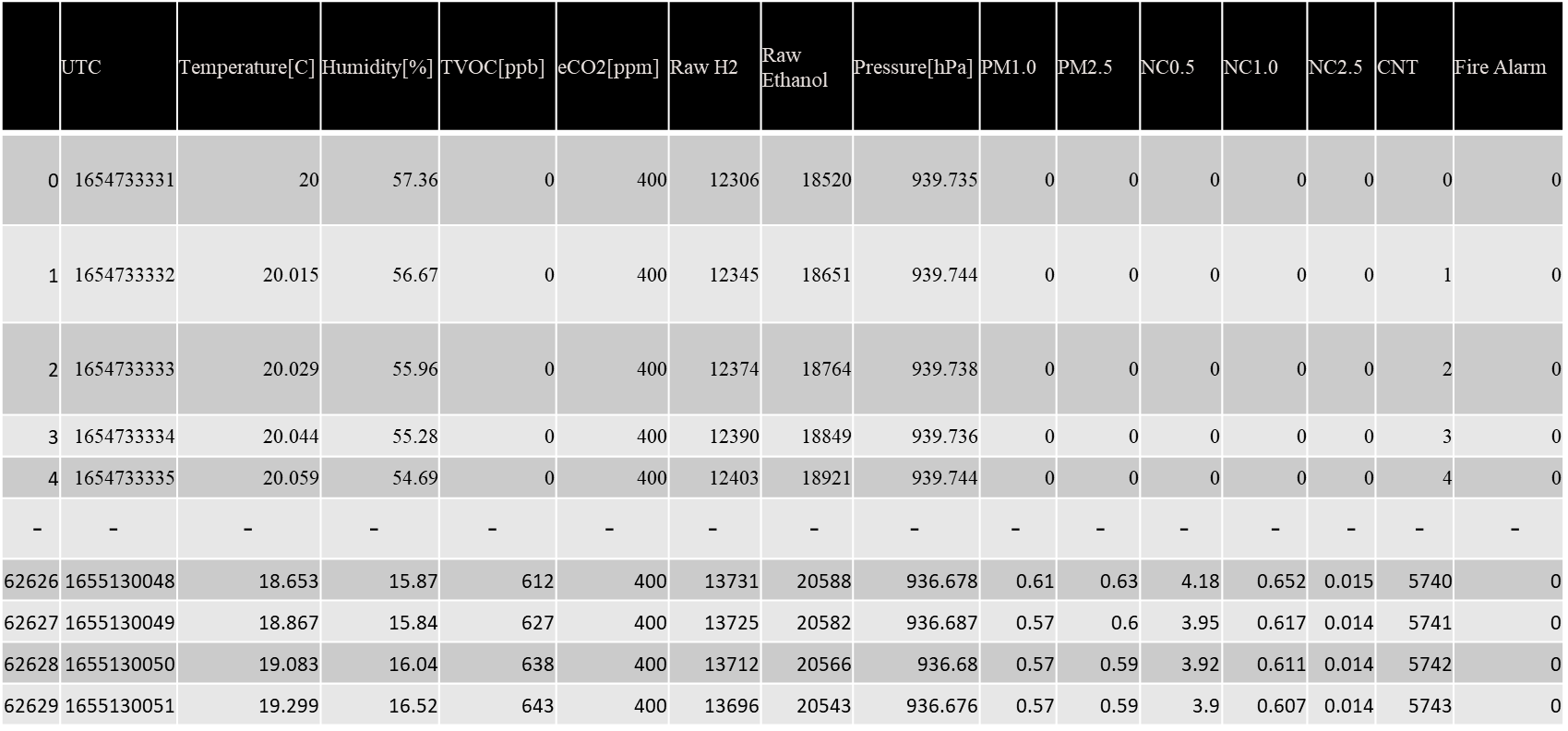
**10- PM2.5:** Particulate matter of diameter less than 2.5 micrometer.

