Classification of Gas-Sensor Array Data using Machine Learning

Major Project
School of Mathematics

Project Supervisors
Dr. Dharmatti Sheetal
Dr. Vinayak B. Kamble

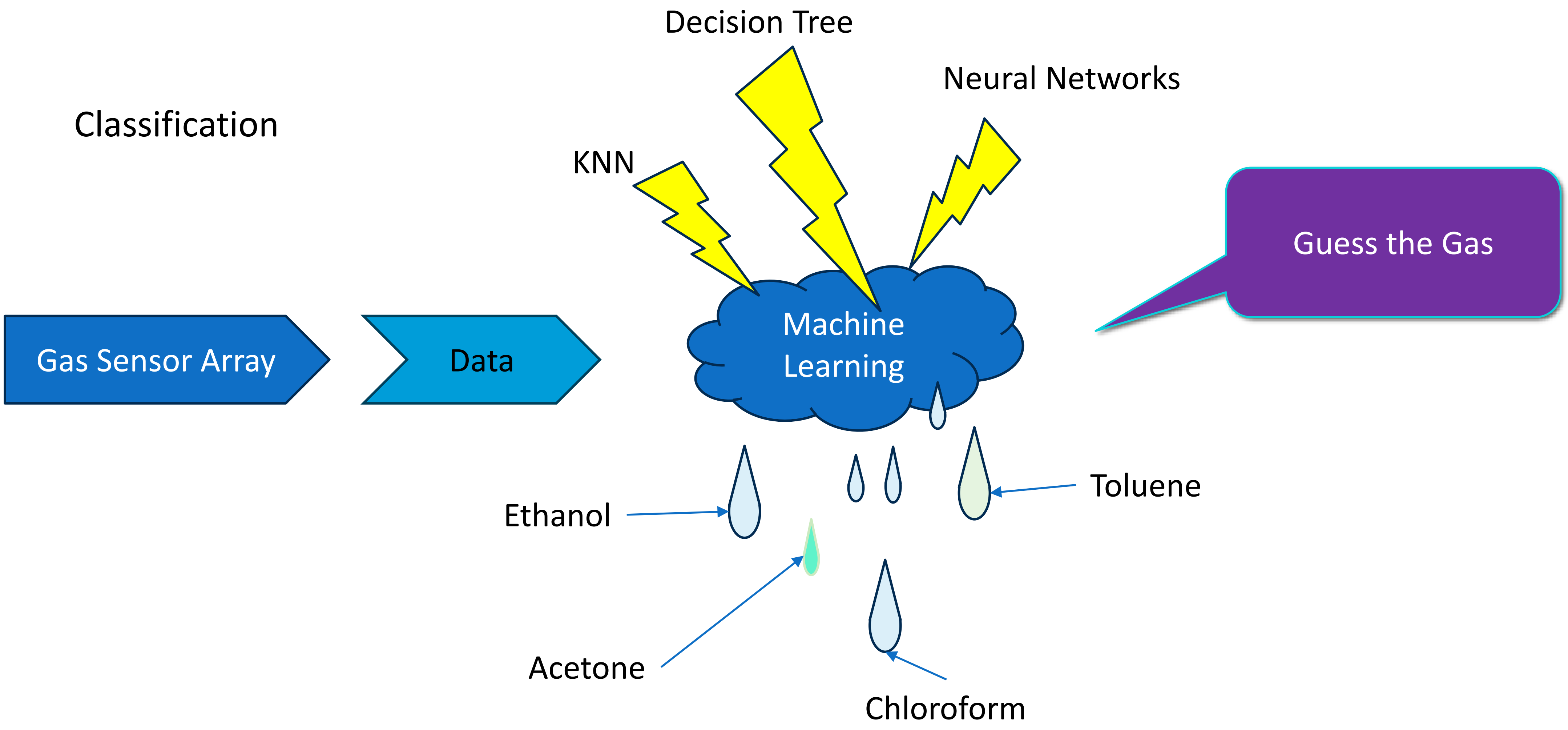
IISER
THIRUVANANTHAPURAM

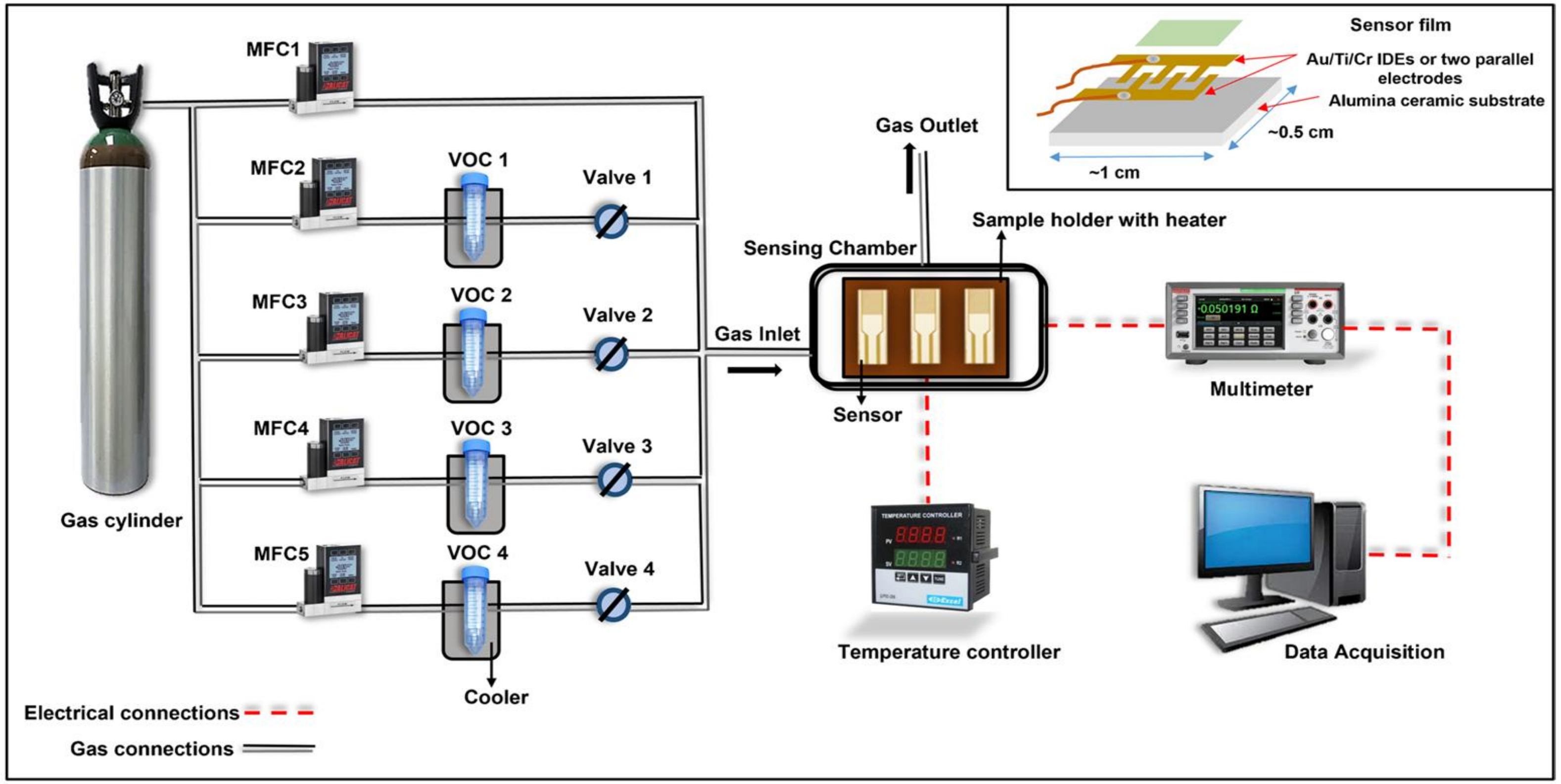
Prashant Sharma
IMS19175

29-04-2024

The Goal

Classification





Volatile Organic Compound(VOC)

	Resistance	Sensor	Temp	Conc	Gas
0	2032.37575	0	200.0	2400	2
1	2032.42944	0	200.0	2400	2
2	2032.60295	0	200.0	2400	2
3	2032.66493	0	200.0	2400	2
4	2032.65253	0	200.0	2400	2
...	...	...	...	...	...
2262999	100272.94300	2	350.0	1000	0
2263000	100295.06810	2	350.0	1000	0
2263001	100314.88840	2	350.0	1000	0
2263002	100335.92470	2	350.0	1000	0
2263003	100356.66760	2	350.0	1000	0

2263004 rows × 5 columns

The schematic diagram of the gas sensing setup used for the experiment

S. Singh et al. "Metal oxide-based gas sensor array for VOCs determination in complex mixtures using machine learning". In: Microchimica Acta 191.4 (2024). Published online: March 13, 2024, pp. 1–10. doi: 10.1007/s00604 024-06258-8.

