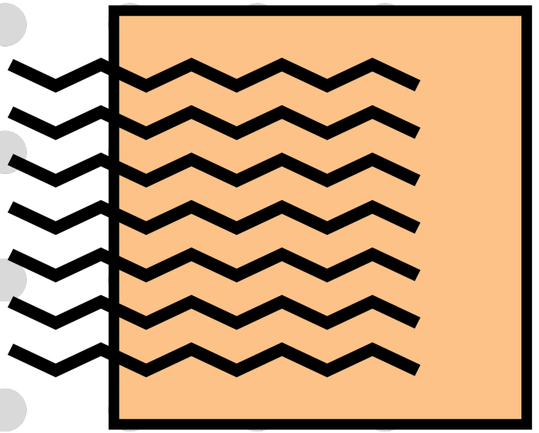


TEXT CLASSIFICATION FOR PROBLEM REPORTING IN BANGKOK'S TRAFFY FONDUE SYSTEM

JUFFNEE KHOTSAWAT
63605073

Master's degree of Data Science and Analytic
King Mongkut's Institute of Technology Ladkrabang

AGENDA

01 INTRODUCTION

02 OBJECTIVES

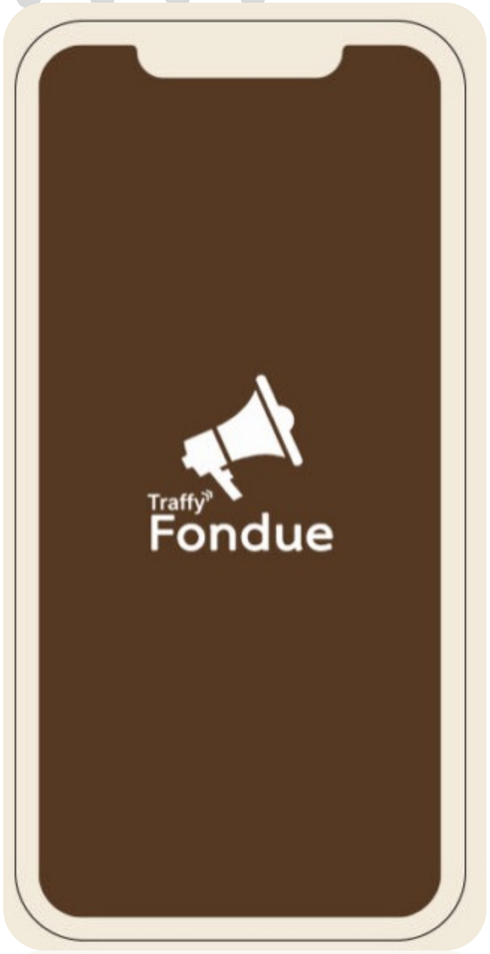
03 METHODOLOGY

04 RESULTS

05 CONCLUSIONS

01 INTRODUCTION

01 INTRODUCTION



WHAT IS “Traffy* Fondue” ?

- An application for receiving problem reports
- Anyone can problems report by using line chatbot
- Informant does not need to know the staff Or know who was responsible for the problem before

BENEFITS OF Traffy* Fondue

- Reduce troubleshooting time
- 78 M baht cost reduction
- Open data that developers can use

01 INTRODUCTION

Type of Problems Reported

1) Clean garbage
2) Electricity and water supply
3) The streetlight is broken
4) Roads, sidewalks
5) Damaged premise
6) Defective equipment and supplies
7) Risk point
8) Disasters: floods, fires, burns
9) Trees, smells, sounds, animals
10) Registration, public relations
11) Help
12) Health
13) Clues of corruption
14) other

01 INTRODUCTION

HOW TO USE Traffy* Fondue?

1. Add Traffy Fondue as a friend in LINE search ID “@traffyfondue”
2. Type the problem you want to report.
3. Take a picture of the problem.
4. Identify the problem type.
5. Share problem location
6. Identify the department

01 INTRODUCTION

No.	Field
1	ticket_id
2	type
3	organization
4	comment
5	coords
6	photo
7	address
8	district
9	subdistrict
10	province
11	Timestamp
12	state

WHAT ABOUT OPEN DATA OF Traffy* Fondue ?

01 INTRODUCTION

WHAT ABOUT PROBLEM?

	Rows
All Data	141,079
Missing data type	62,676
Missing data comment	221

Missing data type are

62,676 rows !

44.43% of all data

01 INTRODUCTION

HOW TO SOLVE PROBLEM?

“This problem was solved by TEXT CLASSIFICATION using NATURAL LANGUAGE PROCESSING (NLP)”

01 INTRODUCTION

WHAT ARE MODEL THAT WE USE?

1. Universal Sentence Encoder (USE)
2. Long Short-Term Memory (LSTM)
3. Deep Neural Network (DNN)
4. Convolutional Neural Network (CNN)

02 OBJECTIVES

02 OBJECTIVES

1. To study problem classification using natural language processing.
2. To solve problems identifying the type of problem report.
3. To compare the performance of model.

03 METHODOLOGY

03 METHODOLOGY

DATA UNDERSTANDING

DATA PREPARATION

MODELING

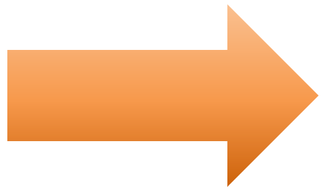
EVALUATION

CONCLUSION

03 METHODOLOGY

DATA UNDERSTANDING

No.	Field
1	ticket_id
2	type
3	organization
4	comment
5	coords
6	photo
7	address
8	district
9	subdistrict
10	province
11	Timestamp
12	state



No.	Field
2	type
4	comment

03 METHODOLOGY

DATA PREPARATION

1. Data Cleaning
2. Prepare data for model

03 METHODOLOGY

DATA PREPARATION

1. Data Cleaning

REMOVE MISSING DATA AND COMPLICATED DATA TYPE

```
1 df = traffy_df.dropna(subset=['type','comment']).reset_index(drop = True)
2 df
```

	type	comment
0	น้ำท่วม	น้ำท่วมขังค่ะ
1	คลอง	เสนอแนะเรื่องการปล่อยน้ำคลองเปรมประชากรจากเขตต...
2	ถนน,น้ำท่วม	ถนนเป็นหลุม น้ำท่วมขัง
3	ท่อระบายน้ำ,คลอง,น้ำท่วม	1.บ้านเป็นบ้านชั้นเดียว\nหลังสุดท้ายในซอย\n2.ค...
4	น้ำท่วม	น้ำท่วมขัง เป็นเวลานานไม่ได้รับการแก้ไข
...	...	...
78390	ถนน,คลอง	าระบายน้ำลงคลองบริเวณนั้น ช้ามาก เพราะปัจจุบั...
78391	ท่อระบายน้ำ	ไม่มีการ ลอก ท่อ แค่วางท่อระบายน้ำให้ใหญ่ขึ้น
78392	น้ำท่วม	จุดเสี่ยงน้ำท่วม จุดเฝ้าระวัง กรณีเขตยกเครื่อง...
78393	ถนน,น้ำท่วม	มีน้ำท่วมบ้าง เพราะพื้นถนนสูงกว่าถนนวิภาวดี ปร...
78394	ท่อระบายน้ำ	ขาดการจัดการขยะที่มีประสิทธิภาพ ส่งผลให้เกิดกา...