**11- NC0.5:** Concentration of particulate matter of diameter less than 0.5 micrometers.

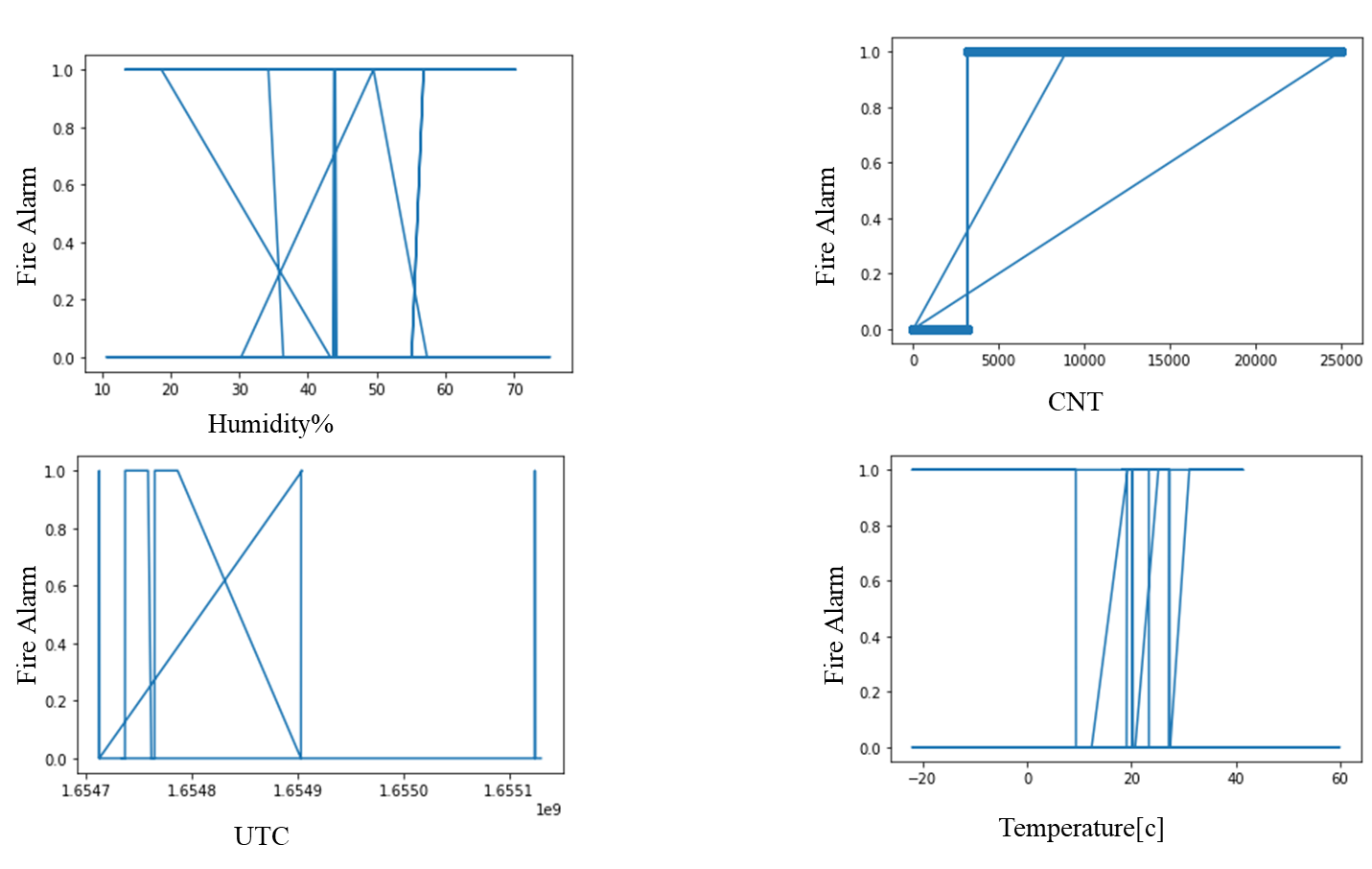
**12- NC1.0:** Concentration of particulate matter of diameter less than 1.0 micrometers.

