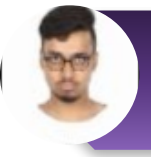
Utility-preserving anonymization for health data publishing

18BCI0247: Rohan Allen

18BCI0109: Rakshith Sachdev



18BCI0207: Shashank Gummuluru

18BCI0253: Phanider Edukulla

18BCI0110:Rishab Krishnamurthy

Under the Guidance of

Dr M Rajasekhara Babu, SCOPE

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Vellore-632014, Tamil Nadu, India

OUTLINE

- Introduction
- Importance of Application
- Idea Problem Statement
- Objectives
- Literature
  - Existing models/methods/algorithms
  - Gaps identified in existing literature
- Architecture
  - Block Diagrams
  - Algorithms
  - Flowcharts
- Requirements
  - Software's
  - Data sets

Introduction

- Protecting the privacy of medical data is extremely vital given the sensitivity of this information. It can lead to severe consequences if it falls into the wrong hands. At the same time preserving the utility of medical records is also necessary so that this information can be used in surveys, analysis, etc to improve the quality of healthcare provided. Our project attempts to delicately balance both these priorities so that neither data privacy nor utility is compromised.

Importance of application

- Introduction
  - Importance/Need of A

- Making electronic health records (EHRs) public to the masses may expose sensitive information and thus compromise the privacy and identity of an individual. Usually health records are anonymized before publishing, therefore satisfying privacy models such as *k*-anonymity. Generalization is the most commonly used anonymization algorithm which leads to immense information loss. Therefore we incorporate a utility preserving model called *h*-ceiling which restricts

Literature Review (Existing models/methods/ algorithms)

- Generalization is traditionally used which causes information loss, and thus it is not prefered.
- Existing techniques for privacy-preserving data sharing deal largely with structured data.
- Current privacy approaches for EHRs focus on detection and removal of patient identifiers from the data, which may be

Literature Review (Proposed models/methods/ algorithms)

- We propose a utility-preserving anonymization for privacy preserving data publishing (PPDP).
- To preserve data utility, the proposed method comprises three parts: (1) utility-preserving model, (2) counterfeit record insertion, (3) catalog of the counterfeit records.
- This anonymization algorithm applies full-domain generalization algorithm. They are compared to show that the utility of EHRs anonymized by the proposed method is

Literature Review (Gaps identified)

- Extreme Data loss.
- Data utility not a key priority.
- Data quality not measured
- Uselessness of EHRs for data analysts and testers.

Problem Statement

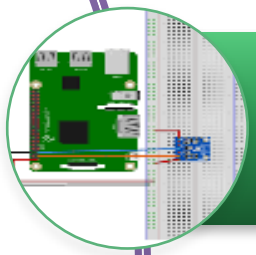
- Introduction
  - Importance/Need of App
- Problem Statement

- To develop a new utility-preserving anonymization algorithm and to show that the utility of EHRs anonymized by the proposed method is significantly better than those anonymized by previous approaches.

Objectives

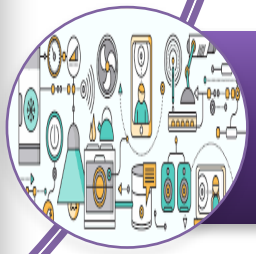
- Introduction
  - Importance/Need of App
- Problem Statement
- Obje

