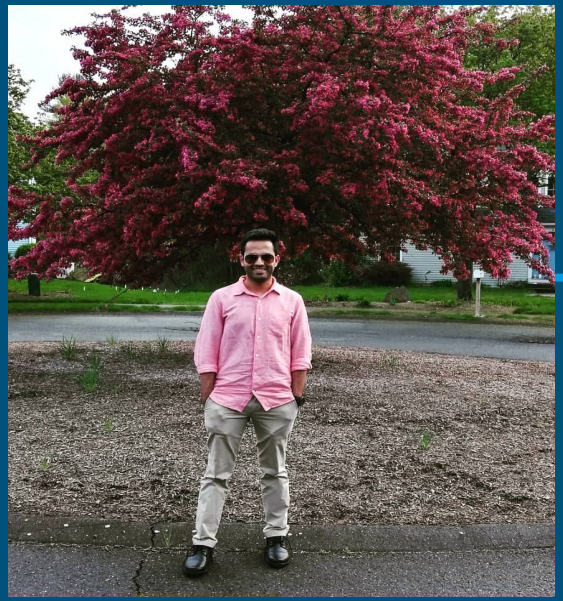
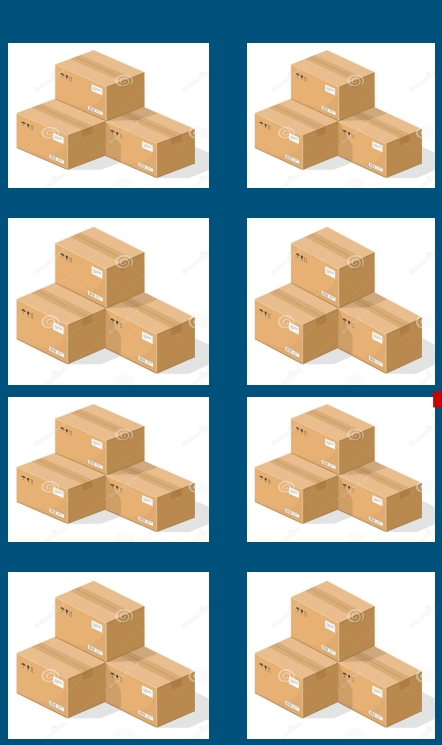
DEEP LEARNING METHODS FOR LTE TRAFFIC DATA CLASSIFICATION



Kartik Rattan
Master's Student
ECE Department, Rutgers University

Kuhu Halder (HS Intern)
John P. Stevens High School
Class of 2019

Motivation and Overview



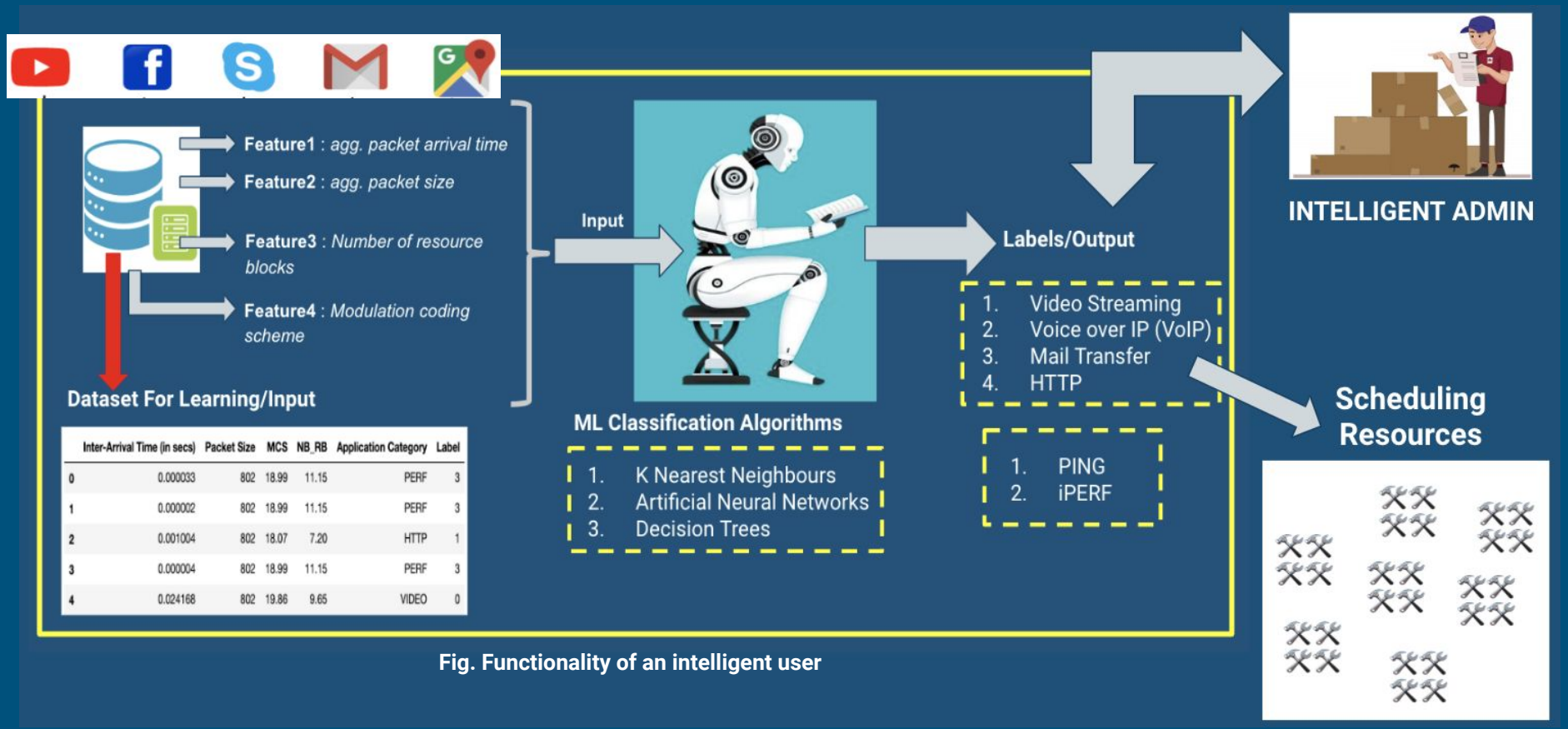
INTELLIGENT ADMIN

STEP 1: APPLICATIONS

STEP 2: CLASSIFY THE APP.

STEP 3: SCHEDULE RESOURCES

Motivation and Overview



	Inter-Arrival Time (in secs)	Packet Size	MCS	NB_RB	Application Category	Label
0	0.000033	802	18.99	11.15	PERF	3
1	0.000002	802	18.99	11.15	PERF	3
2	0.001004	802	18.07	7.20	HTTP	1
3	0.000004	802	18.99	11.15	PERF	3
4	0.024168	802	19.86	9.65	VIDEO	0

Fig. Functionality of an intelligent user

Software implementation for Applications Experimental Setup :