Resistance

$(10^3 - 10^6\ \Omega)$

Sensor

1. NiO-Au(ohmic)
2. CuO-Au(Schottky)
3. ZnO-Au(Schottky)

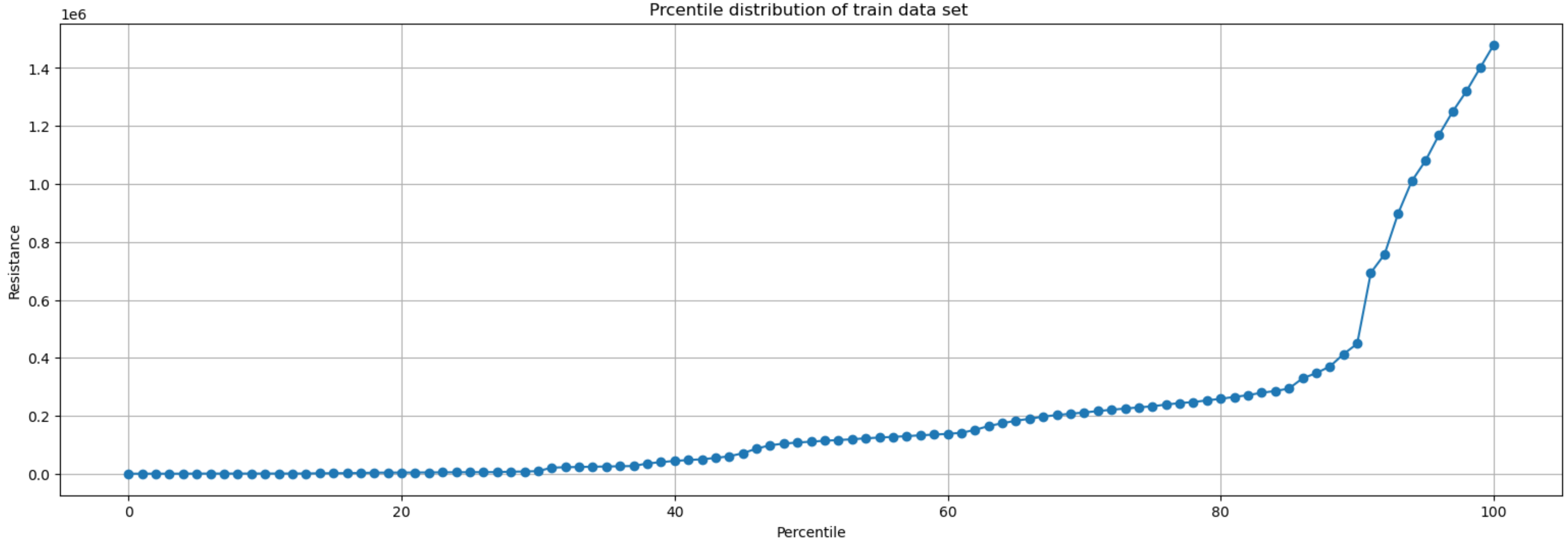
Temperature

Concentration

Gas

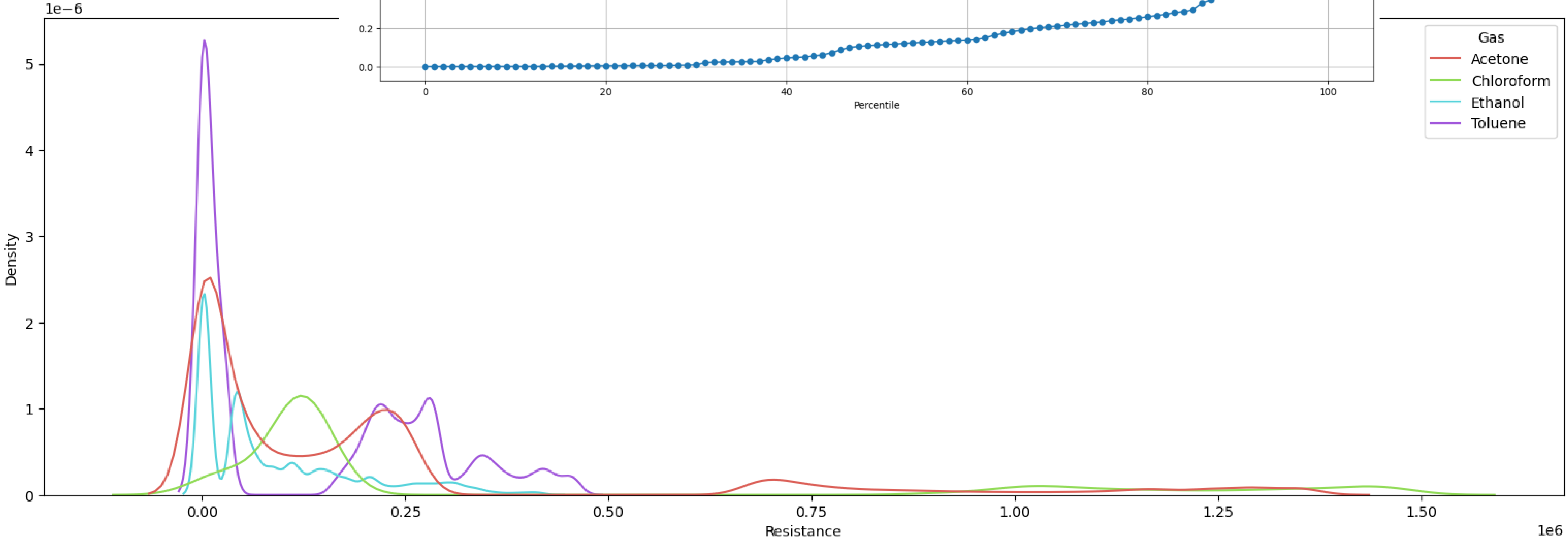
1. Ethanol
2. Acetone
3. Toluene
4. Chloroform

Preprocessing



22L points

K-Density Plot



Preprocessing

Scaling (0-1)

Input features

	Resistance	Sensor	Temp	Conc	Gas
0	2032.37575	0	200.0	2400	2
1	2032.42944	0	200.0	2400	2
2	2032.60295	0	200.0	2400	2
3	2032.66493	0	200.0	2400	2
4	2032.65253	0	200.0	2400	2
...	...	...	...	...	...
2262999	100272.94300	2	350.0	1000	0
2263000	100295.06810	2	350.0	1000	0
2263001	100314.88840	2	350.0	1000	0
2263002	100335.92470	2	350.0	1000	0
2263003	100356.66760	2	350.0	1000	0

2263004 rows × 5 columns

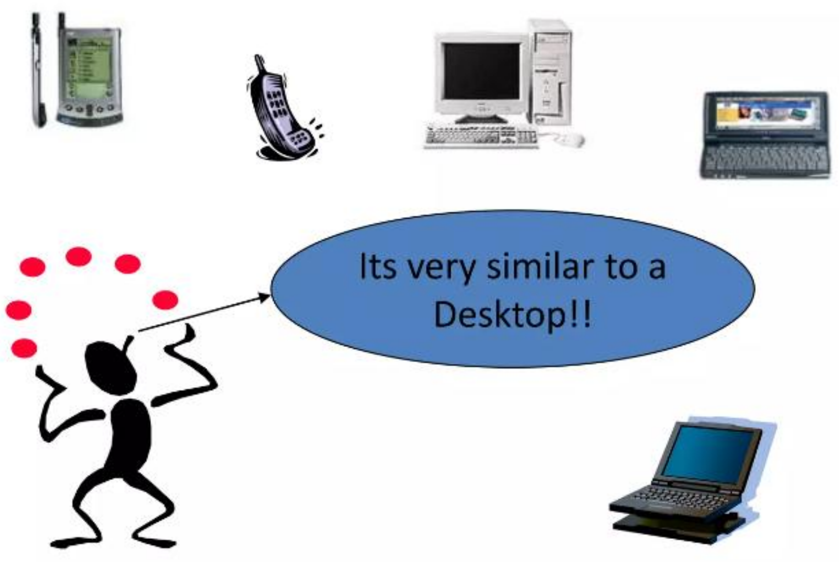
Min-Max Scaler

	Resistance	Sensor	Temp	Conc	Gas
0	0.000789	0	0.0	0.888807	2
1	0.000789	0	0.0	0.888807	2
2	0.000789	0	0.0	0.888807	2
3	0.000789	0	0.0	0.888807	2
4	0.000789	0	0.0	0.888807	2
...	...	...	...	...	...
2262999	0.067206	2	1.0	0.369904	0
2263000	0.067221	2	1.0	0.369904	0
2263001	0.067235	2	1.0	0.369904	0
2263002	0.067249	2	1.0	0.369904	0
2263003	0.067263	2	1.0	0.369904	0

2263004 rows × 5 columns

KNN(K Nearest Neighbor)

- non-parametric, lazy learning algorithm
- classification and regression tasks.
- it doesn't build a model explicitly. Instead, it stores all available cases and classifies new cases based on their similarity to existing cases.



Given a dataset with labeled points,

- classifies new data points by finding the 'K' nearest neighbors in the training data.
- **Classification** the majority class among the K neighbors is assigned to the new data point.

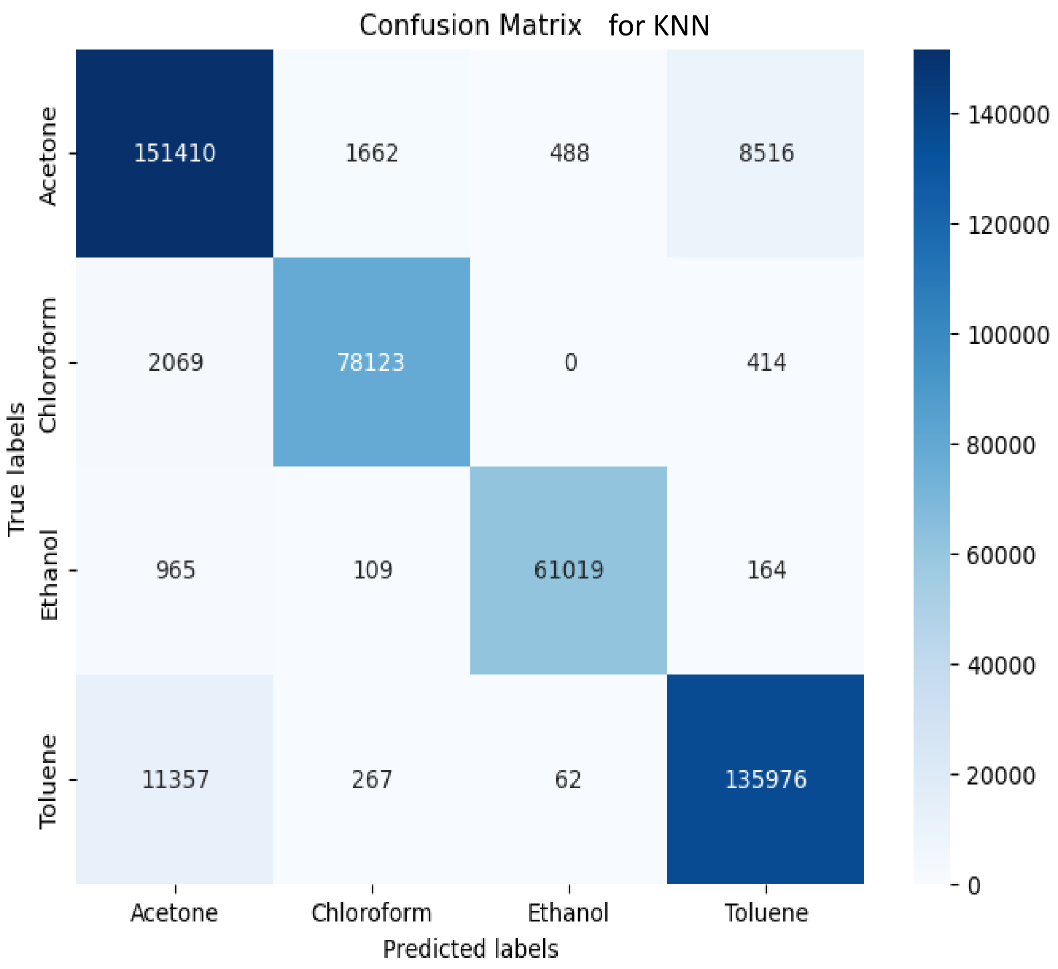
Votes

- **Regression** the output value for the new point is calculated as the average of the values of its K nearest neighbors.