To anonymize and protect EHRs



To preserve utility of EHRs

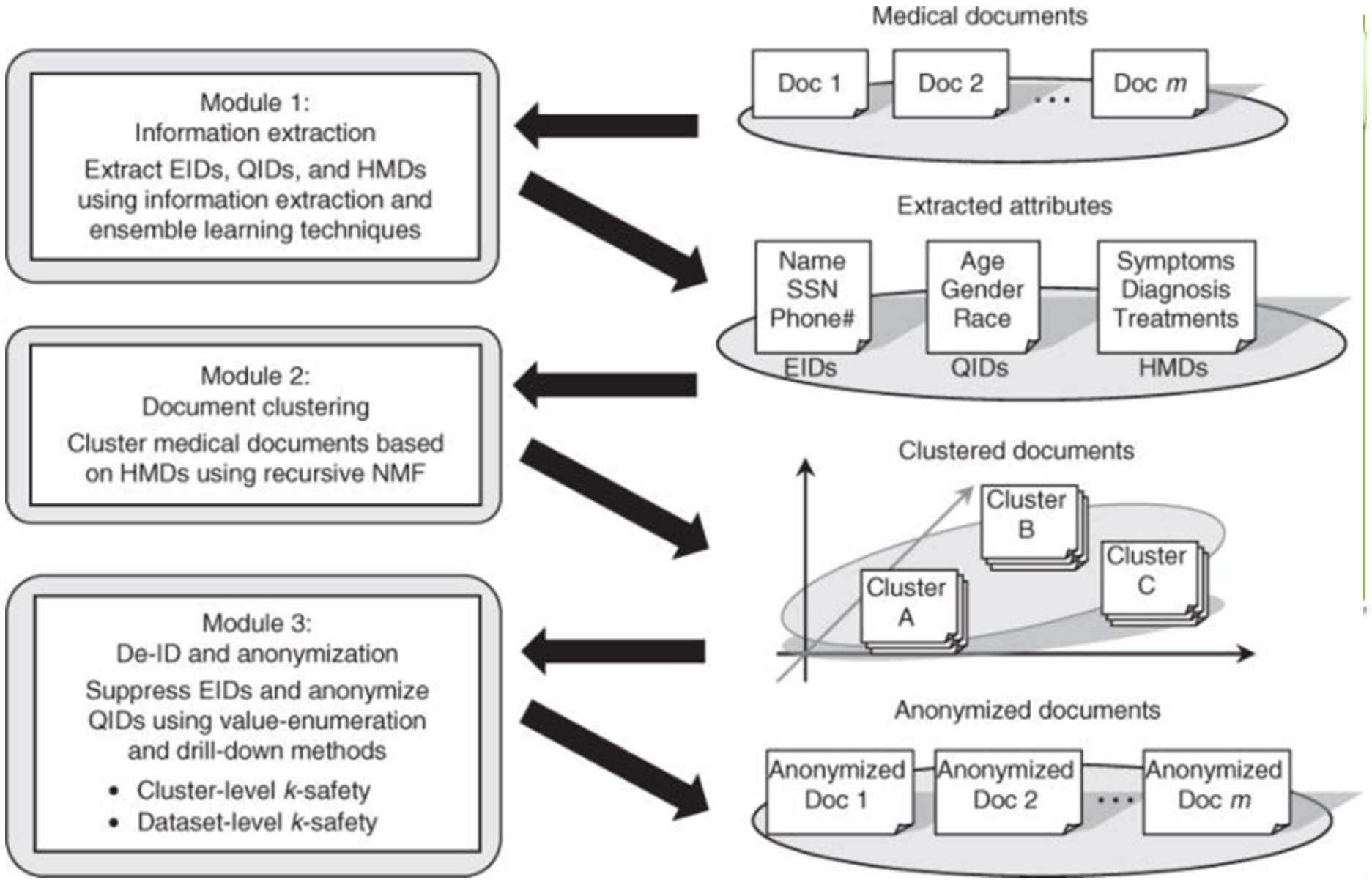
To design an algorithm to balance both privacy and utility of EHRs