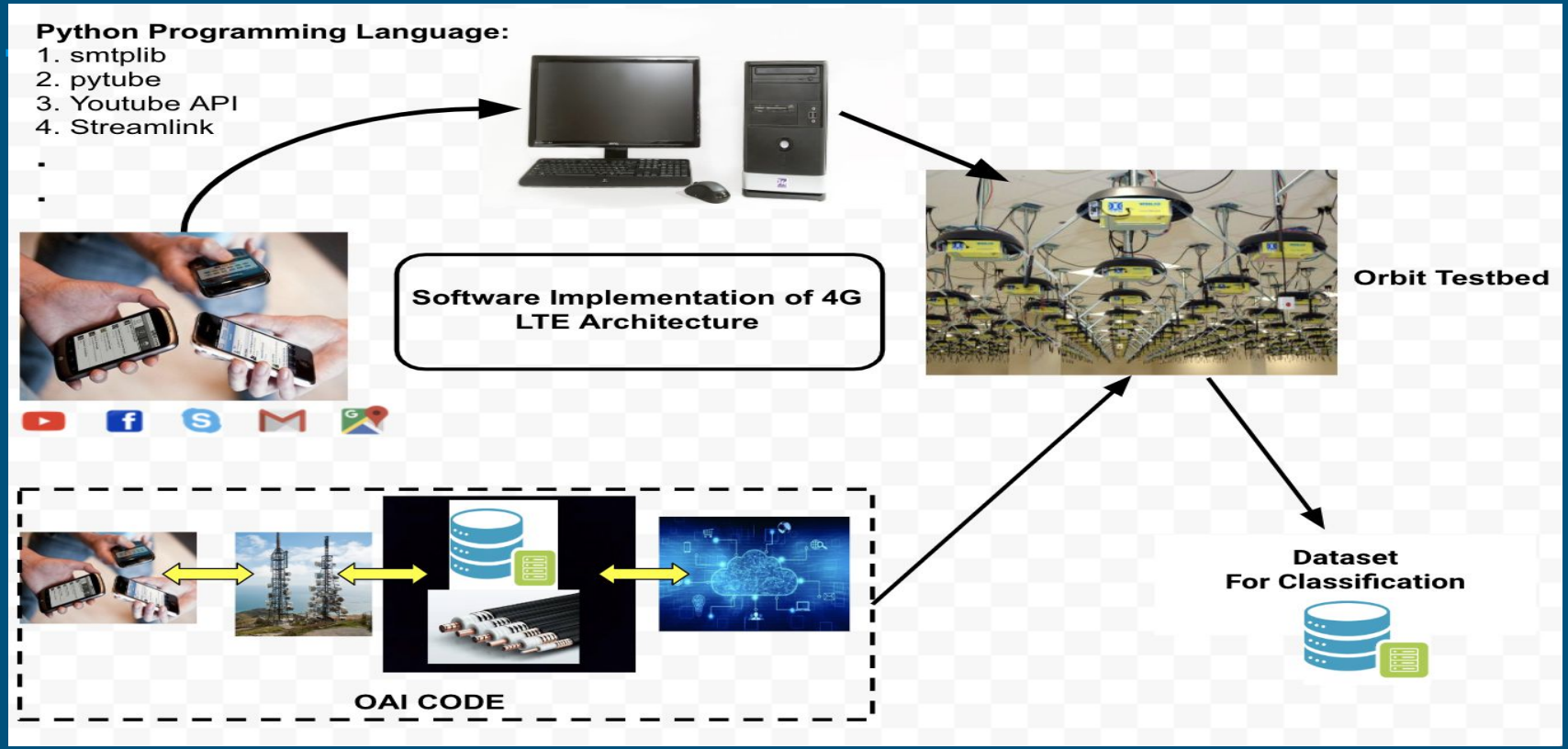
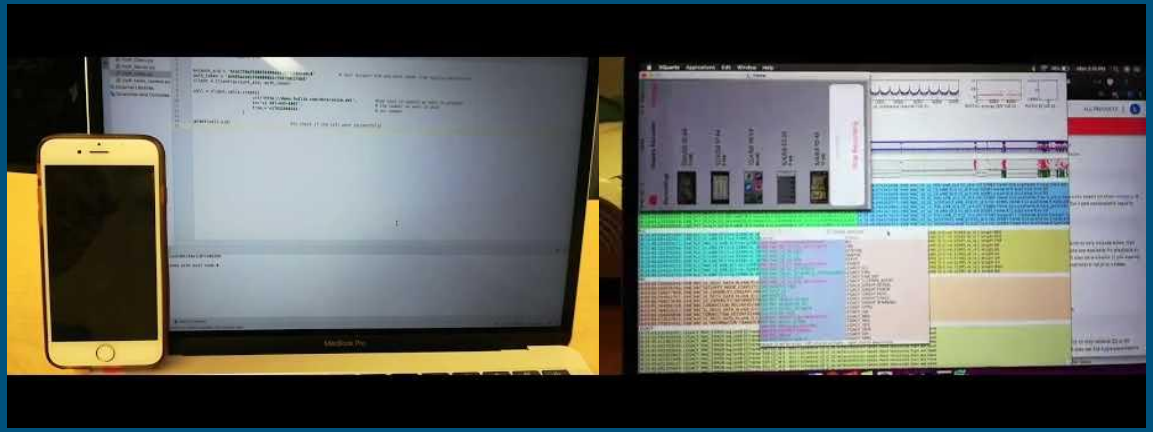
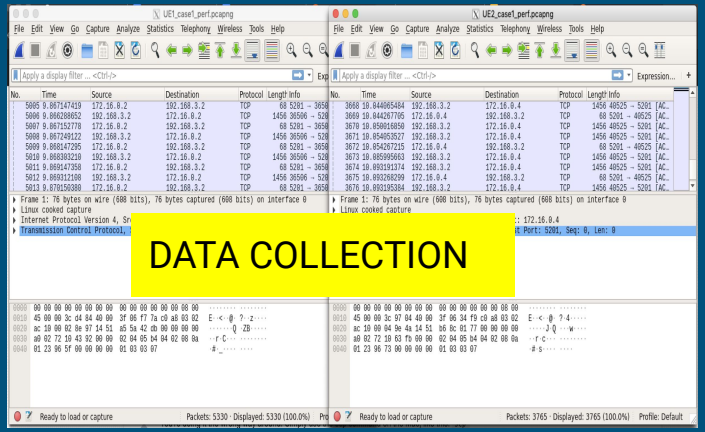
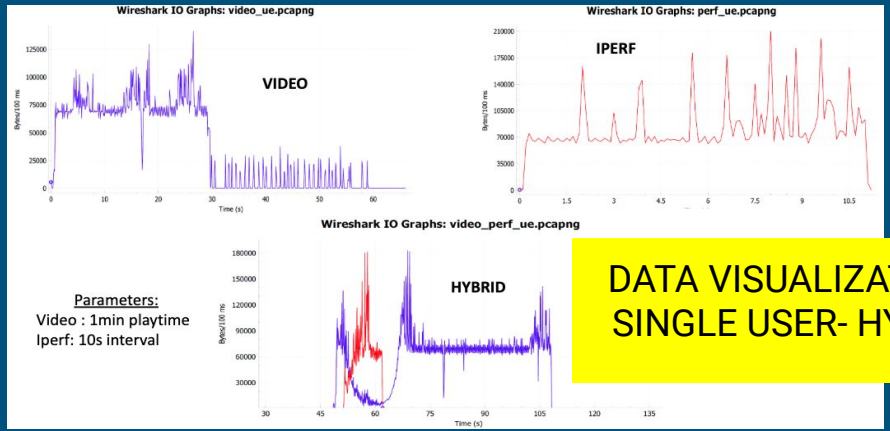