Average

KNN

K	5
Train Data Points	18 lac
Test Data points	4 lac
Accuracy	0.942523
Training Time	18s
Test time	21s



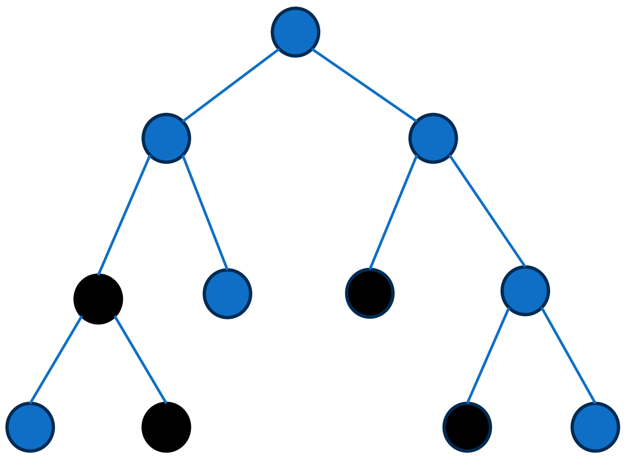
Decision Tree

Gini impurity, measures the impurity of a set of labels.

For a set **S** with **N** data points and **K** classes, the Gini impurity **G(S)** is calculated as:

$$G(S) = 1 - \sum_{i=1}^{K} P_i^2$$

where P_i is the probability of class **K** in set **S.**



Decision tree

Find the split that minimizes the impurity of the resulting subsets.

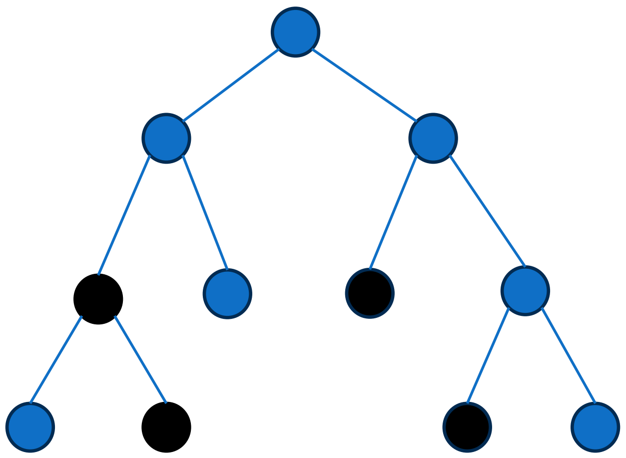
Decision Tree

Suppose, we have a dataset with two features X_1 and X_2 and four classes **[0, 1, 2, 3]**

At each node, the algorithm selects the best feature and threshold to split the data based on the Gini impurity.

For example, at the root node, it might choose X_1 with a threshold of 0.5 to split the data into two subsets.

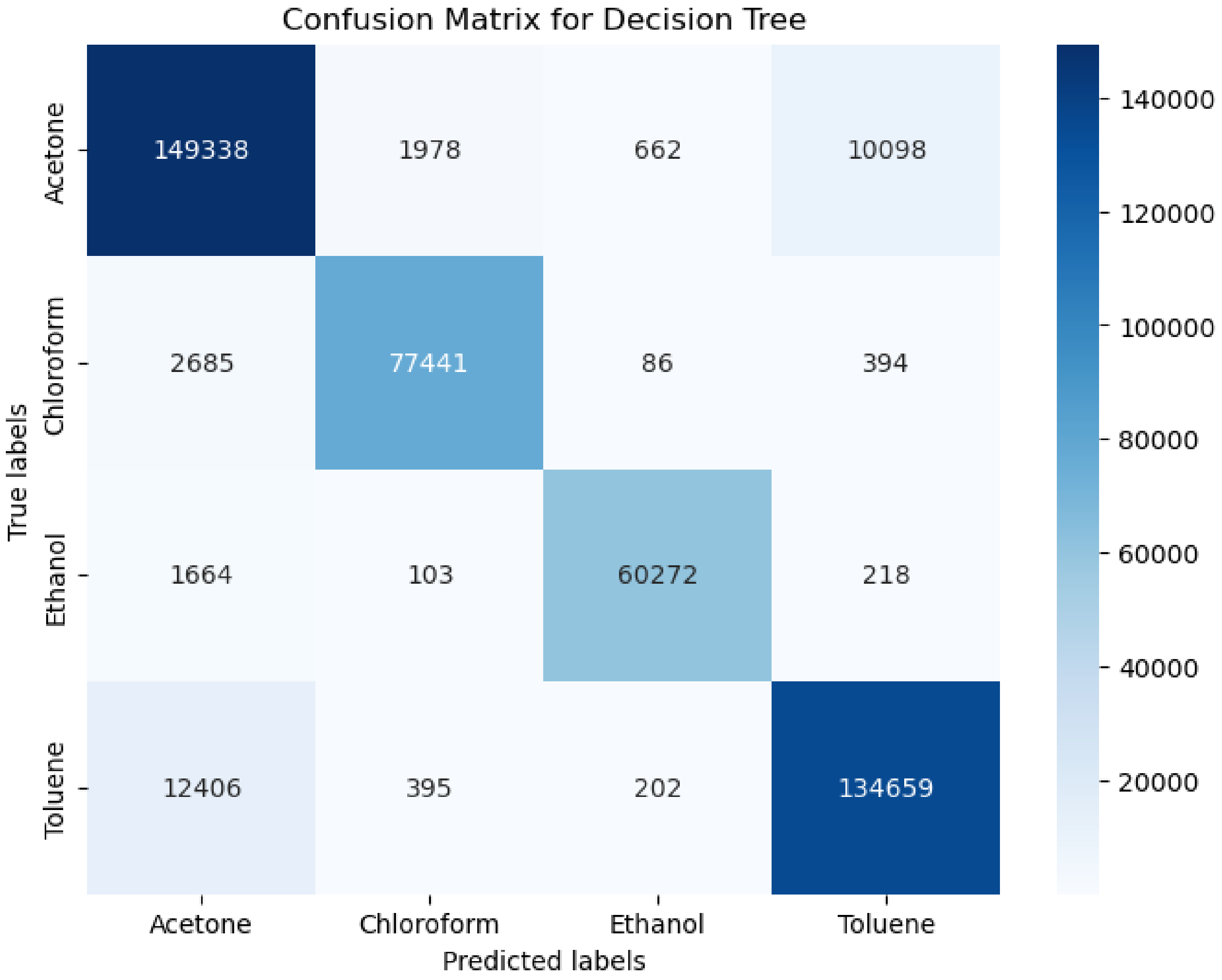
continues recursively until leaf nodes are reached, forming a tree structure.



Decision tree

Decision Tree

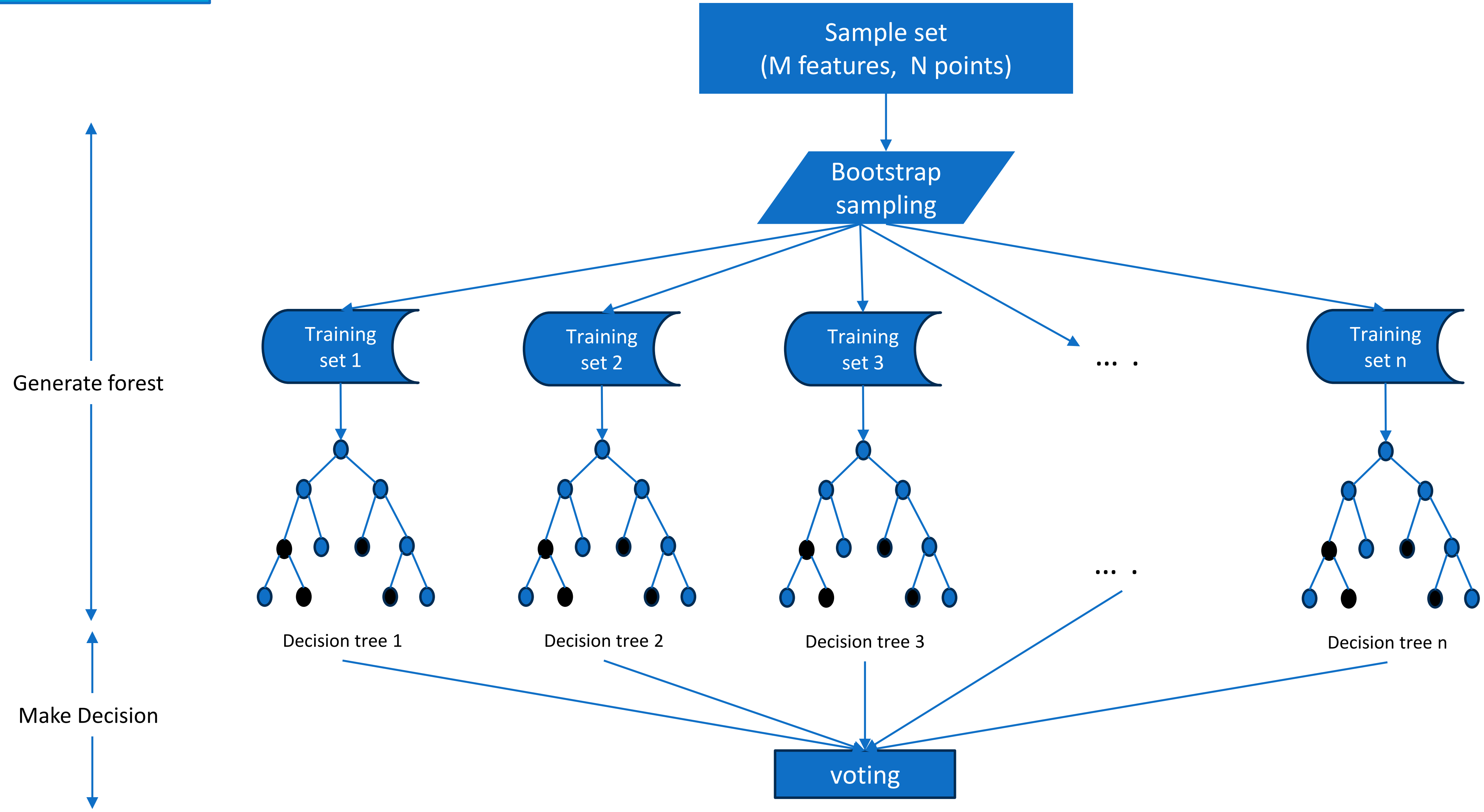
Train Data Points	18 lac
Test Data points	4 lac
Accuracy	0.9317
Training Time	3s
Test time	~0s