78395 rows × 2 columns

```
1 drop_values = [',']
2 new_df=traffy[~traffy['type'].str.contains('|'.join(drop_values))].reset_index(drop = True)
3 new_df
```

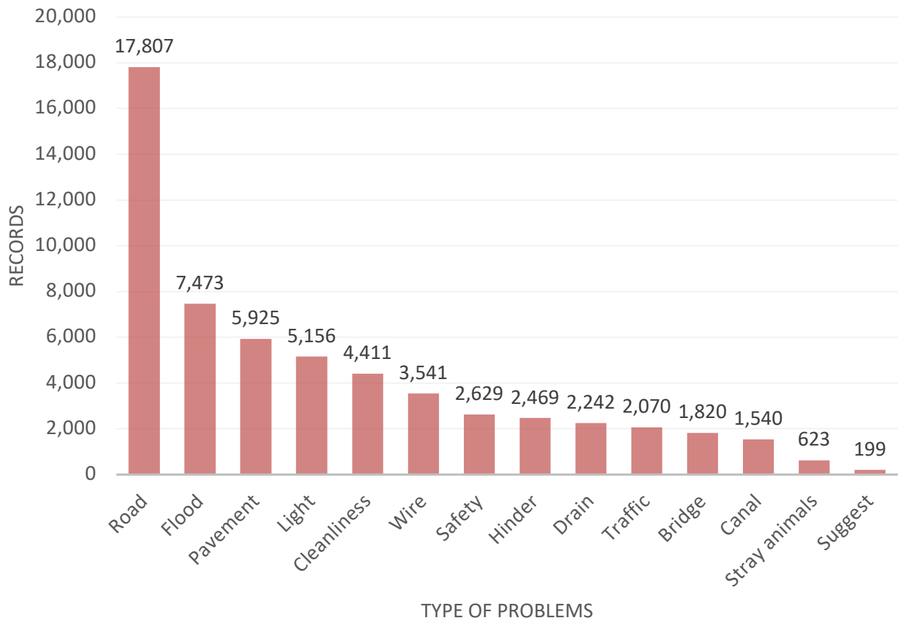
	type	comment
0	น้ำท่วม	น้ำท่วมขังค่ะ
1	คลอง	เสนอแนะเรื่องการปล่อยน้ำคลองเปรมประชากรจากเขตต...
2	น้ำท่วม	น้ำท่วมขัง เป็นเวลานานไม่ได้รับการแก้ไข
3	กีดขวาง	รถจอดกีดขวางป้ายรถเมล์
4	น้ำท่วม	น้ำท่วมขัง
...	...	...
57988	ท่อระบายน้ำ	ไม่มีท่อระบายน้ำ หลังบ้านติดบึง ตอนนี้น้ำในบึง...
57989	ถนน	ฝั่งของถนนด้านนี้พื้นที่ต่ำ อาจจะเพราะไม่ได้ลอ...
57990	ท่อระบายน้ำ	ไม่มีการ ลอก ท่อ แค่วางท่อระบายน้ำให้ใหญ่ขึ้น
57991	น้ำท่วม	จุดเสี่ยงน้ำท่วม จุดเฝ้าระวัง กรณีเขตยกเครื่อง...
57992	ท่อระบายน้ำ	ขาดการจัดการขยะที่มีประสิทธิภาพ ส่งผลให้เกิดกา...