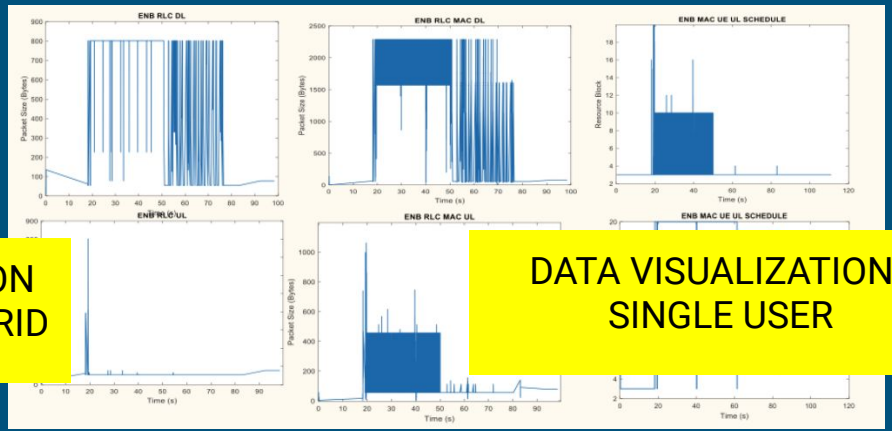
Fig. OAI is used to emulate the LTE architecture on ORBIT testbed

Data Collection Using Wireshark software

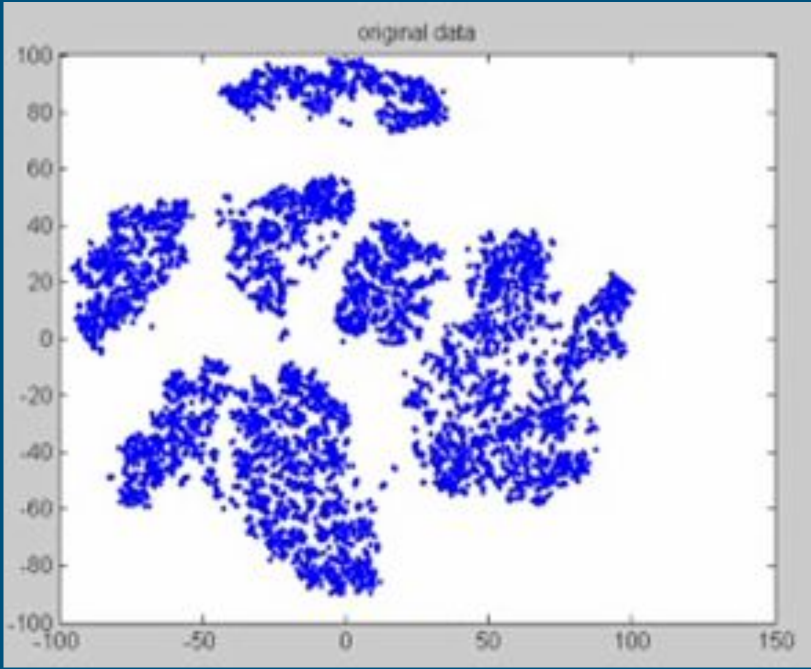


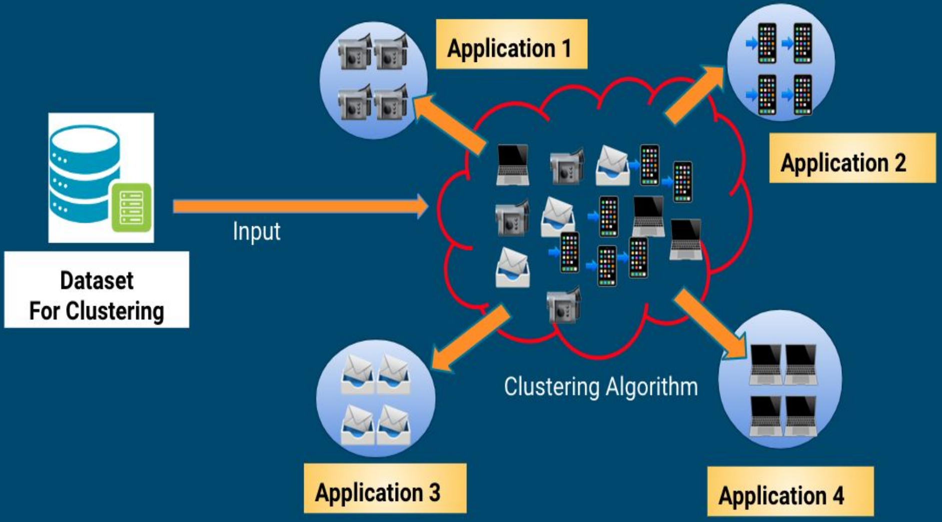






K Means Clustering






```
[1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0]

array([[1.55592935e+00, 8.66901408e+01, 1.20000000e+01, 4.00000000e+00],
       [2.05029430e-02, 2.02500000e+01, 1.20000000e+01, 4.00000000e+00]])
```

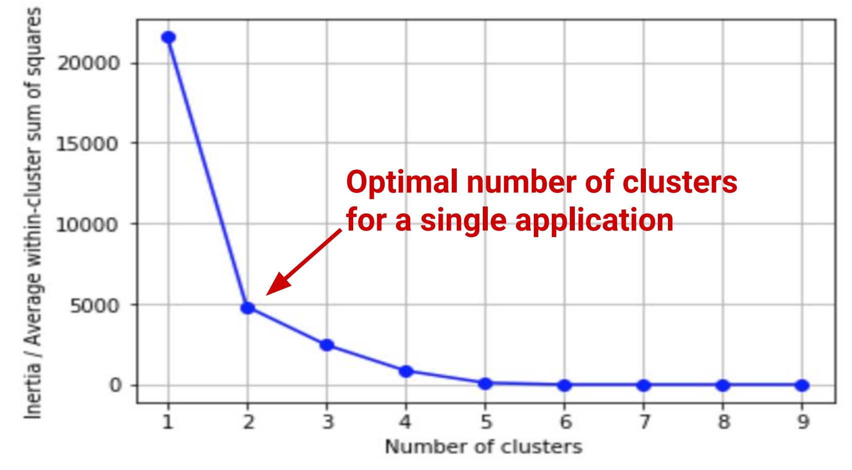


Fig : Inertia / Distortions vs Number of Clusters (Elbow Method) and clustering labels

Classification Method I- Knn Classifier

- Form of supervised learning.
- The computer learns the patterns and the classes from the data input and uses it to classify new observations.
- Predicting the class of a data point based on the class of the majority of its neighbors.

```
K-Nearest Neighbors - Test Confusion Matrix

 Predicted  HTTP  PERF  PING  VIDEO
Actual
HTTP         504     0     0      8
PERF           1  3424     0      0
PING           2     0    18      0
VIDEO          5     0     0   1644

K-Nearest Neighbors - Test accuracy: 0.997

K-Nearest Neighbors - Test Classification Report
              precision    recall  f1-score   support

        HTTP       0.98      0.98      0.98       512
        PERF       1.00      1.00      1.00      3425
        PING       1.00      0.90      0.95        20
       VIDEO       1.00      1.00      1.00      1649

    accuracy                           1.00      5606
   macro avg       0.99      0.97      0.98      5606
weighted avg       1.00      1.00      1.00      5606

Misclassified samples: 16
```

