conference-template-a4_(1)[1].docx

University of Engineering & Management

Document Details

Submission ID
trn:oid:::3618:123966341

Submission Date
Dec 8, 2025, 9:01 AM GMT+5:30

Download Date
Dec 8, 2025, 9:03 AM GMT+5:30

File Name
conference-template-a4_(1)[1].docx

File Size
383.2 KB

5 Pages

2,912 Words

15,838 Characters

26% Overall Similarity

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

Match Groups

72 Not Cited or Quoted 26%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

19% Internet sources
17% Publications
18% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

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

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

Match Groups

- **72** Not Cited or Quoted 26%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

19% Internet sources
17% Publications
18% Submitted works (Student Papers)

Top Sources

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

1	Publication	Iehab Alrassan, Asma Alqahtani. "Detection of DDoS Attacks on Clouds Computin... 3%
2	Internet	doctorpenguin.com 2%
3	Internet	download.bibis.ir 2%
4	Internet	iieta.org 2%
5	Submitted works	Coventry University on 2025-06-20 <1%
6	Internet	www.science.gov <1%
7	Internet	www.opastpublishers.com <1%
8	Submitted works	Coventry University on 2024-03-25 <1%
9	Submitted works	University of KwaZulu-Natal on 2015-01-12 <1%
10	Internet	flosshub.org <1%

11 Internet
eitca.org <1%

12 Internet
pmc.ncbi.nlm.nih.gov <1%

13 Internet
www.airconcept.co.in <1%

14 Submitted works
University of the West Indies on 2021-03-31 <1%

15 Publication
Indranil Maity, Souvik Bhanja. "First principle based computations to evaluate pr... <1%

16 Publication
S.P. Jani, M. Adam Khan. "Applications of AI in Smart Technologies and Manufactu... <1%

17 Internet
www.etd.dbu.edu.et <1%

18 Submitted works
Nottingham Trent University on 2025-04-03 <1%

19 Publication
Pushpa Choudhary, Sambit Satpathy, Arvind Dagur, Dhirendra Kumar Shukla. "Re... <1%

20 Publication
Xin Sui, Wenqi Wang, Chunyang Liu, Peixin Dong. "A guideline study for optimal ... <1%

21 Internet
alicia.concytec.gob.pe <1%

22 Submitted works
Liverpool John Moores University on 2023-02-26 <1%

23 Internet
kyutech.repo.nii.ac.jp <1%

24 Internet
rsisinternational.org <1%

25 Internet

www.researchsquare.com <1%

26 Submitted works

Liverpool John Moores University on 2023-03-15 <1%

27 Submitted works

University of Exeter on 2025-11-01 <1%

28 Internet

www.coursehero.com <1%

29 Internet

www.hindawi.com <1%

30 Internet

www.mdpi.com <1%

31 Publication

Ibrahem Alshybani, Farhad Jaberi, Michael S. Murillo, Yifeng Tian. "Assessment of ... <1%

32 Internet

hal.science <1%

33 Internet

web.realinfo.tv <1%

34 Submitted works

Erasmus University of Rotterdam on 2020-08-12 <1%

35 Submitted works

Bournemouth University on 2023-01-13 <1%

36 Submitted works

Letterkenny Institute of Technology on 2023-08-31 <1%

37 Publication

Marketa Zvelebil. "Understanding Bioinformatics", Garland Science, 2007 <1%

38 Publication

Samarjeet Borah, Ratna Raja Kumar Jambi, Sharifah Sakinah Syed Ahmad, Mahen... <1%

39 Internet

aiforsocialgood.ca <1%

40 Internet

jbc.bj.uj.edu.pl <1%

41 Internet

repositorio.uchile.cl <1%

42 Publication

Alfardus, Asma. "Evaluating Machine Learning for Intrusion Detection in CAN Bus... <1%

43 Publication

Sukhpreet Kaur, Amanpreet Kaur, Manish Kumar. "Recent Advances in Computati... <1%