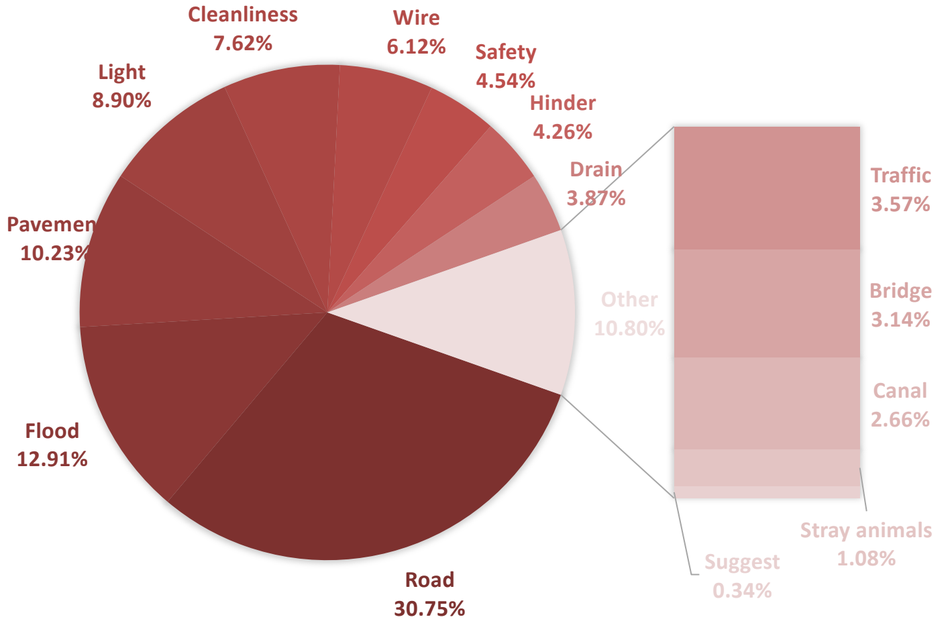
57993 rows × 2 columns

03 METHODOLOGY

DATA PREPARATION

AFTER DATA CLEANING

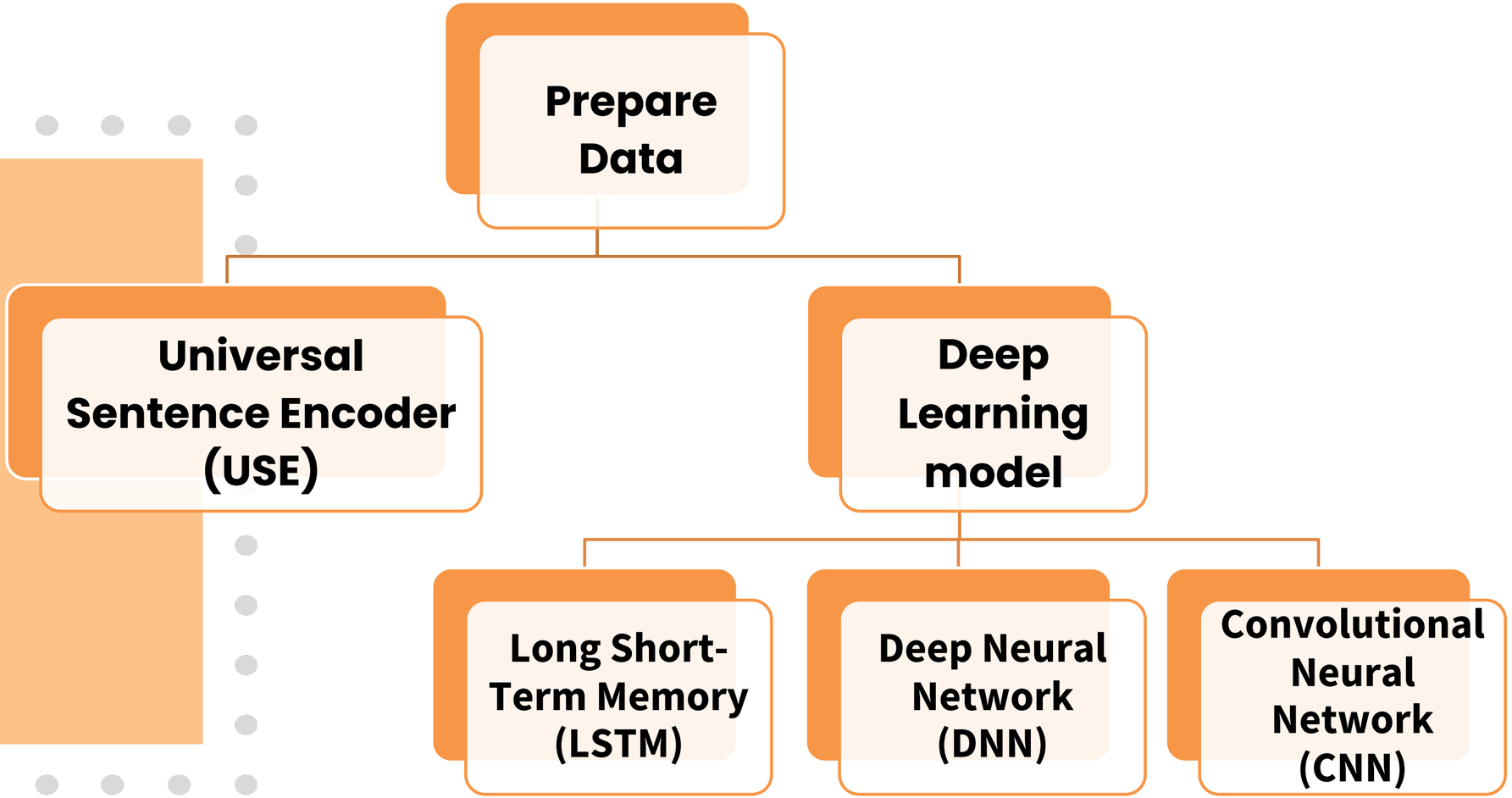




03 METHODOLOGY

DATA PREPARATION

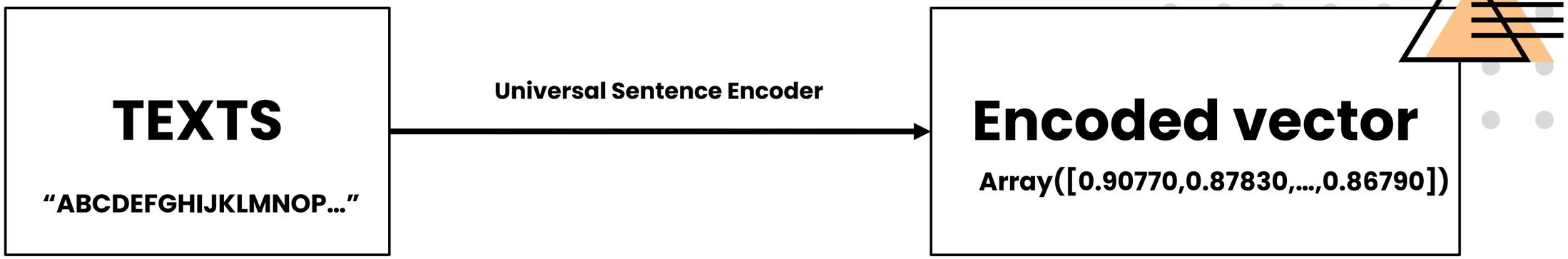
2. Prepare data for model



03 METHODOLOGY

DATA PREPARATION

2. Prepare data for Universal Sentence Encoder (USE)



03 METHODOLOGY

DATA PREPARATION

2. Prepare data for Deep Learning model

1. Word Segmentation
2. Convert word sequences into list of word Indices
3. Pad Word Indices
4. Convert labels into a list of label indices

03 METHODOLOGY

DATA PREPARATION

2. Prepare data for Deep Learning model

“บุคคลเร่ร่อน นอนริมทางเท้า และจอดรถเข็น ซอยสุขุมวิท 24”

Word Segmentation

['บุคคล','เร่ร่อน', ' ', 'นอน', 'ริม', 'ทางเท้า', ' ', 'และ', 'จอด', 'รถเข็น',' ', 'ซอย', 'สุขุมวิท', ' ‘, '24']