Fig: Output snippet with Python code highlighting testing accuracy

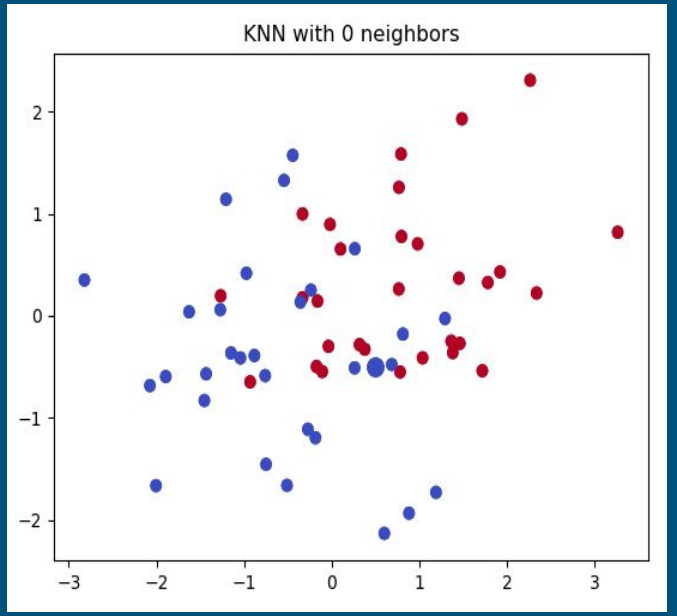


Fig: A KNN Classifier helps us classify data with respect to its neighbors

Classification Method II- Artificial Neural Network

- A set of algorithms that is designed to simulate the human brain and is used to recognize patterns.
- Our network returns a set of probabilities that a single data point belongs to each class
- The target class is the class with the highest probability

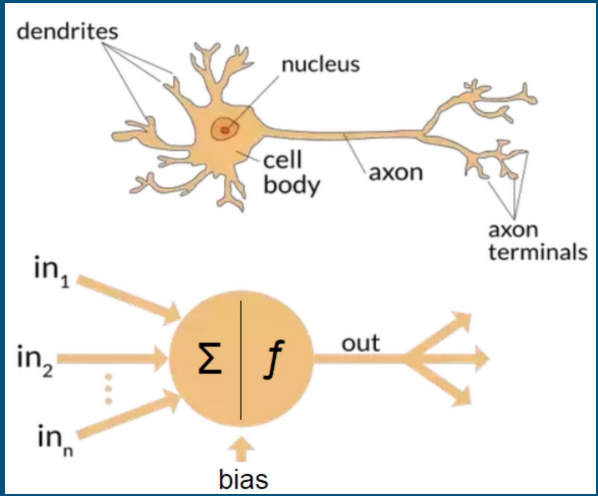


Fig: A neural network

```
Deep Neural Network  - Test accuracy: 0.98

Deep Neural Network - Test Confusion Matrix

Predicted      0     1      3
Actuall
0           1672     0     35
1             11   450     43
2              0     0     22
3              0     0   3373

Deep Neural Network  - Train accuracy: 0.976

Deep Neural Network - Train Confusion Matrix

Predicted      0     1      3
Actuall
0           3703     7    112
1             26  1025    118
2              0     0     51
3              0     0   8037
```

Fig: Output Snippet of Python code highlighting Training and testing accuracy

Classification Method III- Decision Tree

DesicionTrees

```
Dataset Length:: 18686
Dataset Shape::  (18686, 5)
Dataset::
  Application Category   Inter-Arrival Time (in secs)  ...    MCS  NB_RB
0                VIDEO                      0.008782  ...  19.86   9.65
1                 PERF                      0.000002  ...  18.99  11.15
2                VIDEO                      0.013127  ...  19.86   9.65
3                 PERF                      0.000002  ...  18.99  11.15
4                 PERF                      0.000002  ...  18.99  11.15

[5 rows x 5 columns]
['PERF' 'PERF' 'VIDEO' ... 'PERF' 'PERF' 'PERF']
Predictive Result
Accuracy is  100.0
Prediciting a sinlge value
['HTTP']

Process finished with exit code 0
```

Fig: Output Snippet of Python code for the decision tree algorithm (Single UE hybrid Data)

Future Work

- A further study is necessary to increase the number of features in our input vector.

- We get different features from different sources with varying timestamps. We need to aggregate them in sequence for correct labeling.

THANK YOU

APPENDIX - FEATURE SET

Feat 1	Feat 2	Feat 3	Feat 4	Feat 5	Feat 6	Feat 7	Feat 8	Feat 9	Feat 10
Inter-Arrival*	RLC_DL size	RLC_UL size	RLC_MAC_DL size	RLC_MAC_UL size	PDCP_DL size	PDCP_UL size	MCS*	NB_RB*	Port Num

APPENDIX :
PERFORMANCE INDICATORS FOR FEATURE EXTRACTION

ENB_MAC_UE_DL_SDU - MAC downlink SDU for an UE coming from RLC to MAC

ENB_MAC_UE_UL_SCHEDULE - MAC uplink UE scheduling decision

ENB_MAC_UE_UL_SCHEDULE_RETRANSMISSION - MAC uplink UE scheduling retransmission decision

ENB_MAC_UE_UL_PDU - MAC uplink UE received PDU

ENB_MAC_UE_UL_PDU_WITH_DATA - MAC uplink UE received PDU

ENB_MAC_UE_UL_SDU - MAC uplink UE received SDU

ENB_MAC_UE_UL_SDU_WITH_DATA - MAC uplink UE received SDU

APPENDIX :
PERFORMANCE INDICATORS FOR FEATURE EXTRACTION

ENB_MAC_UE_UL_CE - MAC uplink UE received control element

ENB_MAC_UE_DL_PDU_WITH_DATA - MAC downlink PDU for an UE

ENB_MAC_SCHEDULING_REQUEST - MAC scheduling request detected for an UE

ENB_RLC_DL - RLC downlink data

ENB_RLC_UL - RLC uplink data

ENB_RLC_MAC_DL - RLC downlink data

ENB_RLC_MAC_UL - RLC uplink data

ENB_PDCP_UL - PDCP uplink data

ENB_PDCP_DL - PDCP uplink data

APPENDIX
Clustering- K Means Clustering

Finding the optimal number of clusters

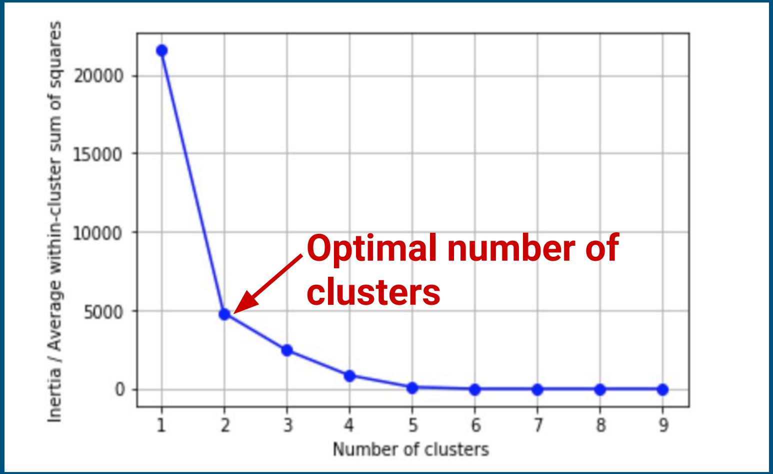


Fig : Inertia / Distortions vs Number of Clusters (Elbow Method)

Cluster Evaluation

```
[1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0]

array([[1.55592935e+00, 8.66901408e+01, 1.20000000e+01, 4.00000000e+00],
       [2.05029430e-02, 2.02500000e+01, 1.20000000e+01, 4.00000000e+00]])
```