44 Publication

Amit Kumar Tyagi. "Data Science and Data Analytics - Opportunities and Challeng... <1%

45 Publication

Arvind Dagur, Sohit Agarwal, Dhirendra Kumar Shukla, Shabir Ali, Sandhya Sharm... <1%

46 Submitted works

ICSDI on 2025-12-02 <1%

47 Submitted works

University of Maryland, Global Campus on 2024-10-08 <1%

Comparative Machine Learning-Based Classification of CO and CO_2 Using Multi-Algorithm Analysis of Gas Sensor Data

Abstract*—This paper concerns a machine learning driven approach for the discrimination of two gases, named Carbon Dioxide (CO) and Carbon Monoxide (CO_2). Detecting them is essential as both are toxic harmful environmental gases and they can lead to several diseases. Support Vector Machine (SVM), k-Nearest Neighbours (k-NN), and Random Forest (Rf) classifier were the algorithms used to analyse the sensor response from a Pt-doped Zinc Oxide (ZnO) based nanotube (NT) sensor and distinguish between the gases. The dataset was split into training and testing subsets in a 70:30 ratio. A cloud-based python 3 environment, Google Colab, was used to train, validate, and test the models. The features used for model training include adsorption energy (E_a adsorption), binding distance, energy of the highest occupied molecular orbital (E_{HOMO}), lowest unoccupied molecular orbital (E_{LUMO}), HOMO-LUMO gap, E_{HOMO} -1, E_{LUMO} +1, chemical potential (μ), electronic conductivity ratio, sensitivity, and recovery time. The target variable indicates the gas type (CO or CO_2). Principal Component Analysis (PCA) was used to visualize the data by reducing its dimension. A total of 4 quality performance metric parameters such as precision, accuracy, F1 score and recall were calculated along with their confusion matrices. Out of the 3 machine learning models, SVM provided the best accuracy of 96.67%.

Keywords*—machine learning, accuracy, support vector machine, Random forest, k-Nearest Neighbor

I. INTRODUCTION

This Carbon monoxide (CO) is a highly toxic gas that is colorless, odorless, and tasteless. It is hazardous to the environment and can be extremely harmful to human health when present at elevated concentrations [1]. Carbon dioxide (CO_2), although generally harmless at low levels, can also become dangerous when its concentration increases in closed or poorly ventilated spaces. Both CO and CO_2 are significant air pollutants commonly encountered in industrial operations, household environments, and broader environmental settings. Unsupervised exposure to elevated concentrations can pose serious health risks and may lead to life-threatening situations. [2-3] Therefore, dependable and efficient detection of these gases is critical for ensuring safety, preventing hazardous incidents, and reducing their potential impact on human health and the environment. The dataset was formed from the readings obtained by the platinum doped zinc oxide nanotube base sensor device. Given it's high surface-to-volume ratio, chemical stability, and adaptable electrical characteristics, ZnO (NTs) are great candidates for gas sensors [4]. Supervised machine learning models were employed to establish and learn the relationship between the input features and the known output labels. Once trained the models can predict the gas for new unseen sensor readings. Machine learning algorithms used are Support Vector Machine (SVM), K-Nearest Neighbour (KNN), and Random Forest. This paper uses machine learning algorithms to investigate the accuracy provided by the techniques and analyse their classification capabilities. Sensor responses for co and co2 often overlap , show variations and are influenced by physiochemical factors.the algorithms are used as they are capable of learning and identifying these complex patterns and non linear trends, they can deal with high dimensional features which can be visualised by using pca.

An extensive variety of machine learning and statistical methods have been investigated in previous research for forecasting CO_2 emissions. The application of multiple regression-based models for predicting CO_2 levels has been reported to be very successful in selecting effective algorithms for environmental forecasting [5]. Some researchers compared the projections for the next decade done by India based on the univariate time-series data spanning several decades [6]. The literature repeatedly points out that accurate CO_2 prediction is a key factor for energy planning, directing emission reduction strategies, and contributing to global sustainability [7]. In this scenario, different machine learning techniques including ordinary least squares regression, support vector machines, and gradient boosting, have been utilized for forecasting emissions related to transportation [8]. Moreover, different model performances have been compared in a study involving 14 models, including various statistical methods like ARMA, ARIMA, SARMA, and SARIMA with daily CO_2 data from the most polluted areas worldwide as input [9]. Then, another set of studies narrowed down the focus to China with the goal of finding the best near-real-time forecasting model, using univariate daily emissions data from recent days [10].