Random Forest

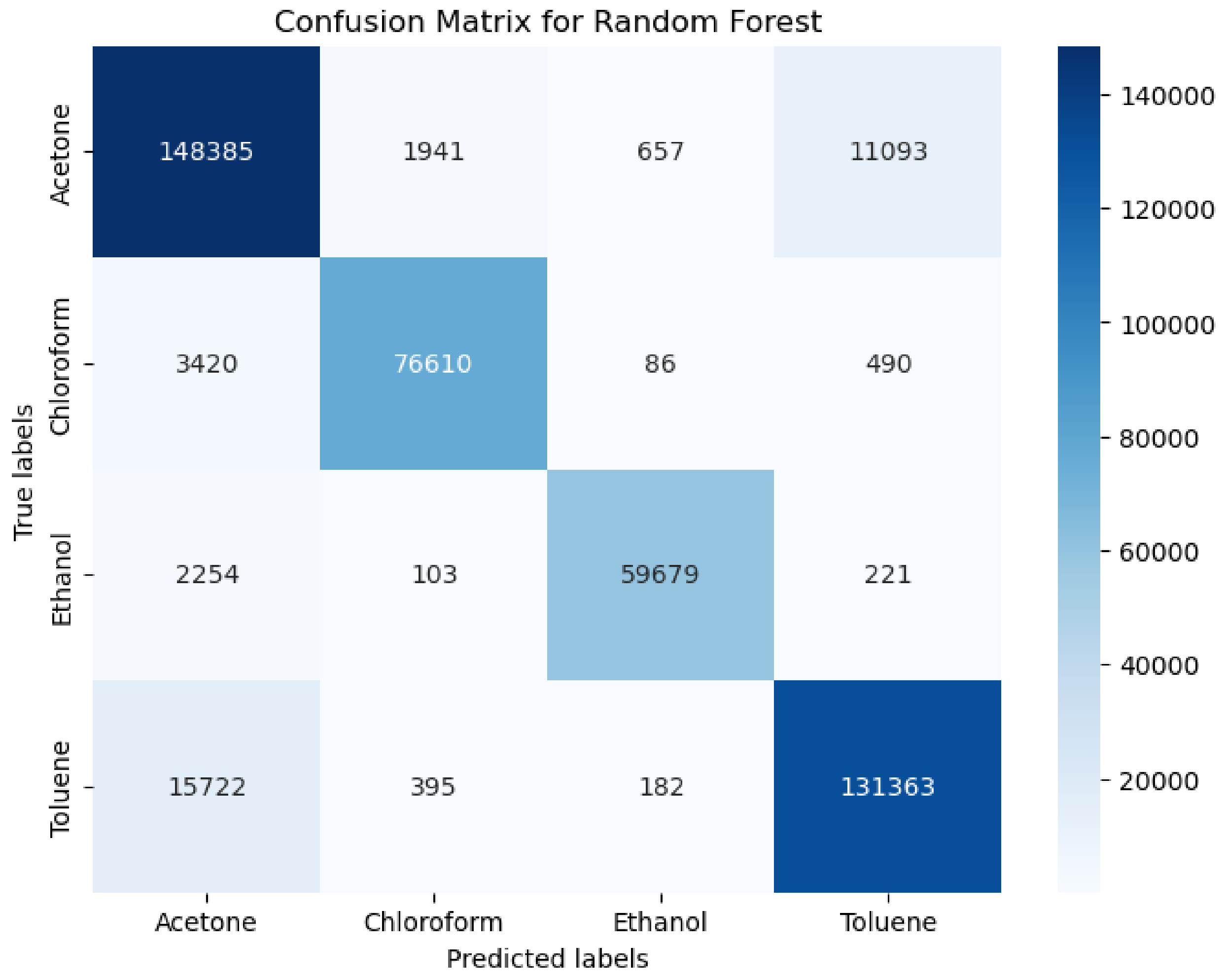
- Random forest is an ensemble classifier which uses many decision trees.
- It builds multiple decision trees and merges them together to get more accurate and stable prediction.
- It can be used for both classification and regression problems

Random Forest



Random Forest

Train Data Points	18 lac
Test Data points	4 lac
Accuracy	0.9192
Training Time	2m10s
Test time	5s



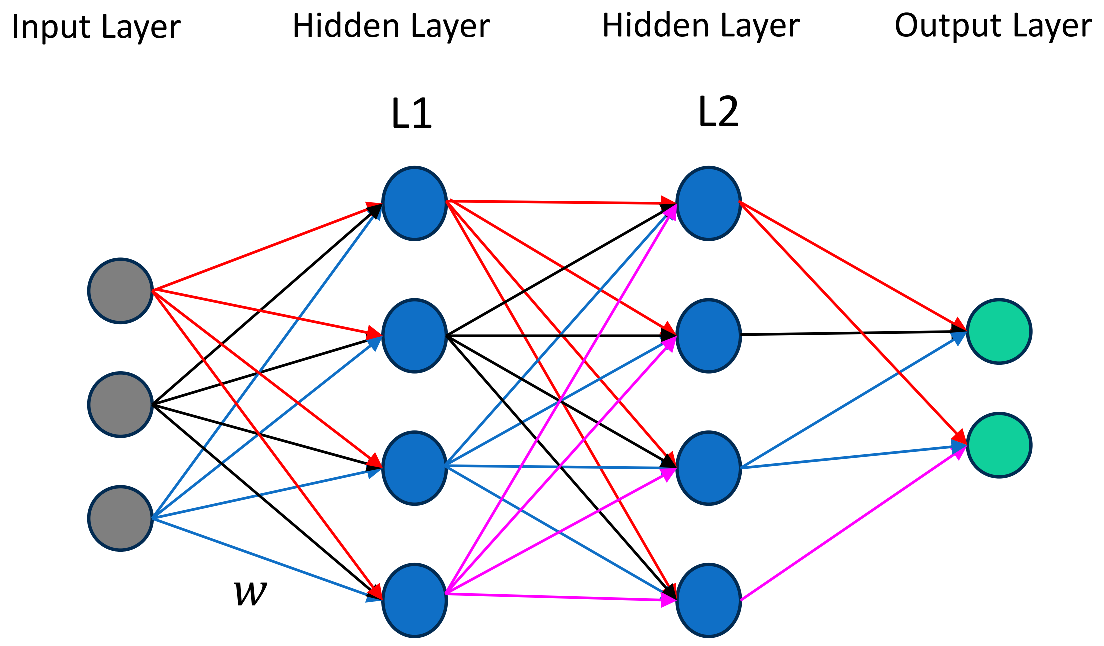
Comparison

Biological Model

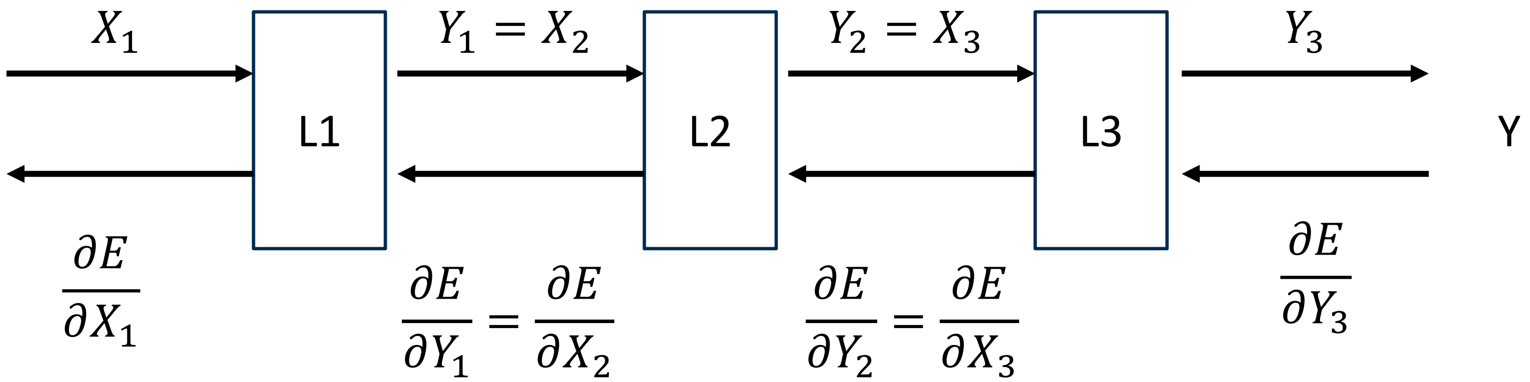
- **Neuron**: an excitable cell
- **Synapse**: connection between neurons
- **electrochemical pulse**
- **collection of neurons** along some pathway through the brain

Artificial Model

- **Neuron**: node
- **Weight**: multiplier on each edge
- **Activation Function**
- **collection of neurons** define some differentiable function


Input Layer
Hidden Layer
Hidden Layer
Output Layer
L1
L2
w

Y = network(X,W)