To compare information loss between proposed and existing algorithm

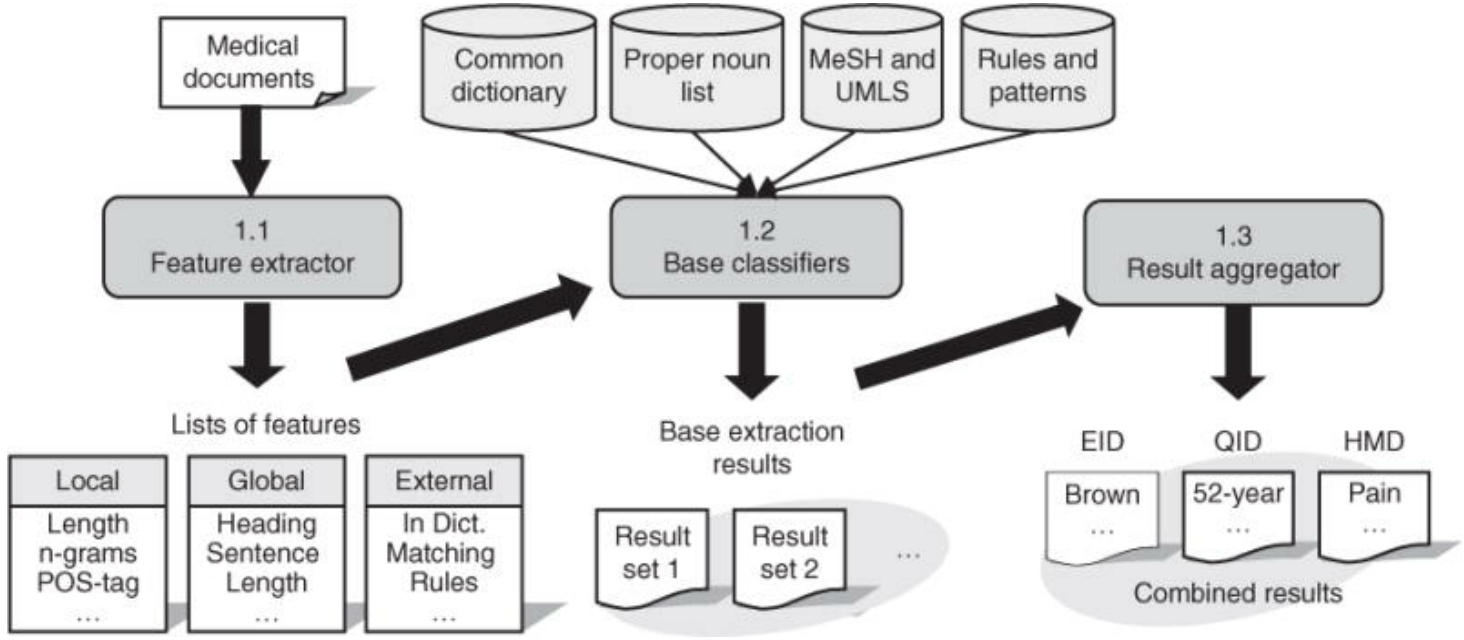
Model

- Introduction
- Importance/Need of App
- Problem Statement
- Objectives
- Design Model



Algorithms

FlowChart


Medical documents
Common dictionary
Proper noun list
MeSH and UMLS
Rules and patterns
1.1 Feature extractor
1.2 Base classifiers
1.3 Result aggregator
Lists of features
Local
Length
n-grams
POS-tag
...
Global
Heading
Sentence Length
...
External
In Dict.
Matching Rules
...
Base extraction results
Result set 1
Result set 2
...
EID
QID
HMD
Brown
...
52-year
...
Pain
...
Combined results

List of Software's

- Introduction
- Importance/Need of App
- Problem Statement
- Objectives
- Design Model
- Requirements
  - Hardware
  - Software

- Jupyter Notebooks
  - Description: to store and run ipynb files

- Python libraries
  - Various libraries used like pandas, numpy, sklearn etc.

Dataset details

- Introduction
- Importance/Need of App
- Problem Statement
- Objectives
- Design Model
- Requirements
  - Hardware
  - Software

- Dataset name

PatientID	Insured	numVisitors	Name	Sex	Age	RoomNum	Bill (in thousand)	docRef	Co
1	0	3	Braund, Mr. Owen Harris	male	22	A/5 21171	7.25		S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26	STON/O2. 3101282	7.925		S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	113803	53.1	C123	S
5	0	3	Allen, Mr. William Henry	male	35	373450	8.05		S

Reference: Hyukki Lee, Soohyung Kim, Jong Wook Kim, and Yon Dohn Chung, "Utility-preserving anonymization for health data publishing", BMC Med Inform Decis Mak. 2017; 17: 104. doi: 10.1186/s12911-017-0499-0.