The aim of the present study is to analyze the comparison among three supervised machine learning algorithms on the basis of their performance when applied to the dataset collected from platinum-doped zinc oxide nanotube gas sensors. As a comparative study, it showcases how precisely each model is able to distinguish between the target gas labels (CO and CO_2) using the provided sensor features. For a guaranteed fair and accurate evaluation, the dataset was divided into a 70:30 ratio (70% training set, 30% testing set). The dimensionality of the dataset was then reduced using principal component analysis (PCA) and trained again for plotting the decision boundary. The models were developed and tested using common python libraries such as NumPy, Pandas, Matplotlib, and Scikit-learn. Among the three algorithms, the Support Vector Machine (SVM) performed the most

effective, achieving an accuracy of 96.67%, while k-nearest neighbor and random forest reached 95.0% and 93.3% accuracy, respectively. These results were further supported by the confusion matrix, which showed that all models demonstrated consistent and strong classification performance.

II. Classification Model Evaluation

A. Sensor Information

The sensor utilized in this work to build the dataset was derived from pristine and Pt/ZnO NT structures optimized in Gaussian 09W. Pristine ZnO had Zn, O and H atoms arranged to eliminate dangling bonds, while the Pt doped version had one Zn atom replaced by Pt which forms a stable catalytic site for the interaction of CO. Various electronic parameters were extracted from this sensor such as adsorption energy, HOMO-LUMO gap and recovery time which were later used as features to train the machine learning algorithms.

B. Support Vector Machine

The Support vector machines (SVMs), are supervised max-margin models used for classification and regression. It was developed at AT&T Bell Laboratories by Vladimir Vapnik. The goal of SVM is to find a hyperplane that separates the data points into different classes. The hyperplane is chosen in such a way that it maximizes the margin, which is the distance between the hyperplane and the closest data points of each class. The closest data points are called the support vectors.

$$\min_{(w,b,\xi)} \frac{1}{2}\|w\|^2 + c\sum_{i=1}^{N} \xi i$$

The svc module from scikit-learn library was used to implement the support vector machine (SVM). a linear kernel was used to classify the data points principal component analysis (PCA) was applied to reduce the dimension and visualize the data. The regularization parameter used was c=1.0 which was used to minimize the error. Support vector machines (SVMs), are supervised max-margin models used for classification and regression. It was developed at AT&T Bell Laboratories by Vladimir Vapnik. The goal of SVM is to find a hyperplane that separates the data points into different classes. The hyperplane is chosen in such a way that it maximizes the margin, which is the distance between the hyperplane and the closest data points of each class. The closest data points are called the support vectors.

Algorithm 1: Support Vector Machine Model

Inputs: Training data X-train, Y-train, testing data X-test, Y-test.
Outputs: Accuracy, Confusion Matrix, F1 score, Recall, Precision, RMSE, MAE
Step 1: Load dataset and split into features 'X' and target 'y'.
Step 2: Clean and preprocess the data
Step 3: Normalize the dataset using Standard Scaler function
Step 4: split dataset into training and testing sets
Step 5: Train SVM model with X-train, Y-train with linear kernel and c=1.0.
Step 6: evaluate and test the model performance
Step 7: find the other metrics-recall, f1 score, precision.
Step 8: Generate the confusion matrix
Step 9: find the errors-RMSE, MAE
Step 10: stop the algorithm
Return Accuracy, Confusion Matrix, recall, precision, F1 score, RMSE, MAE.