Convert word sequences into list of word Indices

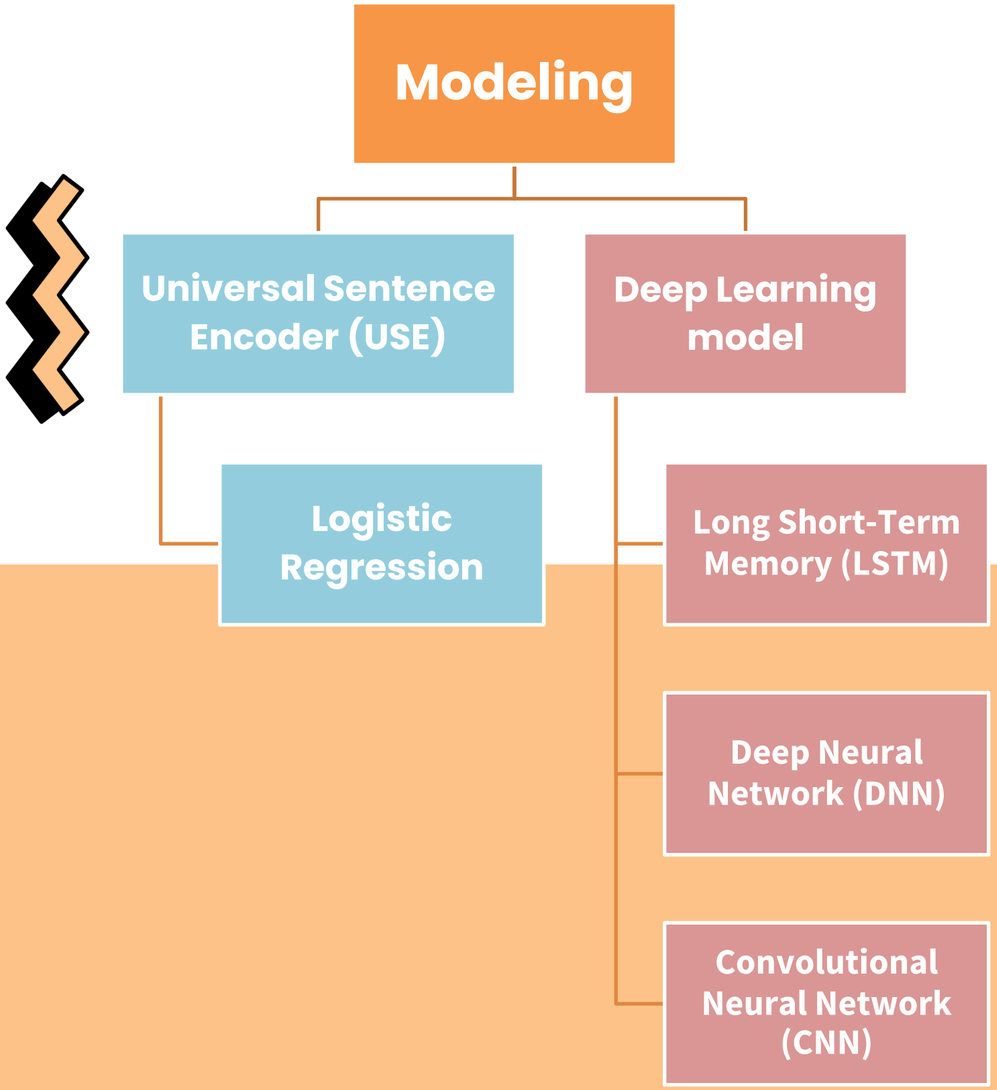
[612, 19752, 1, 696, 1823, 12791, 1, 3, 1816, 4627, 1, 1144, 2540, 1, 520]

Pad Word Indices

[0 0 0 ... 2540 1 520]

Convert labels into a list of label indices

Label: ทางเท้า → **[0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0.]**



03 METHODOLOGY

MODELING

Hyperparameters	Value
Input Value	1,456
Embedding	(1456 , 400)
Activation Function	Softmax
Optimizer Algorithm	Adam
Learning Rate	0.001
Epochs	30
Batch SIZE	32

03 METHODOLOGY

MODELING

Universal Sentence Encoder (USE)

```
1 from sklearn.linear_model import LogisticRegression
2
3 model = LogisticRegression(C=4, max_iter=1000, random_state=42)
```

```
1 model.fit(encoded_train_text, train_labels)
```

AFTER PEPARED DATA

By using Universal Sentence Encoder (USE) for modeling part we use

> **LOGISTIC REGRESSION MODEL**

03 METHODOLOGY

MODELING

Deep Learning model

> LONG SHORT-TERM MEMORY (LSTM)

CREATING MODEL

```
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 1456)]            0

embedding (Embedding)        (None, 1456, 400)         292474400

bidirectional (Bidirectiona  (None, 1456, 128)         238080
l)

bidirectional_1 (Bidirectio  (None, 64)                41216
nal)

dense (Dense)                (None, 14)                910

=================================================================
Total params: 292,754,606
Trainable params: 280,206
Non-trainable params: 292,474,400
_________________________________________________________________
```

TRAINING MODEL

```python
# Uncomment this section when training
lstm_weight_path_object_model='/content/drive/MyDrive/traffy/Model/lstm_model.h5'
# Add callback method for model checkpoint (save a checkpoint when get the best accuracy)
callbacks_list_object_model = [
    ModelCheckpoint(
        lstm_weight_path_object_model,
        save_best_only=True,
        save_weights_only=True,
        monitor='val_categorical_accuracy',
        mode='max',
        verbose=1
    )
]

verbose = 1
epochs = 30
batch_size = 32
print("train with {} epochs and {} batch size".format(epochs, batch_size))
history = model_lstm.fit(train_padded_wordinds, train_dummy_label, epochs=epochs, batch_size=batch_size, verbose=verbose,
                         callbacks=callbacks_list_object_model,
                         validation_data=(test_padded_wordinds, test_dummy_label))
```

03 METHODOLOGY

MODELING

Deep Learning model

> CONVOLUTIONAL NEURAL NETWORK (CNN)

CREATING MODEL

```
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 1456)]            0

 embedding (Embedding)       (None, 1456, 400)         292474400

 conv1d (Conv1D)             (None, 1454, 128)         153728

 conv1d_1 (Conv1D)           (None, 1452, 64)          24640

 global_max_pooling1d (Globa  (None, 64)               0
 lMaxPooling1D)

 dense (Dense)               (None, 14)                910

=================================================================
Total params: 292,653,678
Trainable params: 179,278
Non-trainable params: 292,474,400
_________________________________________________________________
```

TRAINING MODEL

```
cnn_weight_path_object_model='/content/drive/MyDrive/traffy/Model/Max_Pooling.h5'
# Add callback method for model checkpoint (save a checkpoint when get the best accuracy)
callbacks_list_object_model = [
    ModelCheckpoint(
        cnn_weight_path_object_model,
        save_best_only=True,
        save_weights_only=True,
        monitor='val_categorical_accuracy',
        mode='max',
        verbose=1
    )
]

verbose = 1
epochs = 30
batch_size = 32
print("train with {} epochs and {} batch size".format(epochs, batch_size))
history = model_max.fit(train_padded_wordinds, train_dummy_label, epochs=epochs, batch_size=batch_size, verbose=verbose,
                        callbacks=callbacks_list_object_model,
                        validation_data=(test_padded_wordinds, test_dummy_label))
```

03 METHODOLOGY

MODELING

Deep Learning model

> DEEP NEURAL NETWORK (DNN)