Screenshots of the code and output

In [1]:
```
import pandas as pd
import numpy as np
import scipy.stats
%matplotlib inline
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
# get rid of warnings
import warnings
warnings.filterwarnings("ignore")
# get more than one output per Jupyter cell
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
# for functions we implement later
from utils import best_fit_distribution
from utils import plot_result
```

In [2]:
```
df = pd.read_csv("health_data.csv")
```

In [3]:
```
df.shape
df.head()
```

Out[3]: (891, 10)

Out[3]:

	PatientID	Insured	numVisitors	Name	Sex	Age	RoomNum	Bill (in thousand)	docRef	Condition
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	373450	8.0500	NaN	S

Screenshots of the code and output

In [4]:
```
df.drop(columns=["PatientID", "Name"], inplace=True) # dropped because unique for every row
df.drop(columns=["RoomNum", "docRef"], inplace=True) # dropped because almost unique for every row
df.dropna(inplace=True)
```

In [5]:
```
df.shape
df.head()
```

Out[5]: (713, 6)

Out[5]:

	Insured	numVisitors	Sex	Age	Bill (in thousand)	Condition
0	0	3	male	22.0	7.2500	S
1	1	1	female	38.0	71.2833	C
2	1	3	female	26.0	7.9250	S
3	1	1	female	35.0	53.1000	S
4	0	3	male	35.0	8.0500	S

Screenshots of the code and output

In [7]:
```
encoders = [(["Sex"], LabelEncoder()), (["Condition"], LabelEncoder())]
mapper = DataFrameMapper(encoders, df_out=True)
new_cols = mapper.fit_transform(df.copy())
df = pd.concat([df.drop(columns=["Sex", "Condition"]), new_cols], axis="columns")
```

In [8]:
```
df.shape
df.head()
```

Out[8]: (713, 6)

Out[8]:

	Insured	numVisitors	Age	Bill (in thousand)	Sex	Condition
0	0	3	22.0	7.2500	1	2
1	1	1	38.0	71.2833	0	0
2	1	3	26.0	7.9250	0	2
3	1	1	35.0	53.1000	0	2
4	0	3	35.0	8.0500	1	2

Screenshots of the code and output

```
In [9]: df.nunique()

Out[9]: Insured                 2
        numVisitors             3
        Age                    88
        Bill (in thousand)    220
        Sex                     2
        Condition               3
        dtype: int64

In [10]: categorical = []
         continuous = []

In [11]: for c in list(df):
             col = df[c]
             nunique = col.nunique()
             if nunique < 20:
                 categorical.append(c)
             else:
                 continuous.append(c)

In [12]: categorical

Out[12]: ['Insured', 'numVisitors', 'Sex', 'Condition']

In [13]: continuous

Out[13]: ['Age', 'Bill (in thousand)']
```