C. Random Forest

Random forest is an ensemble supervised machine learning algorithm introduced by Leo Breiman in 2001 to attain improved accuracy and reduced overfitting. It uses the principal of decision trees by using multiple of them during training and aggregating their outputs to produce a better and reliable prediction than a single decision tree since a single tree can be prone to high variance.

The algorithm creates each decision tree by using random parts of the data set provided, thus making every tree different. Instead of using all the features at once, it picks a few at random to decide how to split the data. Then every tree provides it's own prediction based on what it learned from the part of the data provided to it.

$$Gini = 1 - \sum_{i=1}^{n} (P_i)^2$$

Each decision tree uses either Gini Index or Entropy as a metric for impurity measurement.

Algorithm 2: Random Forest

Inputs: Training data X-train, Y-train, testing data X-test, Y-test.
Outputs: Accuracy, Confusion Matrix, F1 score, Recall, Precision, RMSE, MAE
Step 1: Load dataset and split into features 'X' and target 'y'.
Step 2: Clean and preprocess the data
Step 3: Normalize the dataset using Standard Scaler function
Step 4: split dataset into training and testing sets
Step 5: Train Random Forest model with X-train, Y-train with n-estimators = 300, max depth = 10 and min samples split =4
Step 6: evaluate and test the model performance
Step 7: find the other metrics-recall, f1 score, precision.
Step 8: Generate the confusion matrix
Step 9: find the errors-RMSE, MAE
Step 10: stop the algorithm
Return Accuracy, Confusion Matrix, recall, precision, F1 score, RMSE, MAE.

D. *k-Nearest Neighbors*

k-NN is a supervised machine learning algorithm devised in the early years of pattern recognition research. It works on the philosophy that data points with similar characteristics appear near each other in a feature space and likely belong to the same category. It is non-parametric and instance based in nature i.e. it does not make an explicit training model but instead stores the entire dataset during prediction.

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

k-NN considers the k number of nearest datasets (neighbors) to the test point and the majority class found from the neighbors is the predicted output for the test point. to find the nearest neighbors, the algorithm calculates the distance typically using the Euclidean Distance metric.

Algorithm 3 : k-Nearest Neighbors

Inputs: Training data X-train, Y-train, testing data X-test, Y-test.
Outputs: Accuracy, Confusion Matrix, F1 score, Recall, Precision, RMSE, MAE
Step 1: Load dataset and split into features 'X' and target 'y'.
Step 2: Clean and preprocess the data
Step 3: Normalize the dataset using Standard Scaler function
Step 4: split dataset into training and testing sets
Step 5: Train k-NN model with X-train, Y-train with k=4
Step 6: evaluate and test the model performance
Step 7: find the other metrics-recall, f1 score, precision.
Step 8: Generate the confusion matrix
Step 9: find the errors-RMSE, MAE
Step 10: stop the algorithm
Return Accuracy, Confusion Matrix, recall, precision, F1 score, RMSE, MAE.

III. METHODOLOGY

The dataset was formed from the experimental values provided by the Pt/ZnO NT based sensor devices designed for the detection of toxic CO and CO_2. ZnO NT are a prominent class of nanostructures with considerable potential in various sensing applications because of their distinctive chemical and physical characteristics [3].The features taken into considerations are $E_a$adsorption, binding distance, E_{HOMO}, E_{LUMO}, (E_{HOMO} -1), (E_{LUMO} + 1), chemical potential (μ), electronic conductivity ratio, sensitivity, and recovery time and target labels are the gases CO and CO_2. A total of 200 rows of data was used. The dataset was cleaned and verified to ensure that no missing values were present. The labels were assigned as 0 for CO and 1 for CO_2. The code was written in google colab supporting python 3 programming language. The dataset was divided into training and testing subsets in the ratio 70:30. To visualize the data, principal component analysis (PCA) was used. All the features were scaled using the standard scaler function and then PCA was applied to reduce the high dimensional dataset into two principal components. These two principal components captured the maximum variance in a 2D representation. The ratio of PCA1:PCA2 for Rf, SVM and k-NN are [0.24508241 0.10788231], [0.24508241 0.10788231], [0.24508241 0.10788231] respectively.