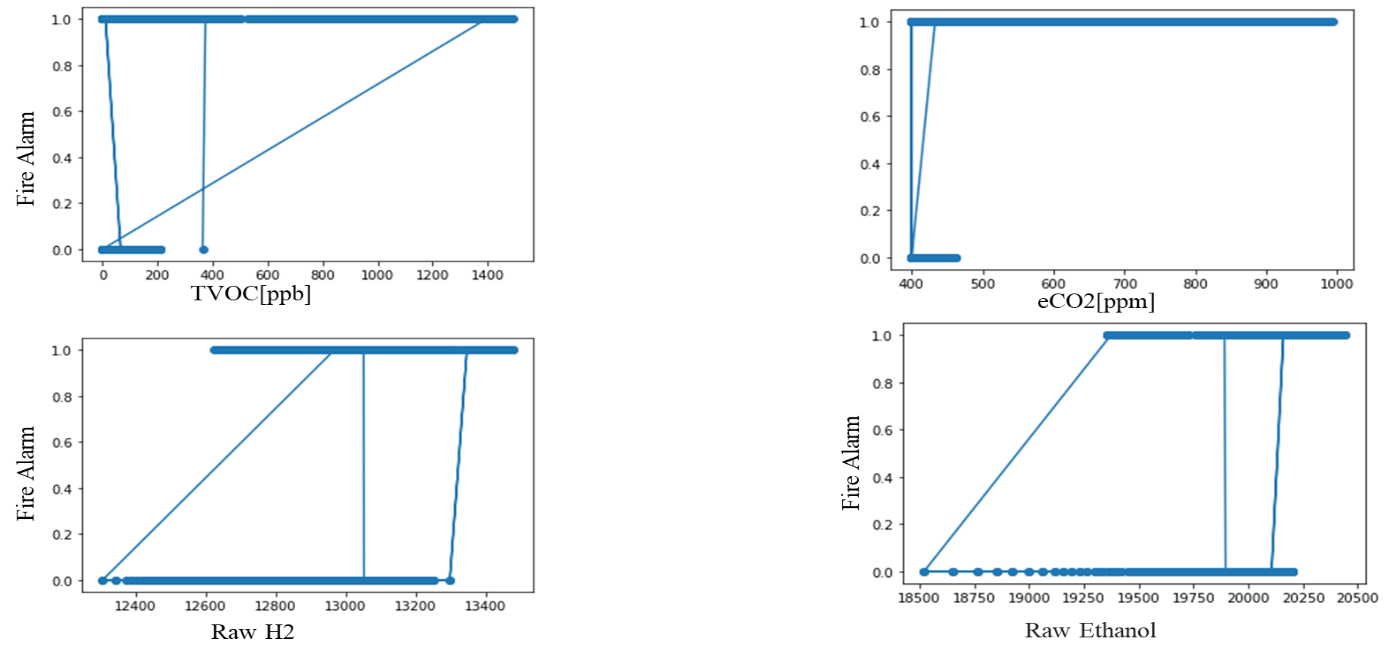
**13- NC2.5:** Concentration of particulate matter of diameter less than 2.5 micrometers.