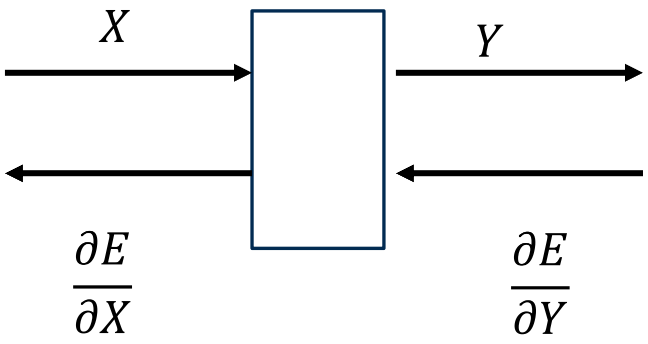


Y^* -desired output

Y –actual output

Calculate the Error (MSE),

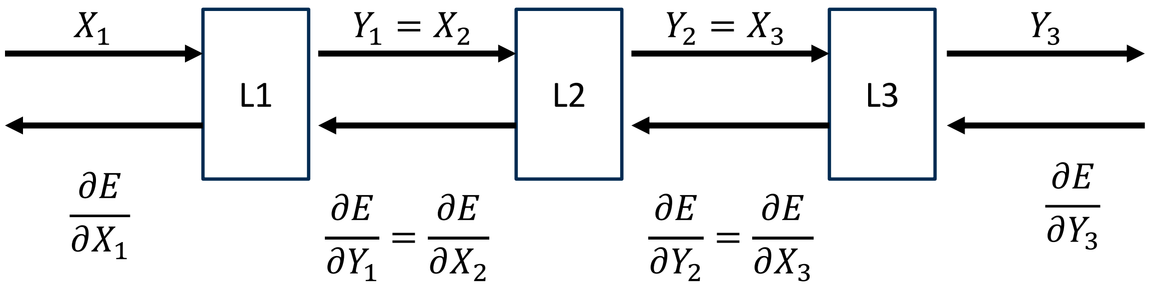
$$E = \frac{1}{2}(Y^* - Y)^2$$

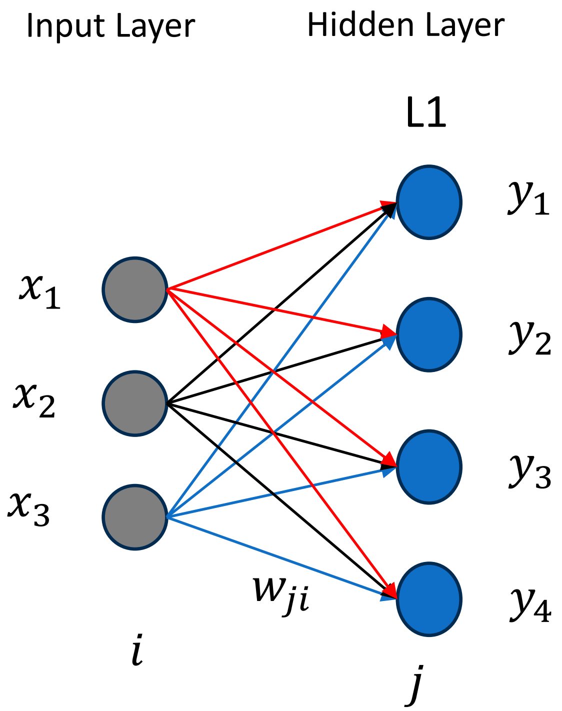


Adjust the parameters using gradient descent

$$W \rightarrow W - \alpha \frac{\partial E}{\partial W} \qquad \frac{\partial E}{\partial W} = \frac{\partial E}{\partial Y} \frac{\partial Y}{\partial W} \qquad \frac{\partial E}{\partial X} = \frac{\partial E}{\partial Y} \frac{\partial Y}{\partial X}$$

Architecture





$$y_1 = w_{11}x_1 + w_{12}x_2 + w_{13}x_3 + b_1$$

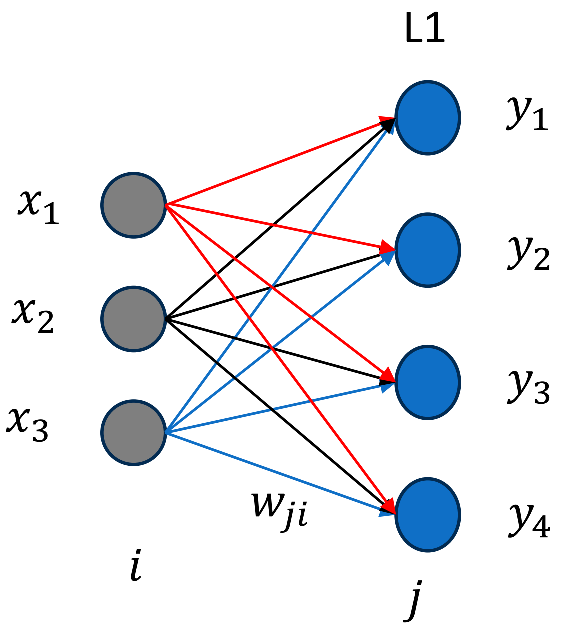
$$y_2 = w_{21}x_1 + w_{22}x_2 + w_{23}x_3 + b_2$$

$$y_3 = w_{31}x_1 + w_{32}x_2 + w_{33}x_3 + b_3$$

$$y_4 = w_{41}x_1 + w_{42}x_2 + w_{43}x_3 + b_4$$

w_{ji} are the weights

Input Layer Hidden Layer



$$y_1 = w_{11}x_1 + w_{12}x_2 + w_{13}x_3 + b_1$$

$$y_2 = w_{21}x_1 + w_{22}x_2 + w_{23}x_3 + b_2$$

$$y_3 = w_{31}x_1 + w_{32}x_2 + w_{33}x_3 + b_3$$

$$y_4 = w_{41}x_1 + w_{42}x_2 + w_{43}x_3 + b_4$$

w_{ji} are the weights

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \\ w_{41} & w_{42} & w_{43} & w_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

$$y_1 = w_{11}x_1 + w_{12}x_2 + w_{13}x_3 + b_1$$
$$y_2 = w_{21}x_1 + w_{22}x_2 + w_{23}x_3 + b_2$$
$$y_3 = w_{31}x_1 + w_{32}x_2 + w_{33}x_3 + b_3$$
$$y_4 = w_{41}x_1 + w_{42}x_2 + w_{43}x_3 + b_4$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \\ w_{41} & w_{42} & w_{43} & w_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_j \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1i} \\ w_{21} & . & . & . \\ \vdots & & \ddots & \vdots \\ w_{j1} & . & \cdots & w_{ji} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_j \end{bmatrix}$$

$$y_j = w_{j1}x_1 + w_{j2}x_2 + \ldots + w_{ji}x_i + b_j \implies \frac{\partial y_j}{\partial w_{ji}} = x_i$$

$$Y = W \cdot X + B$$

$$\implies \frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial y_j} x_i$$

$$\frac{\partial E}{\partial Y} = \begin{bmatrix} \frac{\partial E}{\partial y_1} \\ \frac{\partial E}{\partial y_1} \\ \vdots \\ \frac{\partial E}{\partial y_j} \end{bmatrix}$$ and $$\frac{\partial E}{\partial W} = \begin{bmatrix} \frac{\partial E}{\partial w_{11}} & \frac{\partial E}{\partial w_{12}} & \cdots & \frac{\partial E}{\partial w_{1i}} \\ \frac{\partial E}{\partial w_{21}} & . & . & . \\ \vdots & & \ddots & \vdots \\ \frac{\partial E}{\partial w_{j1}} & . & \cdots & \frac{\partial E}{\partial w_{ji}} \end{bmatrix}$$

$$\frac{\partial E}{\partial W} = \begin{bmatrix} \frac{\partial E}{\partial y_1}x_1 & \frac{\partial E}{\partial y_1}x_2 & \cdots & \frac{\partial E}{\partial y_1}x_i \\ \frac{\partial E}{\partial y_2}x_1 & . & . & . \\ \vdots & & \ddots & \vdots \\ \frac{\partial E}{\partial y_j} & . & \cdots & \frac{\partial E}{\partial y_j}x_i \end{bmatrix}$$

$$\frac{\partial E}{\partial W} = \begin{bmatrix} \frac{\partial E}{\partial y_1}x_1 & \frac{\partial E}{\partial y_1}x_2 & \cdots & \frac{\partial E}{\partial y_1}x_i \\ \frac{\partial E}{\partial y_2}x_1 & . & . & . \\ \vdots & & \ddots & \vdots \\ \frac{\partial E}{\partial y_j} & . & \cdots & \frac{\partial E}{\partial y_j}x_i \end{bmatrix}$$

$$\frac{\partial E}{\partial B} = \begin{bmatrix} \frac{\partial E}{\partial b_1} \\ \frac{\partial E}{\partial b_1} \\ \vdots \\ \frac{\partial E}{\partial b_j} \end{bmatrix}$$

$$y_j = w_{j1}x_1 + w_{j2}x_2 + \ldots + w_{ji}x_i + b_j$$

$$\Rightarrow \frac{\partial y_j}{\partial b_j} = 1$$

$$\Rightarrow \frac{\partial E}{\partial b_j} = \frac{\partial E}{\partial y_j}$$

$$\Rightarrow \boxed{\frac{\partial E}{\partial B} = \frac{\partial E}{\partial Y}}$$

$$\frac{\partial E}{\partial W} = \begin{bmatrix} \frac{\partial E}{\partial y_1} \\ \frac{\partial E}{\partial y_2} \\ \vdots \\ \frac{\partial E}{\partial y_j} \end{bmatrix} \cdot \begin{bmatrix} x_1 & x_2 & \cdots & x_i \end{bmatrix}$$

$$\boxed{\frac{\partial E}{\partial W} = \frac{\partial E}{\partial Y} \cdot X^T}$$

$$y_j = w_{j1}x_1 + w_{j2}x_2 + \dots + w_{ji}x_i + b_j$$

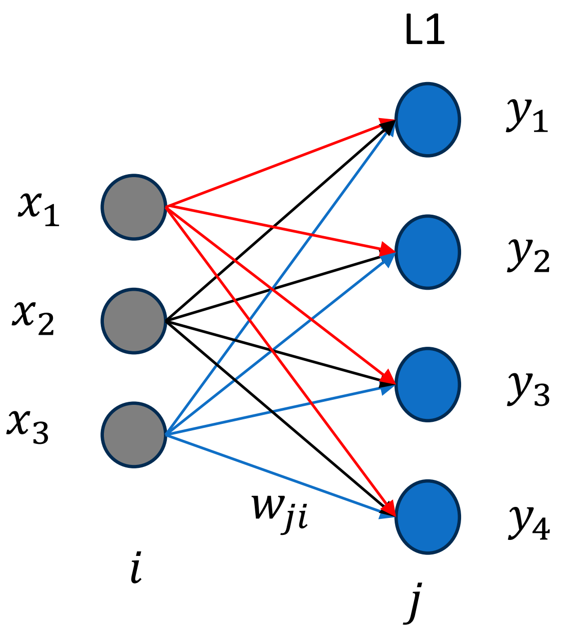
$$\frac{\partial E}{\partial x_i} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial x_i} \qquad \left(\frac{\partial y_j}{\partial x_i} = w_{ji}\right)$$