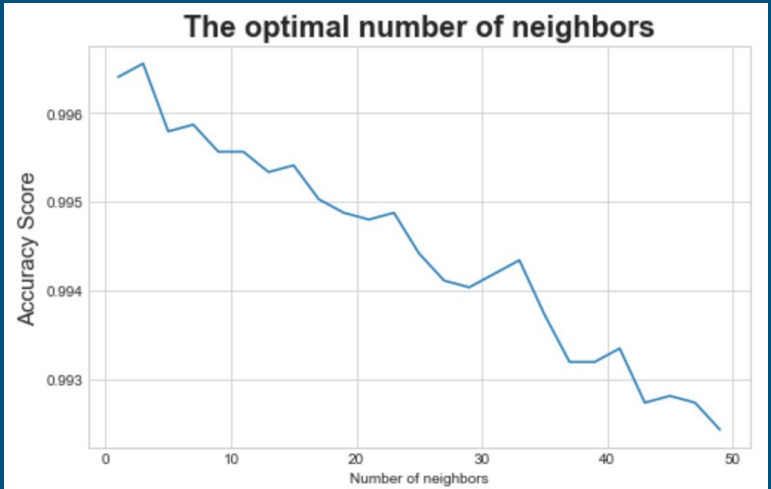
Fig: Generating labels for the data points

	Inter-Arrival Time (in secs)	Packet Size	MCS	NB_RB	Application Category	Label
0	0.000033	802	18.99	11.15	PERF	3
1	0.000002	802	18.99	11.15	PERF	3
2	0.001004	802	18.07	7.20	HTTP	1
3	0.000004	802	18.99	11.15	PERF	3
4	0.024168	802	19.86	9.65	VIDEO	0

Fig: First five rows of the dataset

APPENDIX
Classification Method I- Knn Classifier

Using GridSearchCV for Parameter Tuning




```
The optimal number of neighbors is 3.
```

Fig: Accuracy Score vs Number of Neighbors

Evaluating the Classification Model

```
K-Nearest Neighbors - Test Confusion Matrix

 Predicted  HTTP  PERF  PING  VIDEO
Actual
HTTP         504     0     0      8
PERF           1  3424     0      0
PING           2     0    18      0
VIDEO          5     0     0   1644

K-Nearest Neighbors - Test accuracy: 0.997

K-Nearest Neighbors - Test Classification Report
              precision    recall  f1-score   support

        HTTP       0.98      0.98      0.98       512
        PERF       1.00      1.00      1.00      3425
        PING       1.00      0.90      0.95        20
       VIDEO       1.00      1.00      1.00      1649

    accuracy                           1.00      5606
   macro avg       0.99      0.97      0.98      5606
weighted avg       1.00      1.00      1.00      5606

Misclassified samples: 16
```

Fig: Test Accuracy and Classification Report

APPENDIX
Classification Method II- Artificial Neural Network Classifier Model

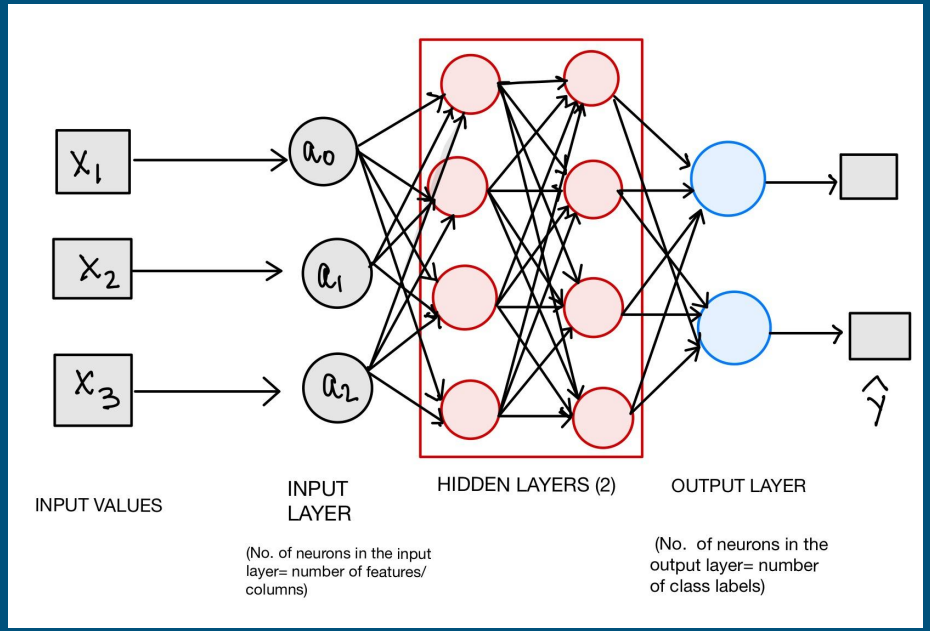


Fig: Building a multi-layer neural network

```
Deep Neural Network  - Test accuracy: 0.98

Deep Neural Network - Test Confusion Matrix

Predicted      0    1     3
Actuall
0           1672    0    35
1             11  450    43
2              0    0    22
3              0    0  3373


Deep Neural Network  - Train accuracy: 0.976

Deep Neural Network - Train Confusion Matrix

Predicted      0     1     3
Actuall
0           3703     7   112
1             26  1025   118
2              0     0    51
3              0     0  8037
```