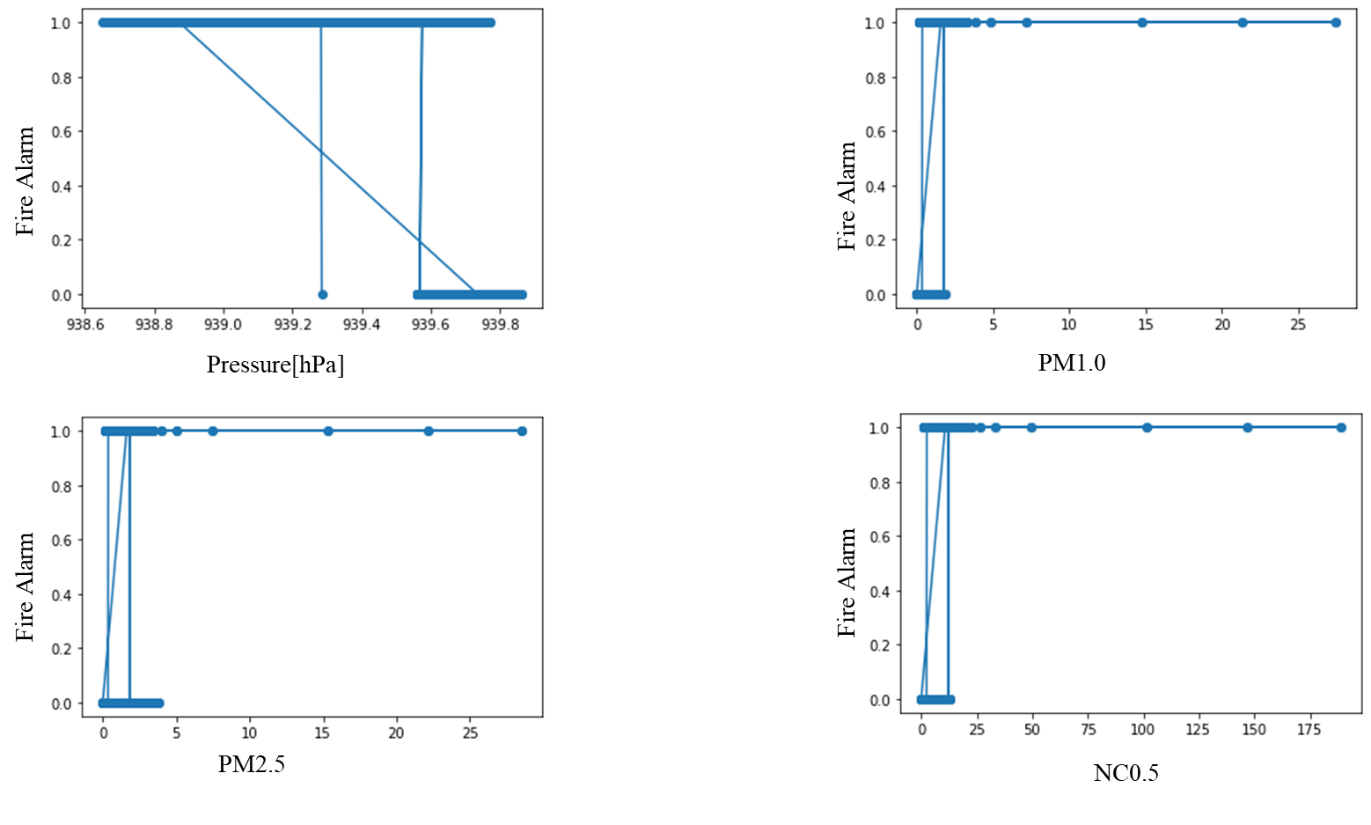
**14- CNT:** Simple Count.

**DATASET:**

**GRAPHS PLOTTED BETWWEN FEATURE AND TARGET VARIABLES:**





****

**3.2 DATA CLEANING**

Data quality has become an important issue. This issue becomes more and more important, where the need for effective decision making is high. In this context, the need for data cleaning to improve data quality is becoming crucial. Duplicate records elimination is a challenging data cleansing task. Here, we present a duplicate records elimination approach to improve the quality of data. We propose a deep learning-based approach for duplicate records detection using a sentence embeddings model. Also, we propose an algorithm for duplicated records correction. Then, we apply the proposed duplicate records elimination approach to analyse the effect of data cleaning on the quality of decisions.

In sum: There are total of 62360 rows and 16 columns in data. Data contains no missing value. We drop UTC, Unnamed 0, CNT attributes as they are of no use to us. After all the modifications we have total of 13 attributes on which we will perform EDA.

**3.4 DATA VISUALISATION**

The smoke detection data set contains 14 feature variables and 1 target variable output which contains symptoms of as features and whether fire alarm rings or not as target.

**METHODOLOGY**

**4.1 PROCEDURE TO SOLVE THE GIVEN PROBLEM**

Here in this project we can use these machine learning algorithms to detect smoke,

* KNN (K-Nearest neighbours).
* SVM (Support Vector Machine).

**KNN (K-Nearest neighbours):**

* K-Nearest Neighbor (KNN) Algorithm for Machine Learning
* K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
* K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
* K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
* K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
* K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
* It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
* KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
* **The K-NN working can be explained on the basis of the below algorithm:**

**Step-1:** Select the number K of the neighbors

**Step-2:** Calculate the Euclidean distance of K number of neighbors

**Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.

**Step-4:** Among these k neighbors, count the number of the data points in each category.

**Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.

**Step-6:** Our model is ready

**SVM (Support Vector Machine):**

* Support Vector Machine Algorithm
* Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.
* The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.
* SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine
* **Types of SVM**

SVM can be of two types:

**Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

**Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

**4.2 MODEL ARCHITECTURE:**

LOADING DATA SET 🡪 IDENTIFYING THE ATTRIBUTES FINDING THE RISK OF FIRE 🡪 COLLECTION OF DATA AND PRE -PROCESSING 🡪 K-NEAREST NEIGHBOURS OR SVM 🡪 ARTIFICIAL NEURAL NETWORKS 🡪 OBTAIN RESULTS 🡪 CONCLUSION