SVM, Rf and k-NN were applied to the training dataset. All the models were evaluated based on their accuracy and precision, recall, F1 score and confusion matrix.

IV. RESUTLS AND DISCUSSIONS

This study aimed to provide a comparison between 3 ML algorithms using the data extracted from a Pt/ZnO NT based gas sensor to predict the type of gas provided (CO or CO_2).

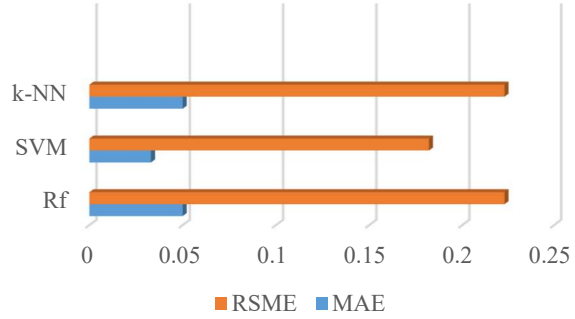

Fig. 1 : bar graph representing the comparison of errors among k-NN, SVM and Rf

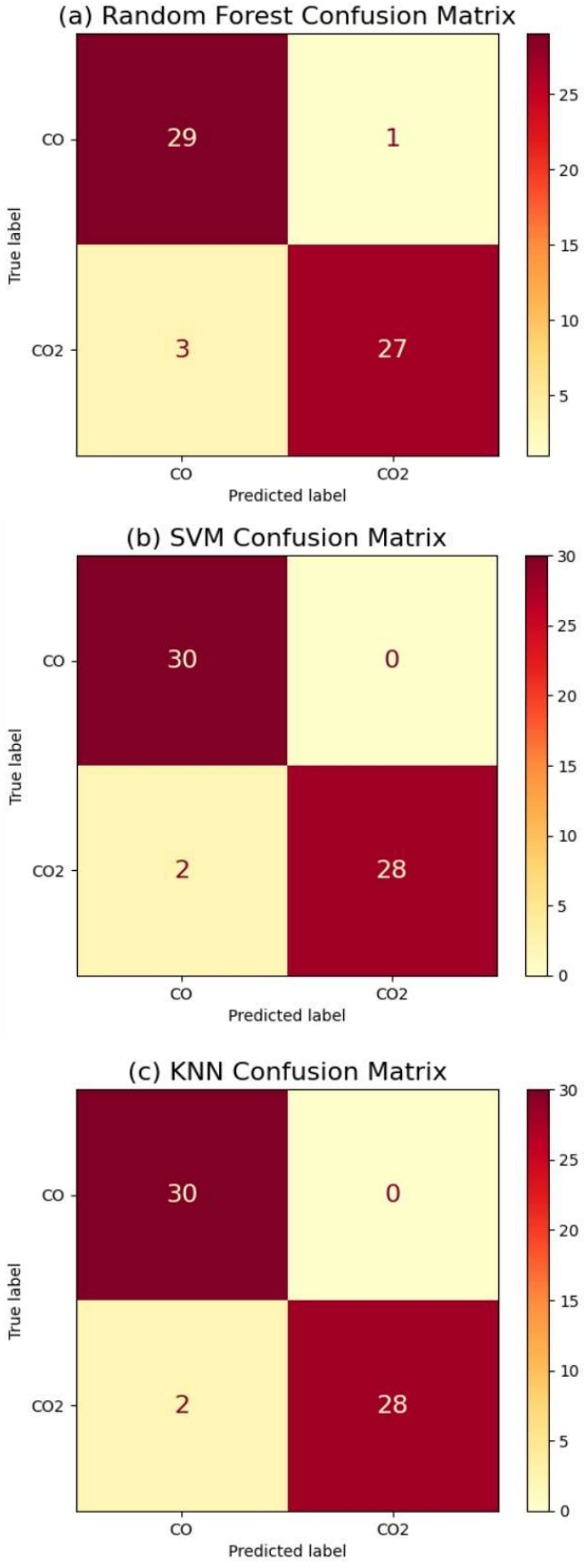

Fig. 2 (a) Heatmap of the confusion matrix obtained from the Random Forest model, (b) Heatmap of the confusion matrix obtained from the SVM model, (c) Heatmap of the confusion matrix obtained from the k-NN model.