CREATING MODEL

```
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 1456)]            0

embedding (Embedding)        (None, 1456, 400)         292474400

bidirectional (Bidirectiona  (None, 1456, 128)         238080
l)

bidirectional_1 (Bidirectio  (None, 64)                41216
nal)

dense (Dense)                (None, 14)                910

=================================================================
Total params: 292,754,606
Trainable params: 280,206
Non-trainable params: 292,474,400
_________________________________________________________________
```

TRAINING MODEL

```python
# Uncomment this section when training
cnn_weight_path_object_model='/content/drive/MyDrive/traffy/Model/cnn_model_weight_object.h5'
# Add callback method for model checkpoint (save a checkpoint when get the best accuracy)
callbacks_list_object_model = [
        ModelCheckpoint(
            cnn_weight_path_object_model,
            save_best_only=True,
            save_weights_only=True,
            monitor='val_categorical_accuracy',
            mode='max',
            verbose=1
        )
]

verbose = 1
epochs = 30
batch_size = 32
print("train with {} epochs and {} batch size".format(epochs, batch_size))
history = model_cnn.fit(train_padded_wordinds, train_dummy_label, epochs=epochs, batch_size=batch_size, verbose=verbose,
                        callbacks=callbacks_list_object_model,
                        validation_data=(test_padded_wordinds, test_dummy_label))
```

03 METHODOLOGY

EVALUATION

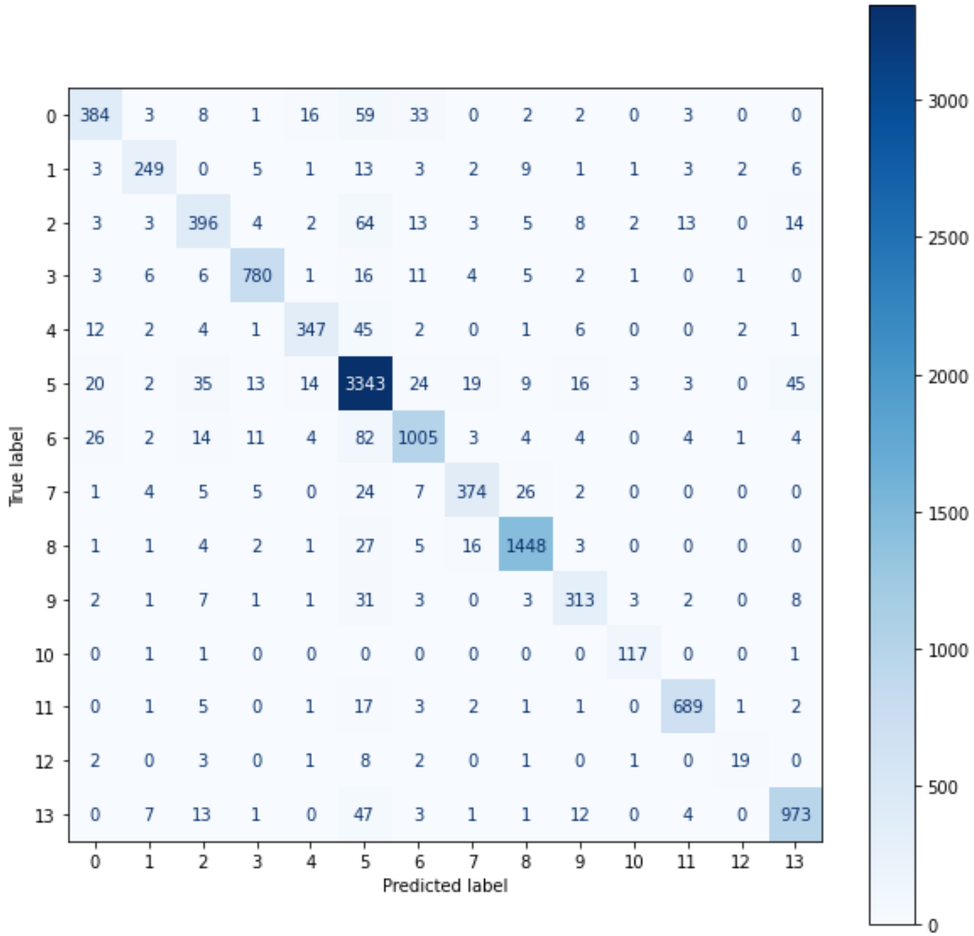
	precision	recall	f1-score	support
กีดขวาง	0.984	0.243	0.389	511
คลอง	0.920	0.268	0.416	298
ความปลอดภัย	0.840	0.079	0.145	530
ความสะอาด	0.899	0.352	0.506	836
จราจร	0.989	0.213	0.350	423
ถนน	0.397	0.999	0.568	3546
ทางเท้า	0.974	0.411	0.578	1164
ท่อระบายน้ำ	0.948	0.406	0.569	448
น้ำท่วม	0.996	0.367	0.536	1508
สะพาน	1.000	0.173	0.295	375
สัตว์จรจัด	1.000	0.292	0.452	120
สายไฟ	0.996	0.383	0.553	723
เสนอแนะ	0.000	0.000	0.000	37
แสงสว่าง	0.952	0.318	0.477	1062
accuracy			0.527	11581
macro avg	0.850	0.322	0.417	11581
weighted avg	0.785	0.527	0.500	11581

“ ACCURACY,
PRECISION,
RECALL,
F1-SCORE,
MARCO AVERAGE,
AND WEIGHTED AVERAGE

04 RESULTS

04 RESULTS

Universal Sentence Encoder (USE)