Fig: Training and testing accuracy

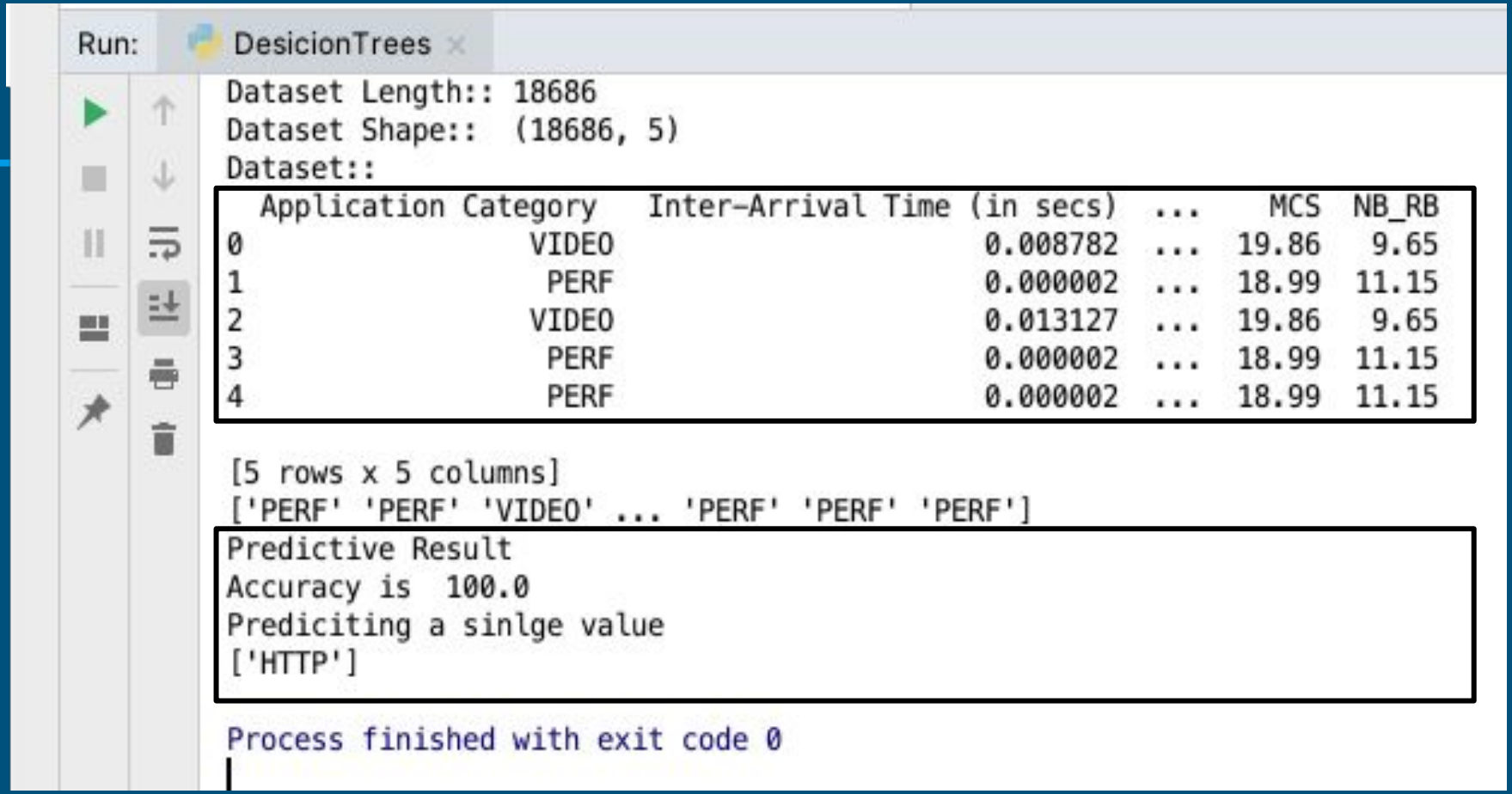
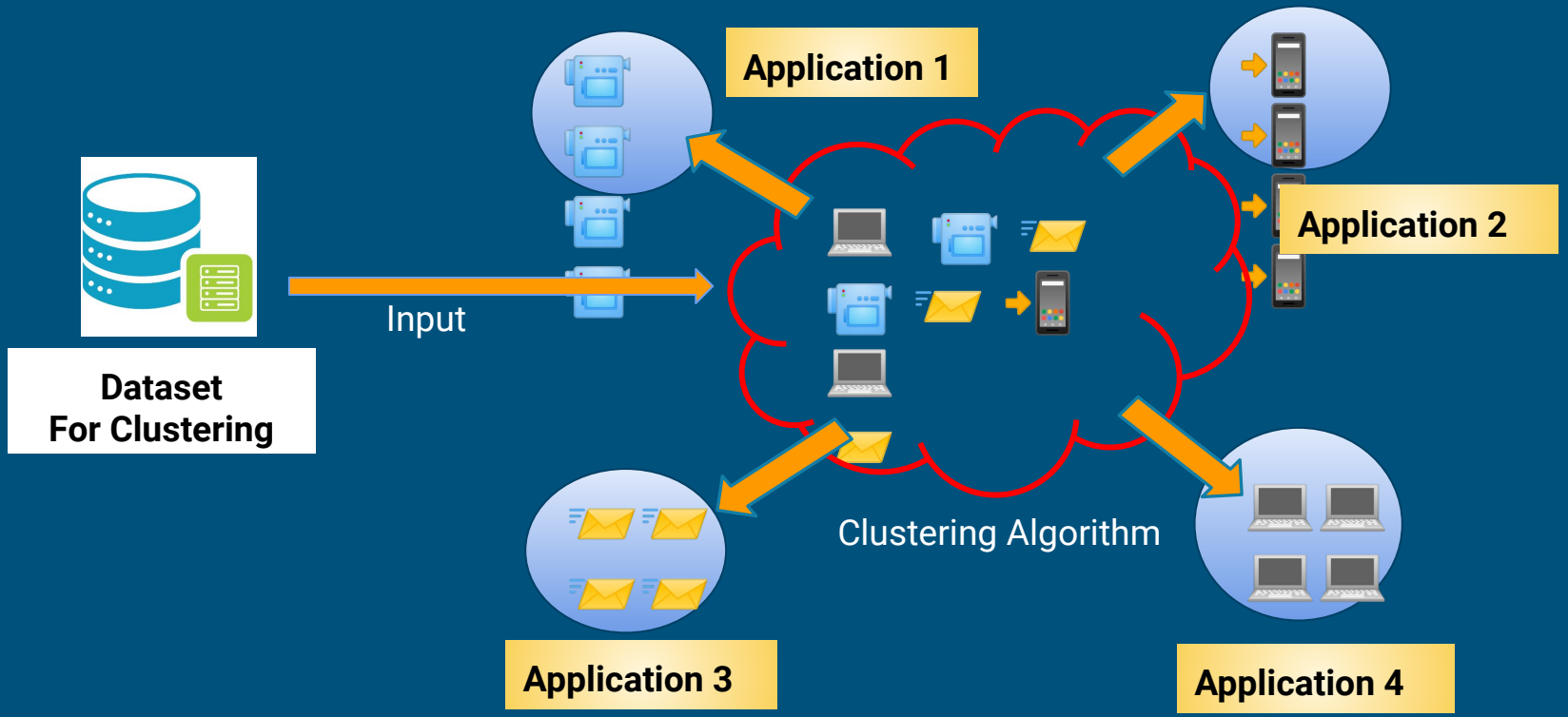


Fig: Output Snippet of Python code for the decision tree algorithm (Single UE hybrid Data)

Junk slide

Clustering - K Means Clustering



Junk slide: **Motivation and Overview**

Feature1 : *agg. packet arrival time*

Feature2 : *agg. packet size*

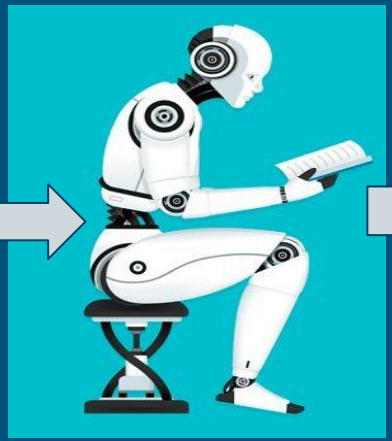
Feature3 : *Number of resource blocks*

Feature4 : *Modulation coding scheme*

Dataset For Learning/Input

	Inter-Arrival Time (in secs)	Packet Size	MCS	NB_RB	Application Category	Label
0	0.000033	802	18.99	11.15	PERF	3
1	0.000002	802	18.99	11.15	PERF	3
2	0.001004	802	18.07	7.20	HTTP	1
3	0.000004	802	18.99	11.15	PERF	3
4	0.024168	802	19.86	9.65	VIDEO	0