METRIC	RANDOM FOREST		SUPPORT VECTOR MACHINE		k-NEAREST NEIGHBOR	
Gas	CO	CO2	CO	CO2	CO	CO2
Recall	0.97	0.90	1.0	0.93	1.0	0.90
Precision	0.91	0.96	0.94	1.0	0.91	1.0
F1-score	0.94	0.93	0.97	0.97	0.95	0.95

Table. I. Table showing distinction of the ML algorithms on the basis of performance metrics

In Table. I, it was observed that from the performance metric parameters, SVM provides the best results in all calculated metrics and for both gases, with k-NN being just slightly worse followed by the Rf classifier.

Further, Fig. 1, showcasing the error metrics calculated for each of the classification algorithms further provides us similar results, where SVM provided the least amount of Mean Absolute Error and Root Mean Square Error both followed by both Rf and k-NN which provided equal errors at the 70% testing data set.

With the help of the confusion matrix, different performance analysis parameters have been calculated namely precision which stresses on the quality of the correctly predicted values. it gives information about the reliability of the model. Recall focuses on the coverage of the model. it generally tells us how many actual samples of co or co2 were successfully identified as CO or CO_2 respectively. F1 score is harmonic mean of recall and precision. it combines recall and precision to give a single value.

Figure 2(a-c) showcases the confusion matrices for the 3 machine learning models which help us to determine the performance of the classification models. The confusion matrix is divided into four different quadrants, namely true positive (TP), true negative (TN), false positive (FP) and false negative (FN). TP and TN provide the correctly predicted malignant and benign cases respectively, on the other hand, FP and FN show the incorrectly predicted malignant and benign cases. The provided heatmap in Fig. 1(a-b) gives a visual representation for highlighting TP, TN, FP and FN [15]

In Fig. 3(a-c) the decision boundary graphs were plotted for all the three algorithms. It visualizes how accurately the classifier separates the two classes CO and CO_2 using PCA. PCA was used to ensure the dimensionality reduction for simpler visualization. The random forest produces a mostly vertical, piecewise boundary showing it forms several linear splits to separate the two classes. it illustrates a clear division between CO (blue) and CO_2 (red) with minor misclassification caused by natural randomness of ensemble. On the other hand SVM performs the best and provides the cleanest, smoothest separation by maximizing margin between CO and CO_2. the hyperplane is almost linear indicating best performance among three. The k-NN algorithm also produces a flexible decision boundary bending according to the local patterns, though being more noise susceptible.

	80%		70%		60%	
	ACCURACY (%)	TIME (S)	ACCURACY (%)	TIME (S)	ACCURACY(%)	TIME(S)
RF	92.5	17.32	93.34	17.711	88.75	12.59
SVM	95.0	9.65	96.67	8.44	95.0	8.81
K-NN	95.0	13.74	95.0	11.25	9.0	9.75

Table. II, accuracy and execution time of ML models on 60,70 and 80% training set