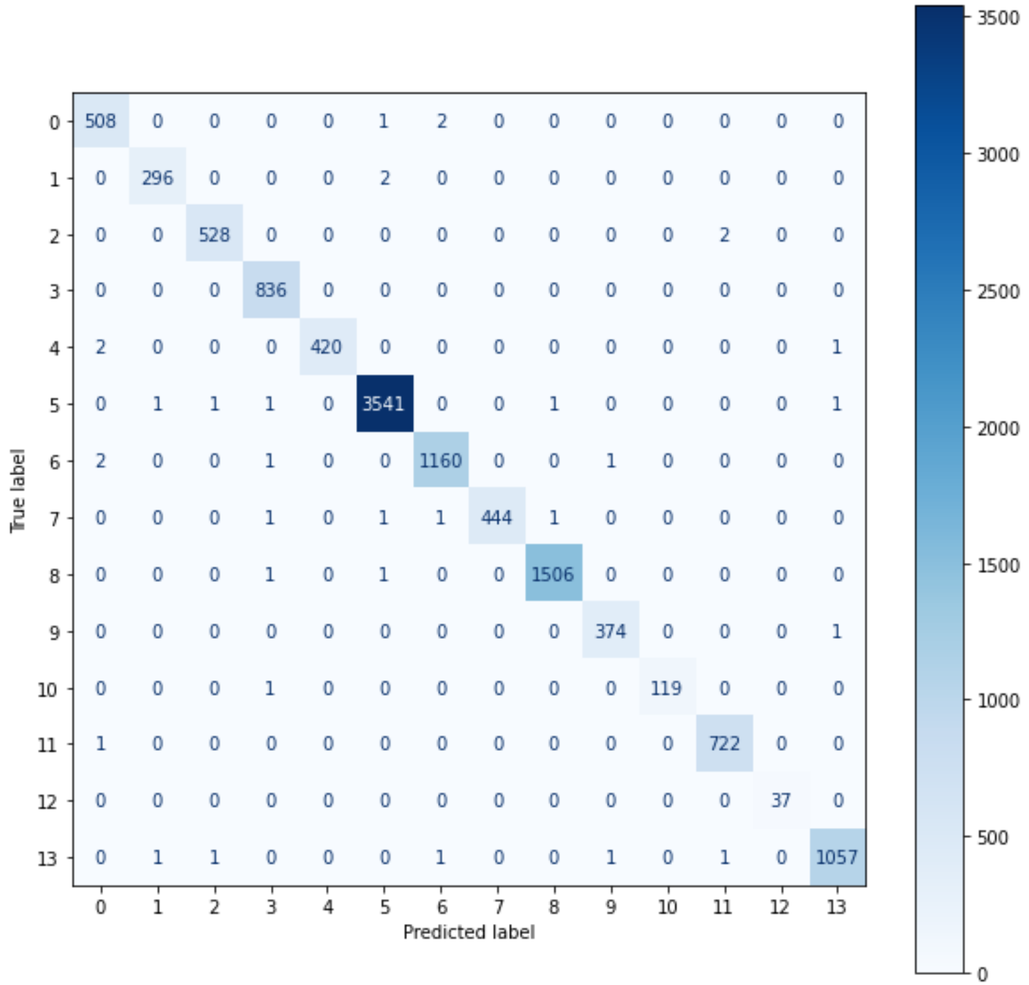


	Precision	Recall	F1-Score	Support
กีดขวาง	0.840	0.751	0.793	511
คลอง	0.883	0.836	0.859	298
ความปลอดภัย	0.790	0.747	0.768	530
ความสะอาด	0.947	0.933	0.940	836
จราจร	0.892	0.820	0.855	423
ถนน	0.885	0.943	0.913	3546
ทางเท้า	0.902	0.863	0.882	1164
ท่อระบายน้ำ	0.882	0.835	0.858	448
น้ำท่วม	0.956	0.960	0.958	1508
สะพาน	0.846	0.835	0.840	375
สัตว์จรจัด	0.914	0.975	0.944	120
สายไฟ	0.956	0.953	0.954	723
เสนอแนะ	0.731	0.514	0.603	37
แสงสว่าง	0.923	0.916	0.920	1062

Accuracy	90.10%
Marco Average	86.30%
Weighted Average	90.00%

04 RESULTS

Long Short-Term Memory (LSTM)