Input



ML Classification Algorithms

1. K Nearest Neighbours
2. Artificial Neural Networks
3. Decision Trees

Labels/Output

1. Video Streaming
2. Voice over IP (VoIP)
3. Mail Transfer
4. HTTP

1. PING
2. iPERF

Motivation and Overview

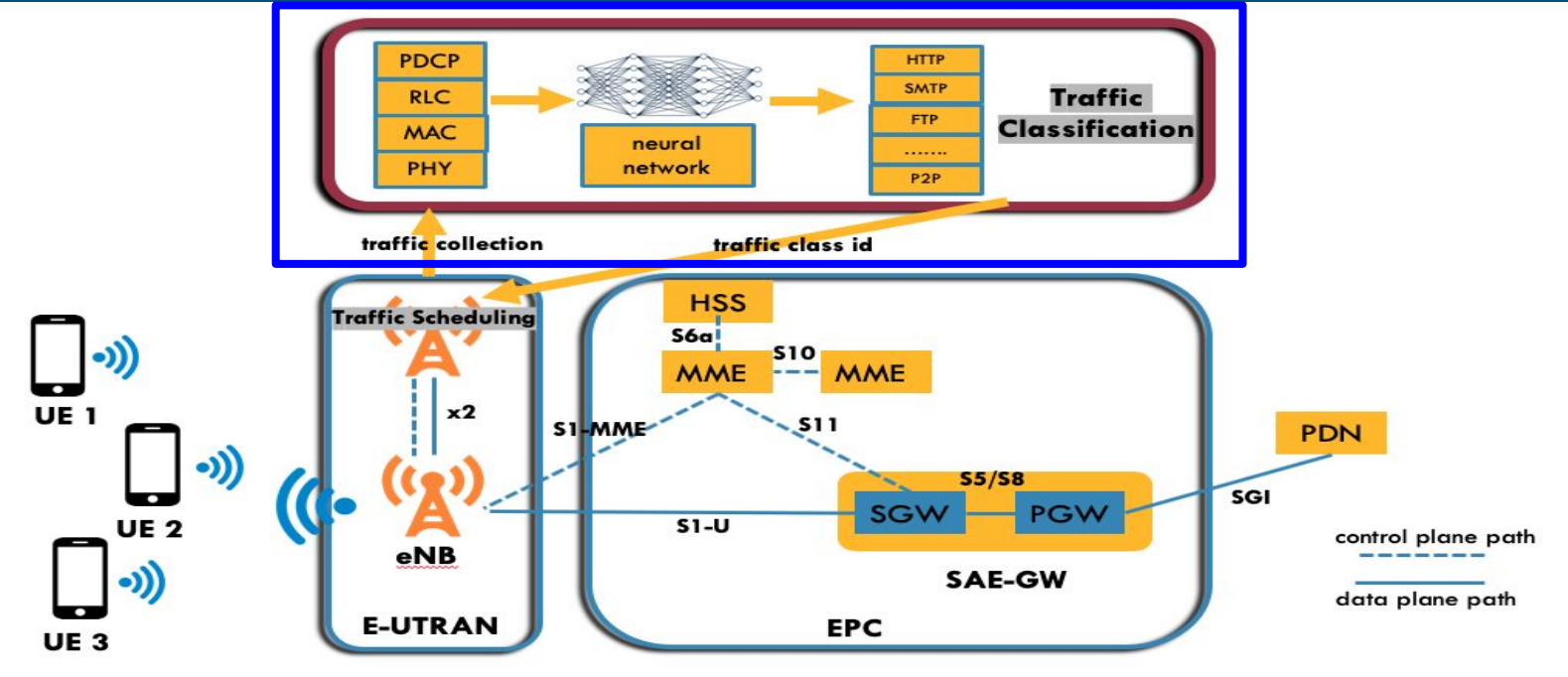


Fig. Summary for Deep Learning Methods For Traffic Data Classification

Dataset For Analysis

1. **enb_RLC_MAC_DL:**
   X(input) = agg. packet arrival time + agg. packet size + num. of RB + MCS

2. **enb_RLC_DL:**
   X(input) = packet arrival time + packet size + num. of RB + MCS

Inter-Arrival*	RLC_DL size	RLC_UL size	RLC_MAC_DL size	RLC_MAC_UL size	PDCP_DL size	PDCP_UL size	MCS*	NB_RB*	Port Num

Table: Complete feature list for classification

Clustering- K Means Clustering

Finding the optimal number of clusters

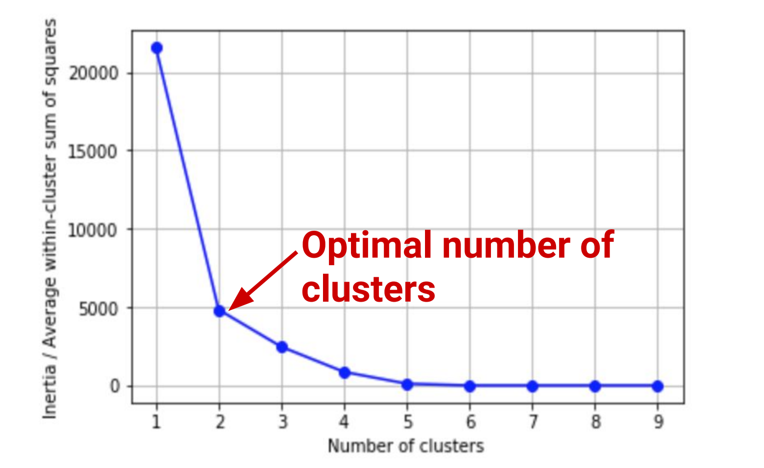


Fig : Inertia / Distortions vs Number of Clusters (Elbow Method)

Cluster Evaluation

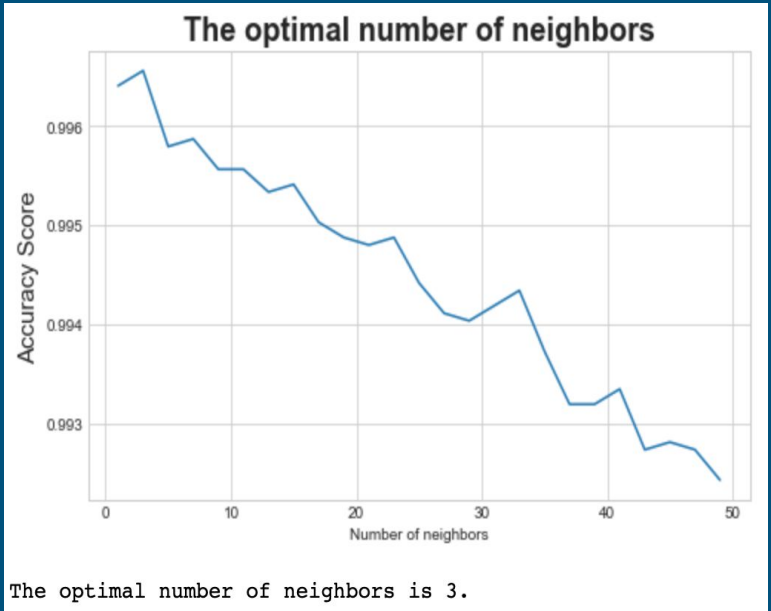
```
[1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0]

array([[1.55592935e+00, 8.66901408e+01, 1.20000000e+01, 4.00000000e+00],
       [2.05029430e-02, 2.02500000e+01, 1.20000000e+01, 4.00000000e+00]])
```

Fig: Generating labels for the data points

Classification Method I- Knn Classifier

Using GridSearchCV for Parameter Tuning




```
The optimal number of neighbors is 3.
```

Fig: Accuracy Score vs Number of Neighbors

Evaluating the Classification Model

```
K-Nearest Neighbors - Test Confusion Matrix

 Predicted  HTTP  PERF  PING  VIDEO
Actual
HTTP         504     0     0      8
PERF           1  3424     0      0
PING           2     0    18      0
VIDEO          5     0     0   1644

K-Nearest Neighbors - Test accuracy: 0.997

K-Nearest Neighbors - Test Classification Report
              precision    recall  f1-score   support

        HTTP       0.98      0.98      0.98       512
        PERF       1.00      1.00      1.00      3425
        PING       1.00      0.90      0.95        20
       VIDEO       1.00      1.00      1.00      1649

    accuracy                           1.00      5606
   macro avg       0.99      0.97      0.98      5606
weighted avg       1.00      1.00      1.00      5606

Misclassified samples: 16
```