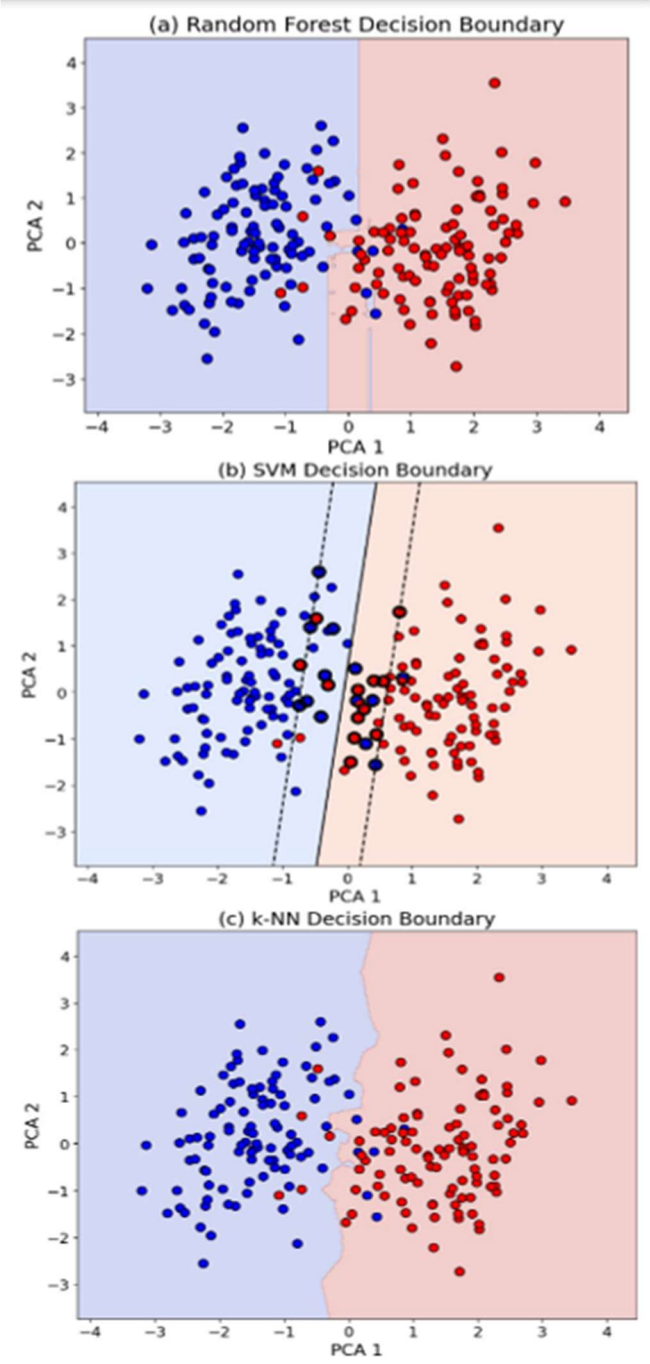

Figure 3(a) Scatter plot of PCA applied testing data set of Random Forest, (b) Scatter plot of PCA applied testing data set of SVM, (c) Scatter plot of PCA applied testing data set of k-NN.

Table II depicts the comparison of accuracy and time taken for execution among different percentages of samples used in training the SVM, Random Forest and k-Nearest Neighbors classifiers. It was observed that, out of the 3 training to test set ratios, the 70% training data set proved itself to be the most accurate overall. As observed, the execution time for the 60% training set is slightly higher but it is an acceptable trade-off for better accuracy provided by the 70% training set. It was also noticed that SVM provided the better accuracy and the fastest execution speed in all fields provided, thus we can that SVM is overall better.

V. CONCLUSION

This paper presents a comparative analysis of the machine learning based classification of CO and CO_2 gases. The ML algorithms used are Random Forest, Support Vector Machine and k-Nearest Neighbors. The dataset was divided into multiple training sets of 60, 70 and 80%, where 70% training was found to be the best with an accuracy of 93.34%, 96.67% and 95.0% for Rf, SVM and k-NN respectively. PCA was used to reduce the dimension of the data in order to make the visualization simpler. Two principal components were selected for a 2D visualization with maximum variance ratio being [0.24508241 0.10788231] for SVM compared to [0.24508241 0.10788231] and [0.24508241 0.10788231] for Rf and k-NN respectively. SVM also recorded the least RMSE and MAE of 0.1826 and 0.034 relative to Rf and k-NN. Overall, SVM performed the best in this case because it is robust to high dimensional features spaces and handles imbalance of classes using margin maximization effectively, whereas the performance of k-NN drops in performance because of the curse of dimensionality. Similarly, the accuracy of Rf is lowered due to the overfitting caused by the small, highly corelated sensor dataset.