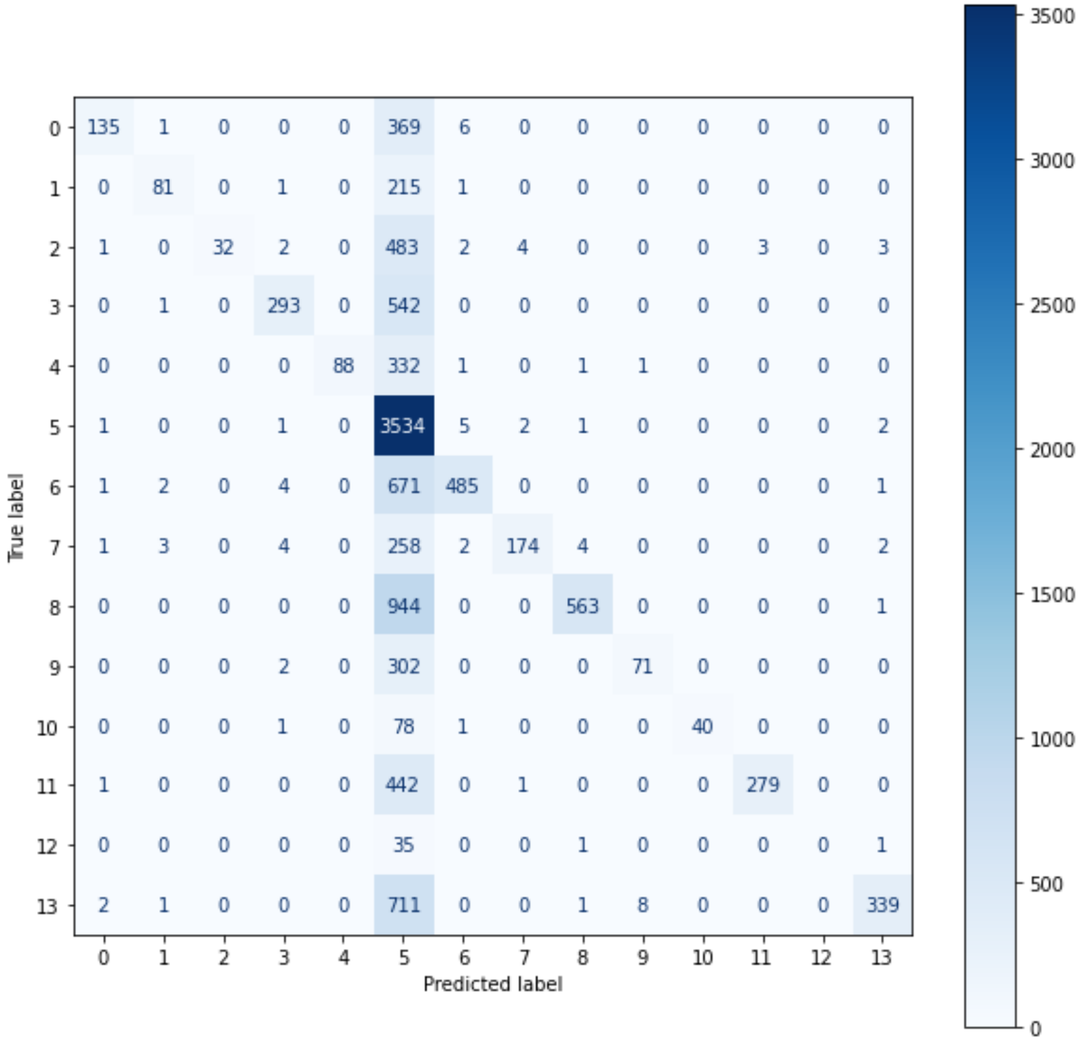


	Precision	Recall	F1-Score	Support
กีดขวาง	0.990	0.994	0.992	511
คลอง	0.993	0.993	0.993	298
ความปลอดภัย	0.996	0.996	0.996	530
ความสะอาด	0.994	1.000	0.997	836
จราจร	1.000	0.993	0.996	423
ถนน	0.999	0.999	0.999	3546
ทางเท้า	0.997	0.997	0.997	1164
ท่อระบายน้ำ	1.000	0.991	0.996	448
น้ำท่วม	0.999	0.999	0.999	1508
สะพาน	0.995	0.997	0.996	375
สัตว์จรจัด	1.000	0.992	0.996	120
สายไฟ	0.996	0.999	0.997	723
เสนอแนะ	1.000	1.000	1.000	37
แสงสว่าง	0.997	0.995	0.996	1062

Accuracy	99.70%
Marco Average	99.60%
Weighted Average	99.70%

04 RESULTS

Deep Neural Network (DNN)

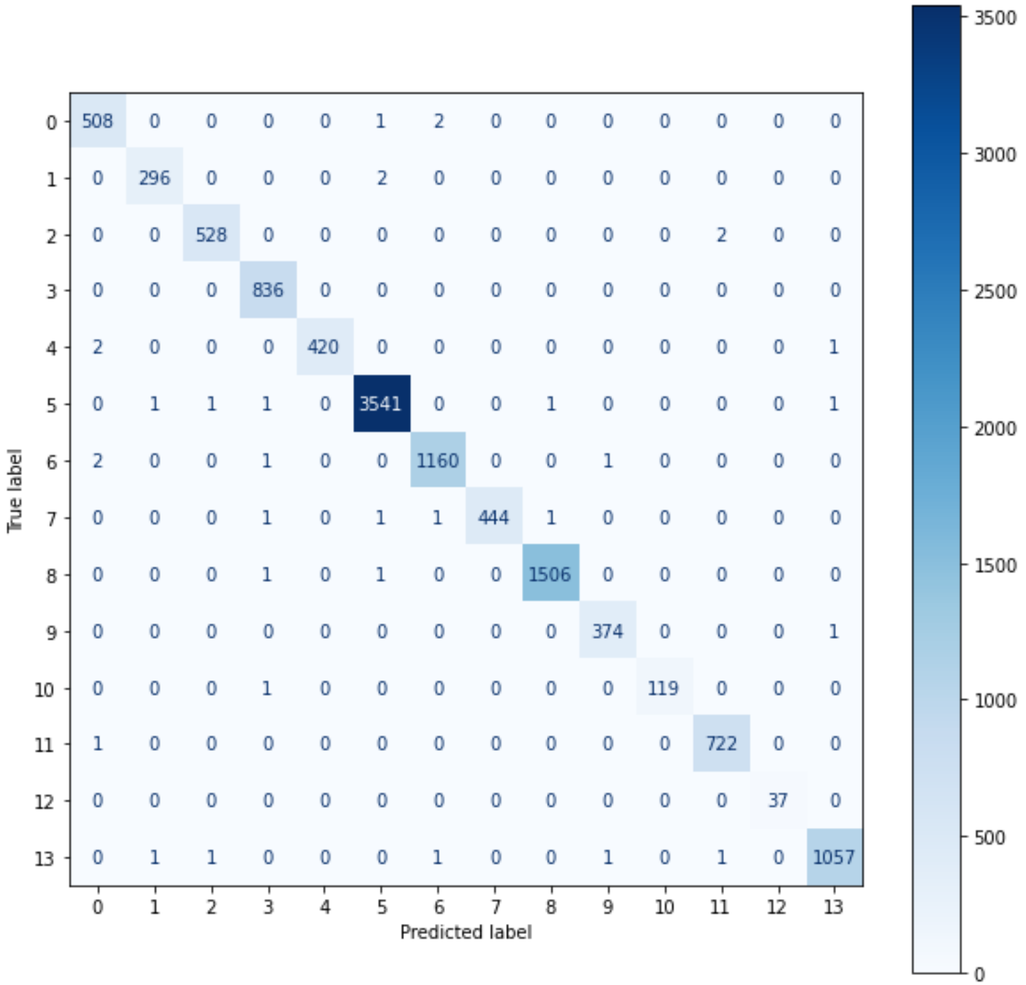


	Precision	Recall	F1-Score	Support
กีดขวาง	0.951	0.264	0.413	511
คลอง	0.910	0.272	0.419	298
ความปลอดภัย	1.000	0.060	0.114	530
ความสะอาด	0.951	0.350	0.512	836
จราจร	1.000	0.208	0.334	423
ถนน	0.396	0.997	0.567	3546
ทางเท้า	0.964	0.417	0.582	1164
ท่อระบายน้ำ	0.961	0.338	0.553	448
น้ำท่วม	0.986	0.373	0.542	1508
สะพาน	0.887	0.189	0.312	375
สัตว์จรจัด	1.000	0.333	0.500	120
สายไฟ	0.989	0.386	0.555	723
เสนอแนะ	0.000	0.000	0.000	37
แสงสว่าง	0.971	0.319	0.481	1062

Accuracy	52.80%
Marco Average	42.10%
Weighted Average	50.20%

04 RESULTS

Convolutional Neural Network (CNN)