Fig: Test Accuracy and Classification Report

Classification Method II- Artificial Neural Network Classifier Model

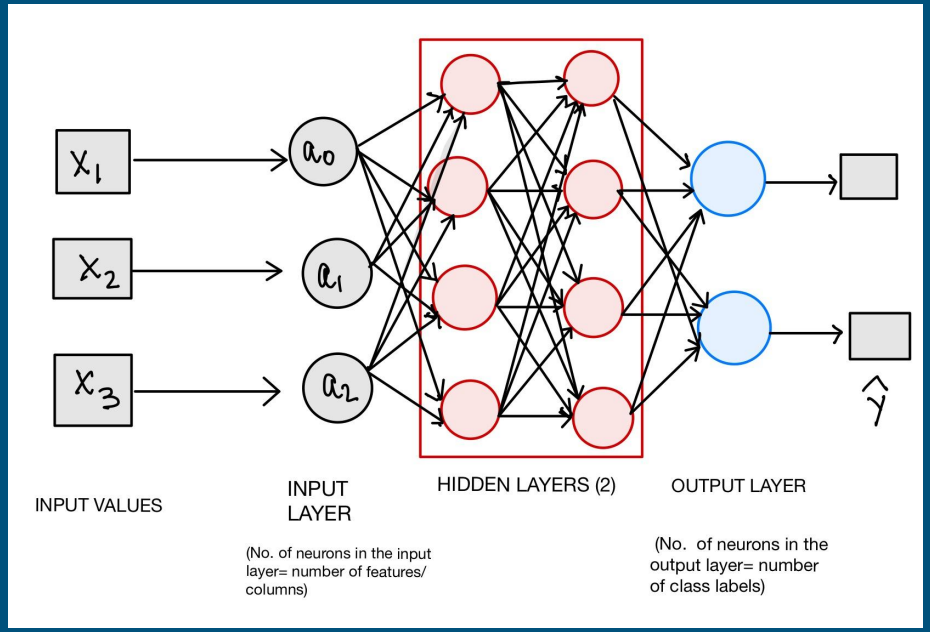


Fig: Building a multi-layer neural network

```
Deep Neural Network  - Test accuracy: 0.98

Deep Neural Network - Test Confusion Matrix

Predicted      0    1      3
Actuall
0           1672    0     35
1             11  450     43
2              0    0     22
3              0    0   3373


Deep Neural Network  - Train accuracy: 0.976

Deep Neural Network - Train Confusion Matrix

Predicted      0     1     3
Actuall
0           3703     7   112
1             26  1025   118
2              0     0    51
3              0     0  8037
```

Fig: Training and testing accuracy

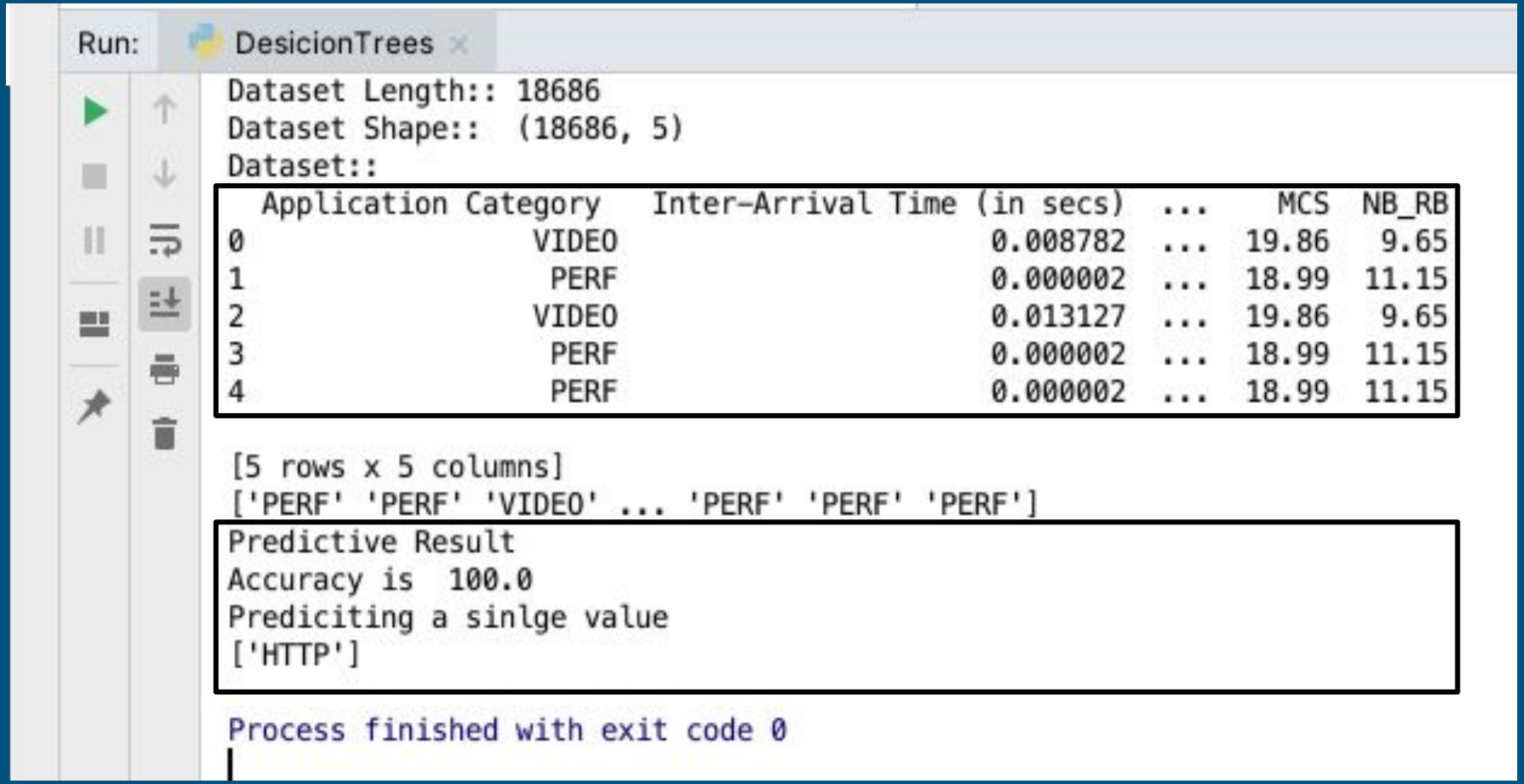
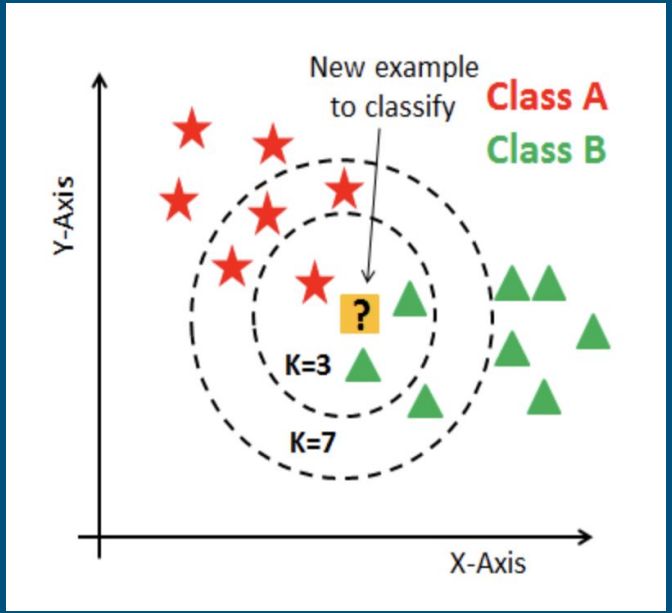


Fig: Output Snippet of Python code for the decision tree algorithm (Single UE hybrid Data)


New example
to classify
Class A
Class B
Y-Axis
?
K=3
K=7
X-Axis