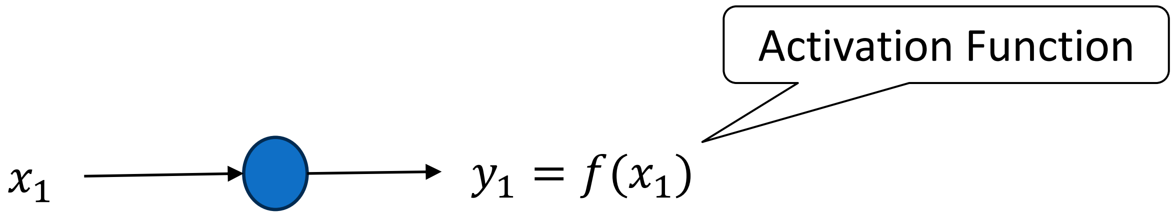
$$\Rightarrow \quad \frac{\partial E}{\partial x_i} = \frac{\partial E}{\partial y_j} w_{ji}$$

$$\frac{\partial E}{\partial X} = \begin{bmatrix} \frac{\partial E}{\partial x_1} \\ \frac{\partial E}{\partial x_2} \\ \vdots \\ \frac{\partial E}{\partial x_j} \end{bmatrix} = \begin{bmatrix} \frac{\partial E}{\partial y_1} w_{11} & + \frac{\partial E}{\partial y_1} w_{21} & + \dots & + \frac{\partial E}{\partial y_1} w_{j1} \\ \frac{\partial E}{\partial y_2} w_{12} & + \cdots & & \vdots \\ \vdots & & & \\ \frac{\partial E}{\partial y_j} w_{1j} & + \cdots & + & \frac{\partial E}{\partial y_j} w_{ji} \end{bmatrix}$$

$$\frac{\partial E}{\partial X} = \begin{bmatrix} w_{11} & w_{21} & \dots & w_{j1} \\ w_{12} & . & . & . \\ \vdots & & \ddots & \vdots \\ w_{1j} & . & \dots & w_{ji} \end{bmatrix} \begin{bmatrix} \frac{\partial E}{\partial y_1} \\ \frac{\partial E}{\partial y_1} \\ \vdots \\ \frac{\partial E}{\partial y_j} \end{bmatrix}$$

$$\boxed{\frac{\partial E}{\partial X} = W^T \cdot \frac{\partial E}{\partial Y}}$$





$$\frac{\partial E}{\partial X} = \begin{bmatrix} \frac{\partial E}{\partial x_1} \\ \frac{\partial E}{\partial x_2} \\ \vdots \\ \frac{\partial E}{\partial x_j} \end{bmatrix}$$

and

$$\frac{\partial E}{\partial x_i} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial x_i}$$

$$\frac{\partial E}{\partial x_i} = \frac{\partial E}{\partial y_j} f'(x_i)$$

$$\frac{\partial E}{\partial X} = \frac{\partial E}{\partial Y} \odot f'(X)$$

Calculate the Error (MSE), Y^* -desired output
Y –actual output

$$Y^* = \begin{bmatrix} y_1^* \\ y_2^* \\ \vdots \\ y_j^* \end{bmatrix} \qquad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_j \end{bmatrix}$$

$$\frac{\partial E}{\partial Y} = \begin{bmatrix} \frac{\partial E}{\partial y_1} \\ \frac{\partial E}{\partial y_1} \\ \vdots \\ \frac{\partial E}{\partial y_j} \end{bmatrix}$$

$$E = \frac{1}{n}\sum_{j=1}^{n}\left(y_j^* - y_j\right)^2$$

$$\frac{\partial E}{\partial y_j} = \frac{2}{n}\left(y_j^* - y_j\right)$$

$$\boxed{\frac{\partial E}{\partial Y} = \frac{2}{n}\left(Y^* - Y\right)}$$

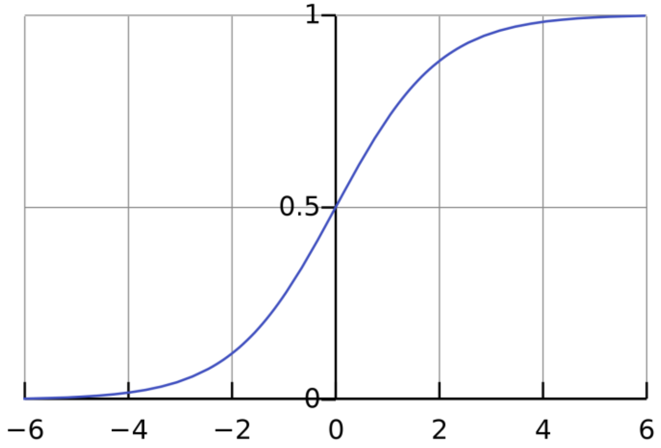
$$\boxed{\frac{\partial E}{\partial Y} = \frac{2}{n}\left(Y^* - f(X)\right)}$$

$$\frac{\partial E}{\partial X} = \frac{\partial E}{\partial Y} \odot f'(X)$$

$$\frac{\partial E}{\partial X} = \left(\frac{2}{n}\left(Y^* - f(X)\right)\right) \odot f'(X)$$

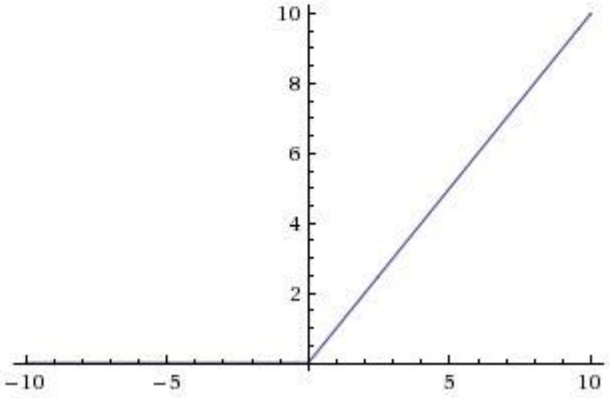
$$\frac{\partial E}{\partial W} = \left(\frac{2}{n}\left(Y^* - f(X)\right)\right) \cdot X^T$$

Sigmoid



$$S(x) = \frac{1}{1 + e^{-x}}$$

ReLU



$$f(x) = \max(0, x)$$

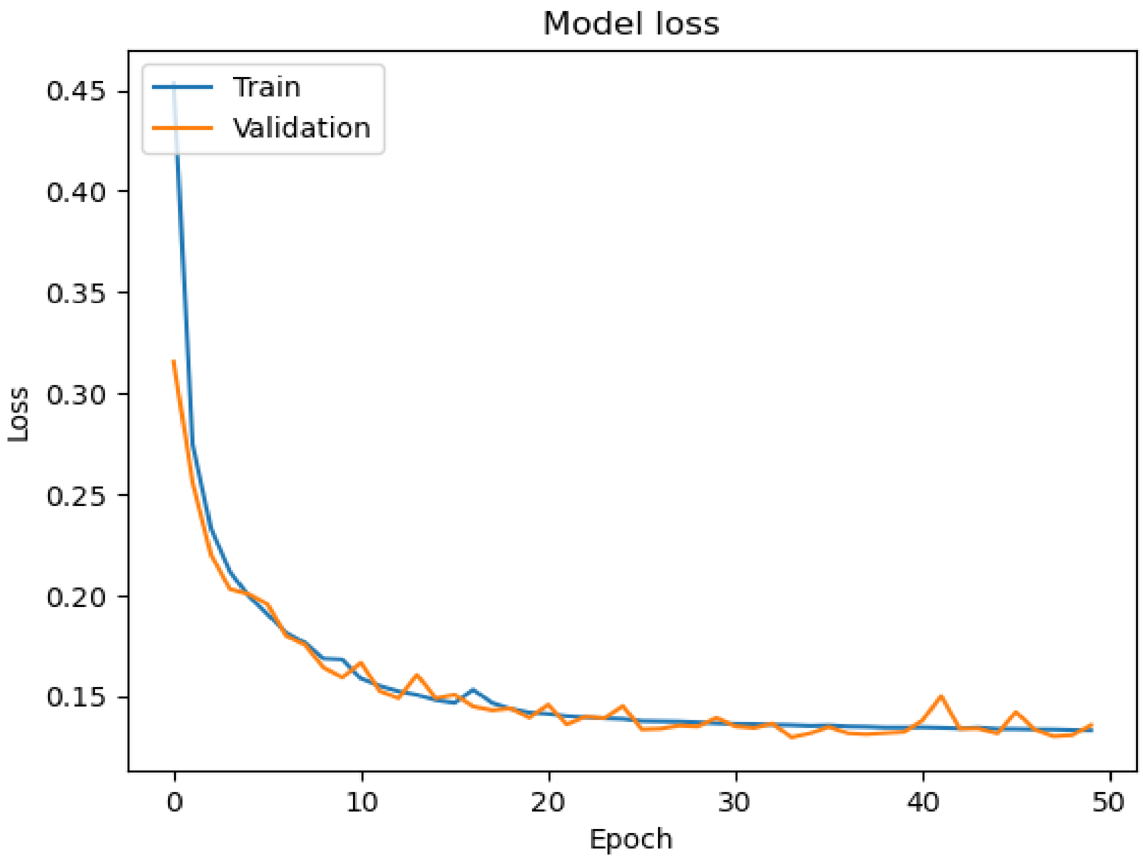
- The **batch size** is a number of samples processed before the model is updated.
- The number of **epochs** is the number of complete passes through the training dataset.