**4.3 SOFTWARE DESCRIPTION**

**Software requirements:**

**Operating system:** Windows 11

**Platform**: google colab.

**Programing language:** python

**RESULTS**

* **Code**

import numpy as np

import pandas as pd

smoke = pd.read\_csv(“ /content/smoke\_detection\_iot[1].csv”)

smoke.head()

smoke.info()

smoke.drop(['Unnamed: 0', 'UTC'], axis = 1, inplace = True)

Total = smoke.isnull().sum().sort\_values(ascending = False)

Percent = (smoke.isnull().sum()\*100/smoke.isnull().count()).sort\_values(ascending = False)

missing\_data = pd.concat([Total, Percent], axis = 1, keys = ['Total', 'Percentage of Missing Values'])

missing\_data

smoke.describe().T

import matplotlib.pyplot as plt

import seaborn as sns

fig, ax = plt.subplots(nrows = 5, ncols = 3, figsize = (15, 20))

for feature, subplot **in** zip(smoke, ax.flatten()):

if feature == 'Fire Alarm':

continue

sns.boxplot(x = smoke['Fire Alarm'], y = smoke[feature], showfliers = False, ax = subplot)

subplot.set\_xticks([0, 1], ['No Alarm', 'Alarm'])

plt.suptitle('Feature Correlations with Target Variable (Fire Alarm)', fontsize = 20)

plt.tight\_layout(pad = 2)

plt.show()

corr = smoke.corr()

corr

plt.figure(figsize = (14, 12))

sns.heatmap(corr, annot = True, fmt = '.2f')

plt.title('Correlation Heatmap', fontsize = 20)

plt.show()

y = smoke['Fire Alarm']

x = smoke.drop('Fire Alarm', axis = 1)

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.2, random\_state = 10)

X\_train.shape, X\_test.shape, y\_train.shape, y\_test.shape

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.fit\_transform(X\_test)

sns.countplot(x = y\_train)

plt.text(x = 0 - 0.1, y = y\_train.value\_counts()[0] + 500, s = y\_train.value\_counts()[0])

plt.text(x = 1 - 0.1, y = y\_train.value\_counts()[1] + 500, s = y\_train.value\_counts()[1])

plt.xticks([0, 1], ['No Alarm', 'Alarm'])

plt.ylabel('Count')

plt.tight\_layout(pad = -1)

plt.title('Class Imbalance', fontsize = 15)

plt.show()

from sklearn import metrics

from sklearn.metrics import classification\_report, confusion\_matrix, roc\_auc\_score, roc\_curve

def plot\_confusion\_matrix(y\_test, y\_pred, model\_name):

cm = confusion\_matrix(y\_test, y\_pred)

conf\_matrix = pd.DataFrame(data = cm,

columns = ['Predicted: No Alarm', 'Predicted: Alarm'],

index = ['Actual: No Alarm', 'Actual: Alarm'])

sns.heatmap(conf\_matrix, annot = True, cbar = False, fmt = 'd', linewidth = 0.5, annot\_kws = {'size': 25})

plt.xticks()

plt.yticks()

plt.title(model\_name + " Confusion Matrix", fontsize = 20)

plt.show()

score\_card = pd.DataFrame(columns = ['Model', 'Precision Score', 'Recall Score', 'f1-Score', 'AUC Score', 'Accuracy Score'])

index\_iter = iter(range(1, 10))

def update\_score\_card(y\_test, y\_pred, model\_name):

global score\_card

new\_score = pd.DataFrame({'Model': model\_name,

'Precision Score': metrics.precision\_score(y\_test, y\_pred),

'Recall Score': metrics.recall\_score(y\_test, y\_pred),

'f1-Score': metrics.f1\_score(y\_test, y\_pred),

'AUC Score': roc\_auc\_score(y\_test, y\_pred),

'Accuracy Score':metrics.accuracy\_score(y\_test, y\_pred)},

index = [next(index\_iter)])

score\_card = pd.concat([score\_card, new\_score])

return score\_card

### **#K-Nearest Neighbours (KNN)**

from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n\_neighbors = 5)

knn = knn.fit(X\_train, y\_train)

y\_pred\_knn = knn.predict(X\_test)

print(classification\_report(y\_test, y\_pred\_knn))

plot\_confusion\_matrix(y\_test, y\_pred\_knn, 'K-Nearest Neighbours (KNN)')

update\_score\_card(y\_test, y\_pred\_knn, 'K-Nearest Neighbours (KNN)')