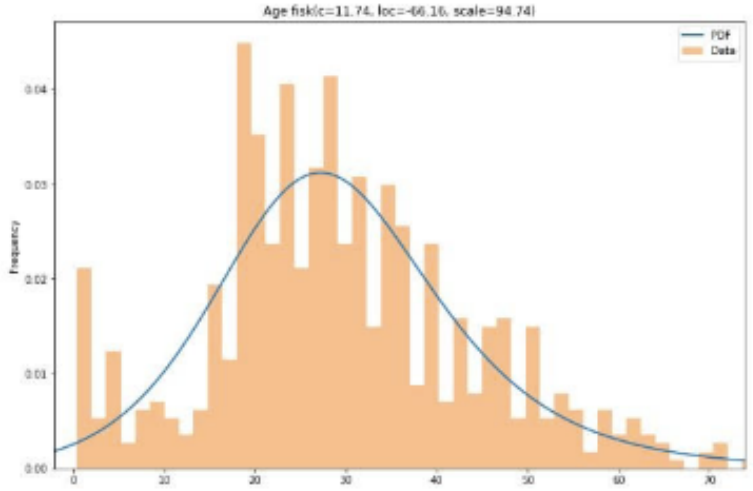
Screenshots of the code and output

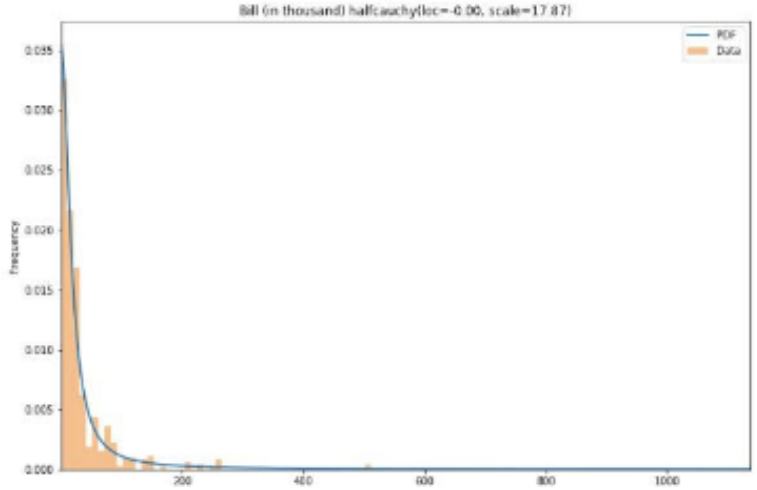
```
In [14]: for c in categorical:
             counts = df[c].value_counts()
             np.random.choice(list(counts.index), p=(counts/len(df)).values, size=5)

Out[14]: array([0, 0, 0, 0, 0])

Out[14]: array([1, 3, 3, 3, 2])

Out[14]: array([1, 0, 0, 0, 1])

Out[14]: array([2, 1, 2, 1, 2])

In [15]: # https://stackoverflow.com/a/37616966/1820480

In [16]: best_distributions = []

In [17]: # for c in continuous:
         #     data = df[c]
         #     best_fit_name, best_fit_params = best_fit_distribution(data, 50)
         #     best_distributions.append((best_fit_name, best_fit_params))

In [18]: best_distributions

Out[18]: []

In [19]: best_distributions = [
             ('fisk', (11.744665309421649, -66.15529969956657, 94.73575225186589)),
             ('halfcauchy', (-5.537941926133496e-09, 17.86796415175786))]
```

Analysis of the result

```
In [19]: best_distributions = [
             ('fisk', (11.744665309421649, -66.15529959956657, 94.73575225186589)),
             ('halfcauchy', (-5.537941926133498e-09, 17.867964151757 86))]
```

```
In [20]: plot_result(df, continuous, best_distributions)
```





Screenshots of the code and output

In [21]:
```
def generate_like_df(df, categorical_cols, continuous_cols, best_distributions, n, seed=0):
    np.random.seed(seed)
    d = {}

    for c in categorical_cols:
        counts = df[c].value_counts()
        d[c] = np.random.choice(list(counts.index), p=(counts/len(df)).values, size=n)

    for c, bd in zip(continuous_cols, best_distributions):
        dist = getattr(scipy.stats, bd[0])
        d[c] = dist.rvs(size=n, *bd[1])

    return pd.DataFrame(d, columns=categorical_cols+continuous_cols)
```

In [22]:
```
gendf = generate_like_df(df, categorical, continuous, best_distributions, n=100)
```

In [23]:
```
gendf.shape
gendf.head()
```

Out[23]: (100, 6)

Out[23]:

	Insured	numVisitors	Sex	Condition	Age	Bill (in thousand)
0	0	1	1	0	25.406552	9.474289
1	1	3	0	2	51.812626	11.859376
2	1	1	1	2	12.387505	19.327654
3	0	2	1	2	54.595218	43.251377
4	0	3	1	2	45.181993	10.322591

In [24]:
```
gendf.columns = list(range(gendf.shape[1]))
```

In [25]:
```
gendf.to_csv("output.csv", index_label="id")
```

In [26]:
```
gendf.shape
gendf.head()
```

Out[26]: (100, 6)

Out[26]:

	0	1	2	3	4	5
0	0	1	1	0	25.406552	9.474289
1	1	3	0	2	51.812626	11.859376
2	1	1	1	2	12.387505	19.327654
3	0	2	1	2	54.595218	43.251377
4	0	3	1	2	45.181993	10.322591

Applicability category

- Introduction
- Importance/Need of App
- Problem Statement
- Objectives
- Design Model
- Requirements
  - Hardware
  - Software
- Applicability cat y

Applicable in:

Health Car

Source: Frost & Sullivan

References

1. Hyukki Lee, Soohyung Kim, Jong Wook Kim, and Yon Dohn Chung, “Utility-preserving anonymization for health data publishing”, BMC Med_Inform Decis Mak. 2017; 17: 104. doi: 10.1186/s12911-017-0499-0.
2. Xiao-Bai Li, Jialun Qin, “Anonymizing and Sharing Medical Text Records”, Published in final edited form as: Inf Syst Res. 2017; 28(2): 332–352.
3. https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.html


Thank
You

YouTube Video link of project presentation

YouTube Link

TIMD 20W…	/ Replace with Title of Project*/

/* Replace with YouTube HTML link*/