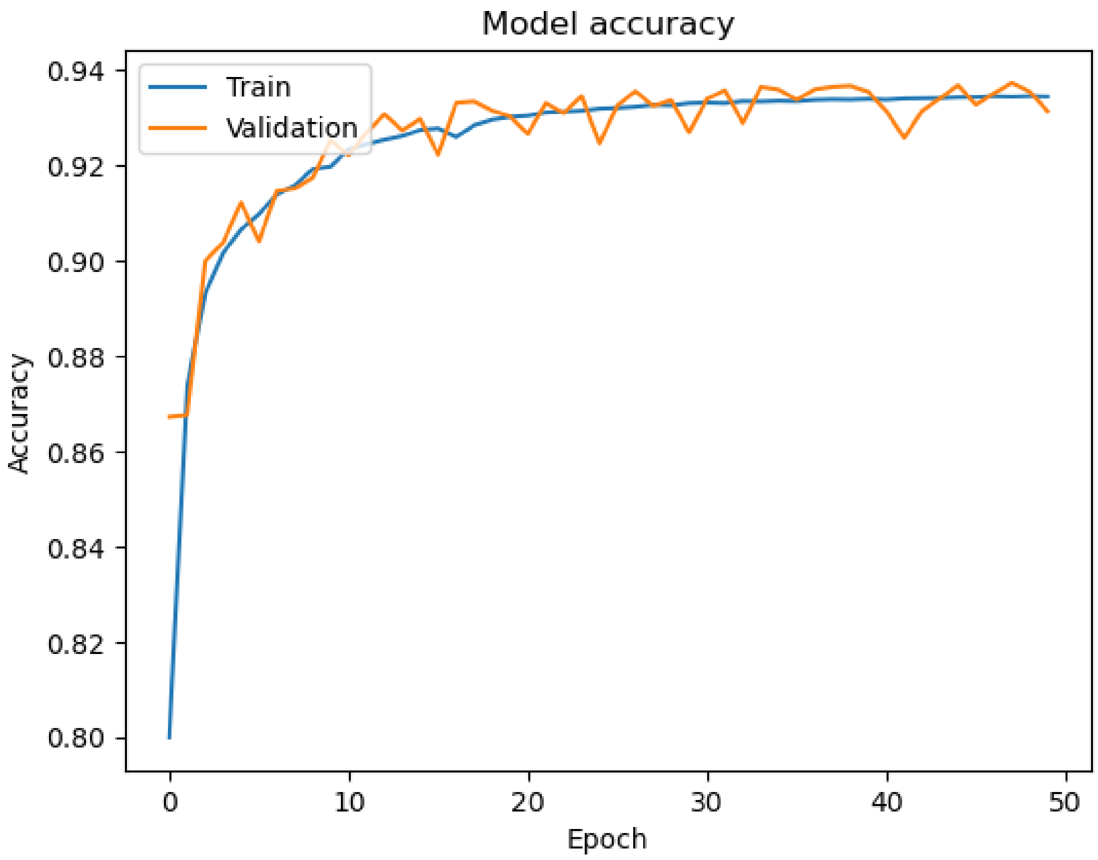
Neural Networks

Model Training

loss	0.1341156
Accuracy	0.93225824
Time	23m4s

epoch	50
Batch size	32
Validation split	0.1



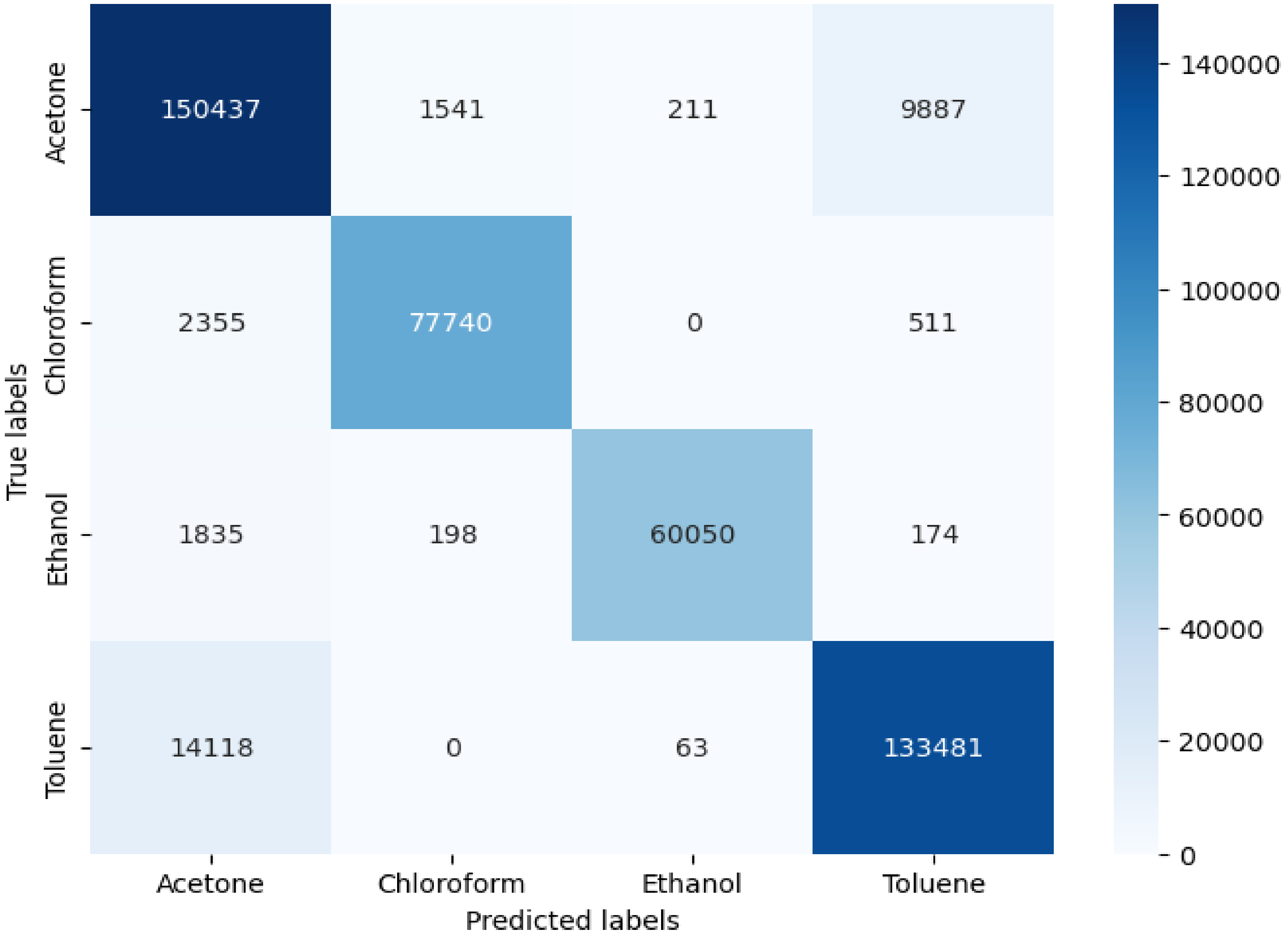


Neural Networks

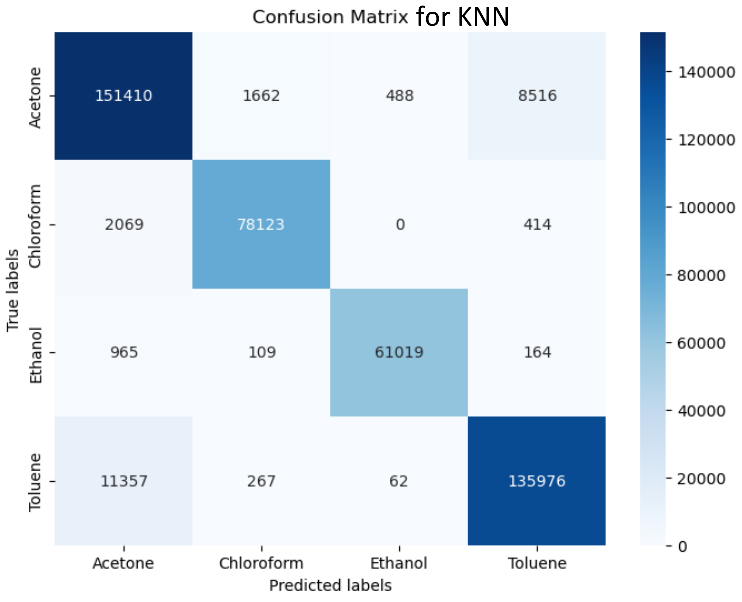
Result

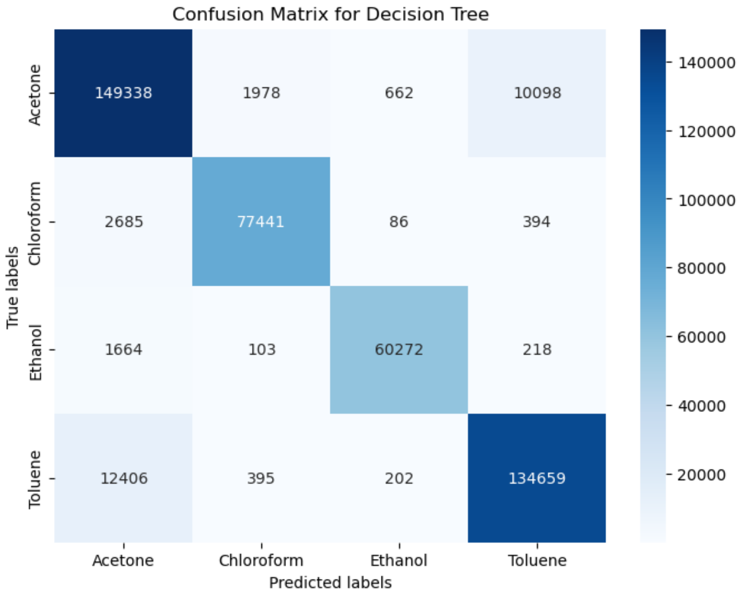
Train Data Points	18 lac
Test Data points	4 lac
Accuracy	0.91743
Training Time	23m
Test time	9s