### # **Support Vector Classifier (SVC)**

search\_parameters = {

'gamma' : ['scale', 'auto'],

'C': [12, 13, 14]

}

from sklearn.model\_selection import GridSearchCV

from sklearn.svm import SVC

svc\_init = SVC(kernel = 'rbf')

svc\_grid = GridSearchCV(estimator = svc\_init, param\_grid = search\_parameters, scoring = 'accuracy', cv = 5)

svc\_grid.fit(X\_train, y\_train)

svc\_grid.best\_params\_

svc = SVC(kernel = 'rbf', gamma = svc\_grid.best\_params\_['gamma'], C = svc\_grid.best\_params\_['C'])

svc.fit(X\_train, y\_train)

y\_pred\_svc = svc.predict(X\_test)

print(classification\_report(y\_test, y\_pred\_svc))

plot\_confusion\_matrix(y\_test, y\_pred\_svc, 'Support Vector Classifier (SVC)')

**output:**

| Model | Precision Score | Recall Score | f1-Score | AUC Score | Accuracy Score |
| --- | --- | --- | --- | --- | --- |
| K-Nearest Neighbours (KNN) | 0.999658 | 0.999486 | 0.999572 | 0.999464 | 0.999469 |
| Support Vector Machine (SVM) | 0.999658 | 0.999658 | 0.999550 | 0.999550 | 0.999576 |

Now we have the final Accuracy, which says that SVM is the model which is Performing the

best.

****

**CONCLUSION AND FUTURE SCOPE**

This paper discusses the various machine learning such as k-nearest neighbour & svm which were applied to the data set. It utilizes the data such as temperatue, humidity ..etc then tries to predict the whether the fire is present or not.

In this paper, an IoT based fire alarm system is designed. It provides a message alert to the owner and fire station in case of emergency. This research differentiates states of the surroundings in terms of sleep, passive or active state using a decision tree algorithm. Hence a low cost, IoT system is developed which is ideal as a fire safety solution in residential areas and hospitals.

In future developments, we will add diverse sensors which can monitor other harmful gases produced. We will also further develop the State Determining algorithm to increase the scalability of the system. Later we will add a self testing module which will self diagnose problems in the alarm system and alert the owner of a possible device failure.

**REFERENCES**

[1] W. Phillips III, M. Shah and N. V. Lobo, “Flame recognition in video”,Pattern Recognition Lett. 23, pp. 319–327, 2002.

[2] B. U. Toreyin, Y. Dedeoglu, U. G ¨ ud¨ ukbay and A. E. C ¨ . etin , “Computer vision based method for real–time fire and flame detection”, Pattern Recognition Lett. 27, pp. 49–58, 2006.[3] "SMOKE ALARM SAFETY TIPS". Safety Information. National Fire Protection Association. Archived from the original on 2009-08-21. Retrieved 2009-05-17.

[4] "Smoke Alarms". Hansard - Mr Christopher Gulaptis MP, Private Member's

Statements, New South Wales Parliamentary Debates, Legislative Assembly, New South

Wales, Australia 20 June 2013, pp.22218. Archived from the original on 29 October

2013. Retrieved 2013-06-26 .

[5] A. Shashank, Rajiv Vincent, Arun Kumar Sivaraman, A. Balasundaram, M. Rajesh, S. Ashokkumar, "Power Analysis of Household Appliances using IoT," International Conference on System, Computation, Automation and Networking (ICSCAN), IEEE Xplore, pp. 1-5, 2021.

[6] Sowah, Robert, et al., “Implementation of a fire detection and control system for automobiles using machine learning,” in Proceedings of Industry Applications Society Annual Meeting, 2016.

[7] Abdul Quadir Md, Christy Jackson J, Prassanna J, Shivam Ubarhande, VijayaKumar V "An Efficient Way to Detect DDoS Amplification attacks. Journal of Intelligent & Fuzzy Systems, Pre-press, pp. 1-8, 2020