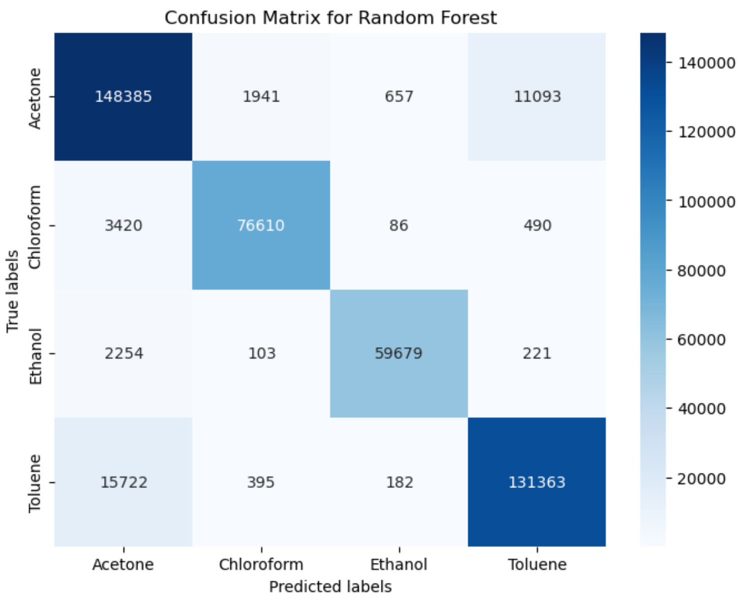
Confusion Matrix for NN

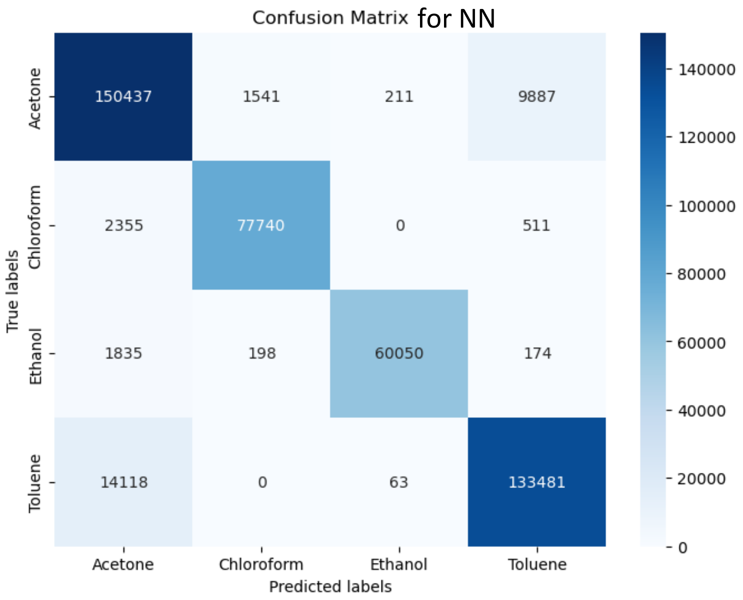


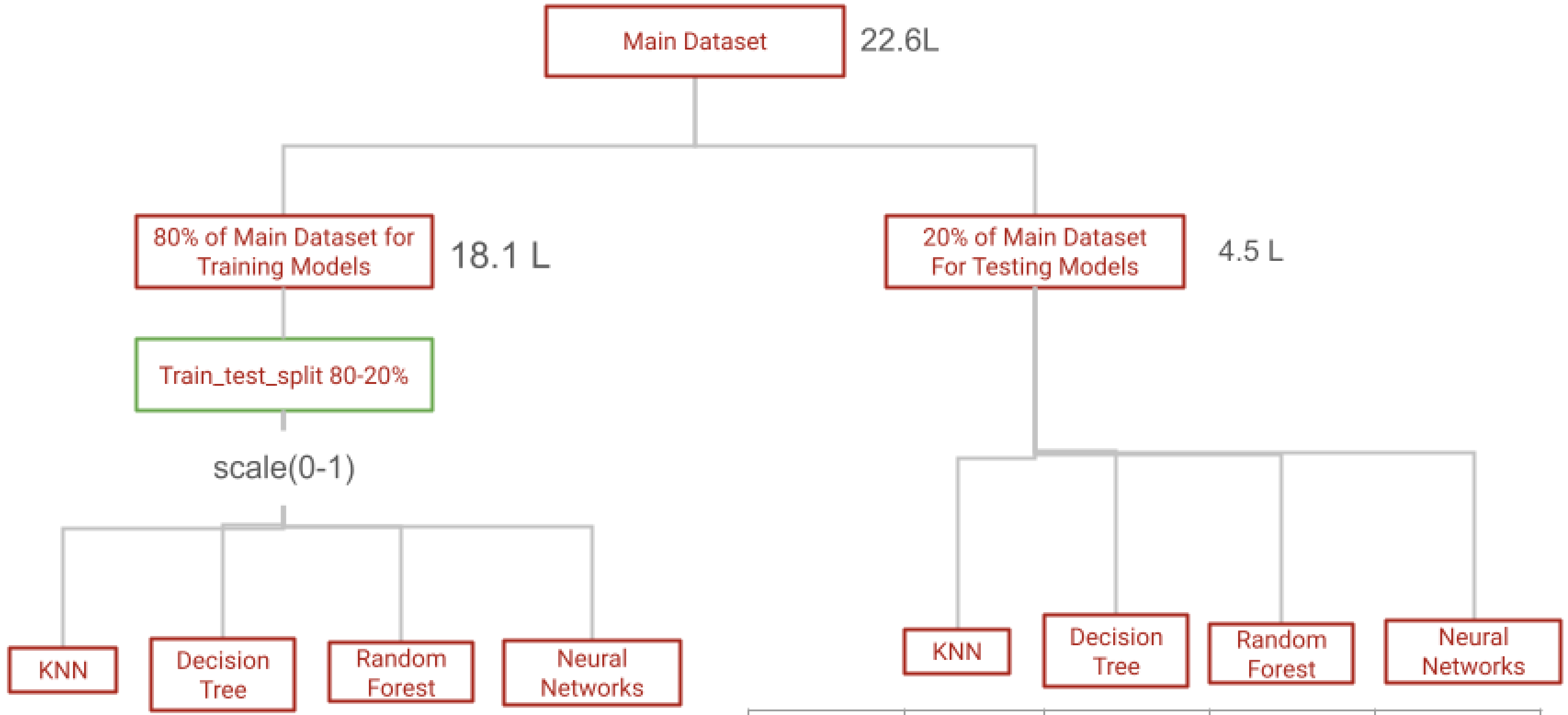
Results


Confusion Matrix for KNN
True labels
Acetone
Chloroform
Ethanol
Toluene
151410 1662 488 8516
2069 78123 0 414
965 109 61019 164
11357 267 62 135976
Predicted labels
Acetone Chloroform Ethanol Toluene
140000 120000 100000 80000 60000 40000 20000 0


Confusion Matrix for Decision Tree
True labels
Acetone
Chloroform
Ethanol
Toluene
149338 1978 662 10098
2685 77441 86 394
1664 103 60272 218
12406 395 202 134659
Predicted labels
Acetone Chloroform Ethanol Toluene
140000 120000 100000 80000 60000 40000 20000


Confusion Matrix for Random Forest
True labels
Acetone
Chloroform
Ethanol
Toluene
148385 1941 657 11093
3420 76610 86 490
2254 103 59679 221
15722 395 182 131363
Predicted labels
Acetone Chloroform Ethanol Toluene
140000 120000 100000 80000 60000 40000 20000

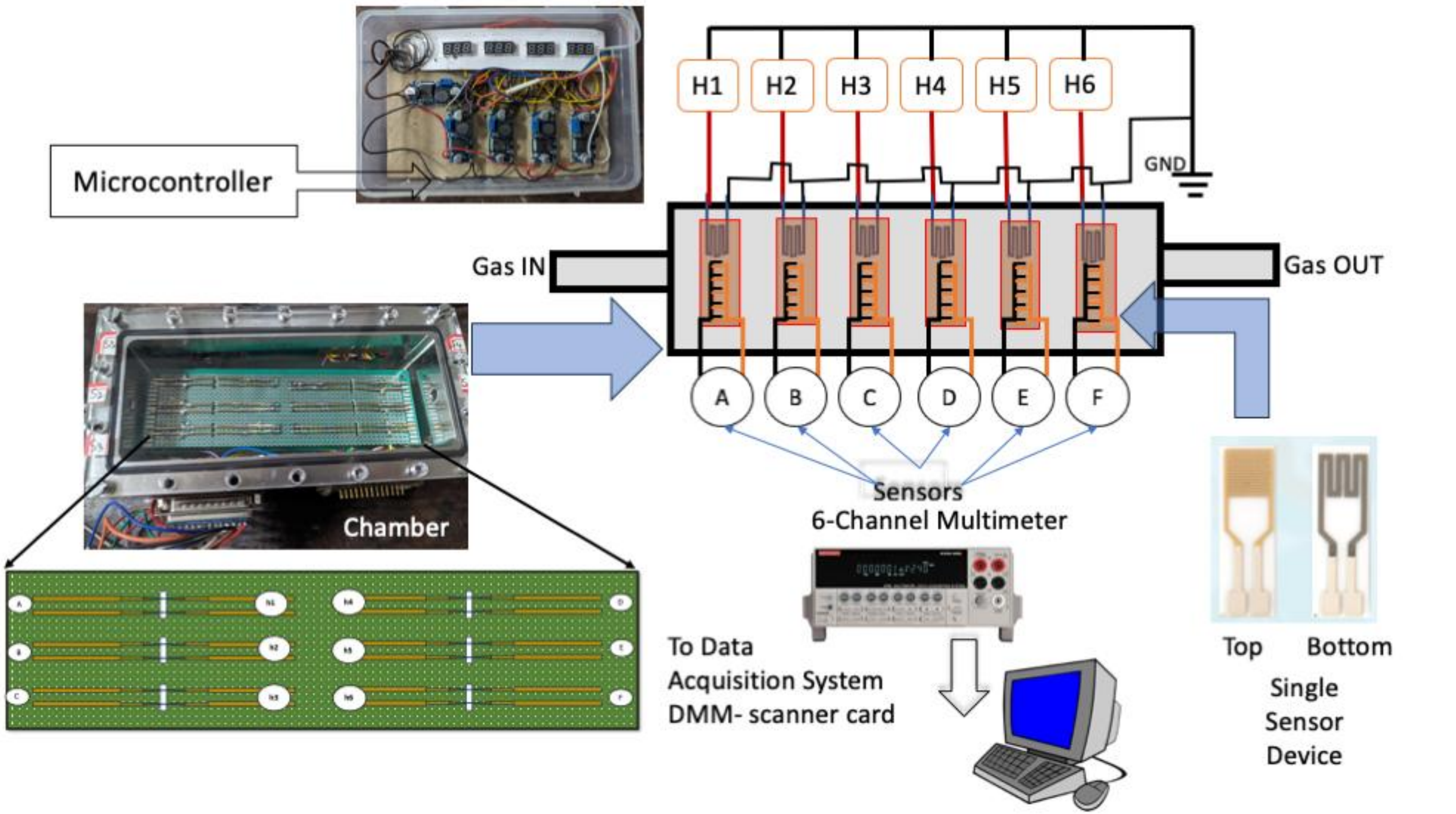

Confusion Matrix for NN
True labels
Acetone
Chloroform
Ethanol
Toluene
150437 1541 211 9887
2355 77740 0 511
1835 198 60050 174
14118 0 63 133481
Predicted labels
Acetone Chloroform Ethanol Toluene
140000 120000 100000 80000 60000 40000 20000 0



Accuracy	0.942523	0.931388	0.919031	0.932258
time	18s	3s	2m10s	23m4s

Accuracy	0.942352	0.931747	0.919213	0.91743
time	21s	0s	5s	9s

16


Microcontroller
H1
H2
H3
H4
H5
H6
GND
Gas IN
Gas OUT
A
B
C
D
E
F
Chamber
Sensors
6-Channel Multimeter
To Data
Acquisition System
DMM- scanner card
Top
Bottom
Single
Sensor
Device

Conclusion

- PCA ~ Skew Results
- KNN, Decision tree, Random forest and NN. ✓
- accuracy of above 90%. ✓
- New Six Sensor Setup. ✓