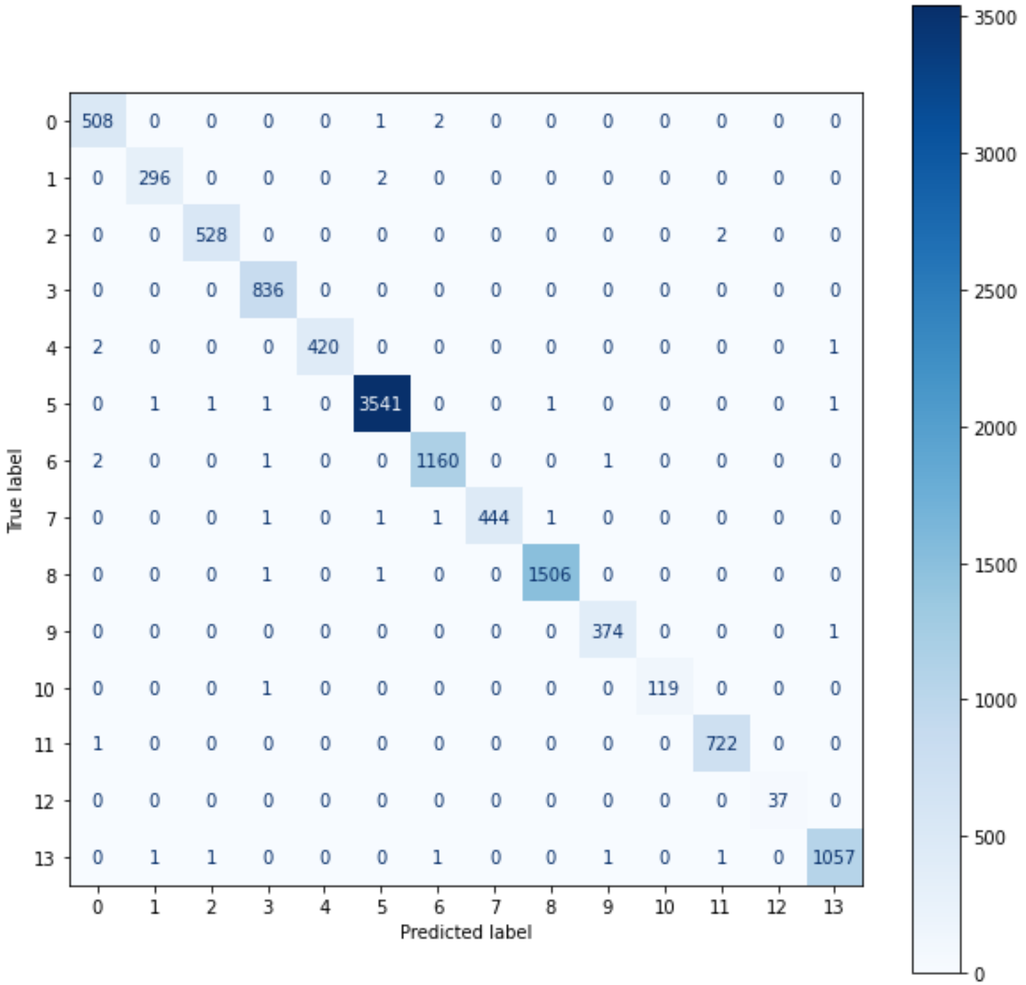


	Precision	Recall	F1-Score	Support
กีดขวาง	0.990	0.994	0.992	511
คลอง	0.993	0.993	0.993	298
ความปลอดภัย	0.996	0.996	0.996	530
ความสะอาด	0.994	1.000	0.997	836
จราจร	1.000	0.993	0.996	423
ถนน	0.999	0.999	0.999	3546
ทางเท้า	0.997	0.997	0.997	1164
ท่อระบายน้ำ	1.000	0.991	0.996	448
น้ำท่วม	0.999	0.999	0.999	1508
สะพาน	0.995	0.997	0.996	375
สัตว์จรจัด	1.000	0.992	0.996	120
สายไฟ	0.996	0.999	0.997	723
เสนอแนะ	1.000	1.000	1.000	37
แสงสว่าง	0.997	0.995	0.996	1062

Accuracy	95.60%
Marco Average	94.60%
Weighted Average	95.60%

04 RESULTS

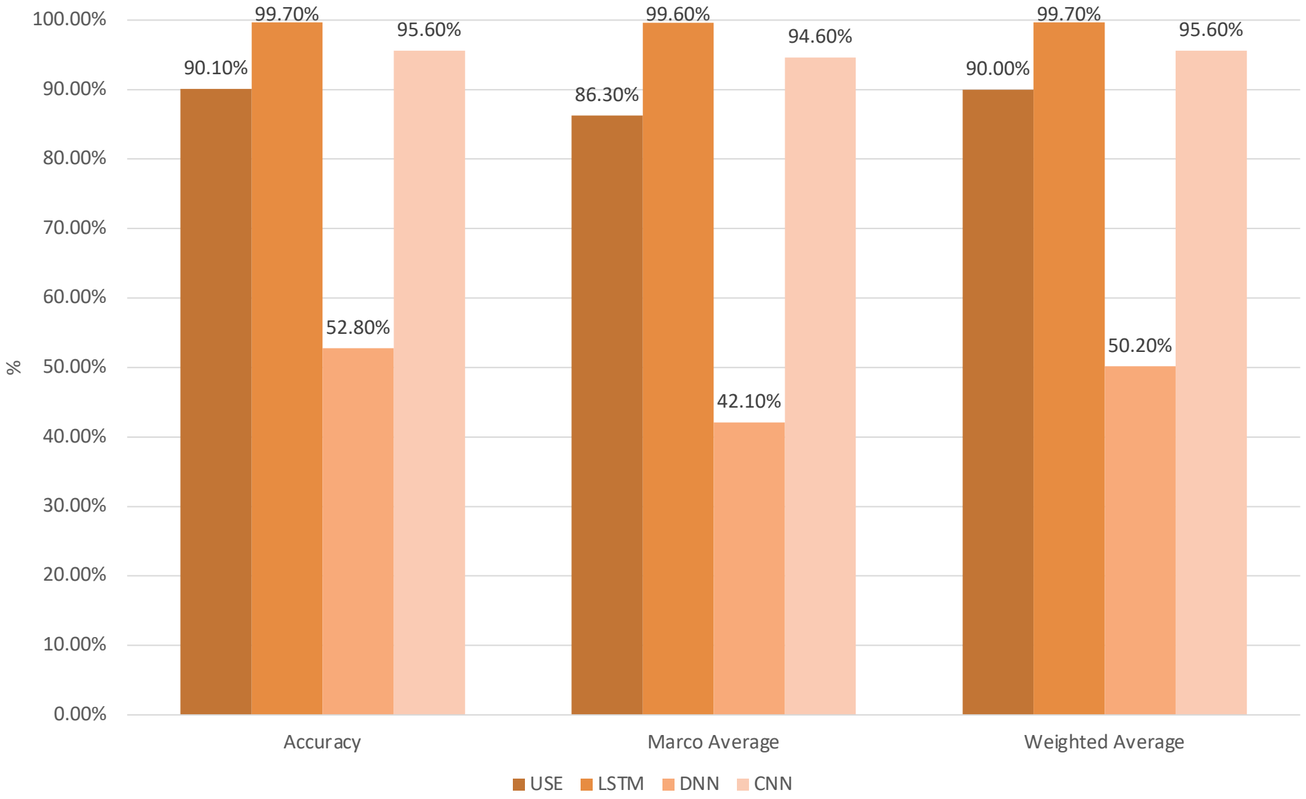
Convolutional Neural Network (CNN)

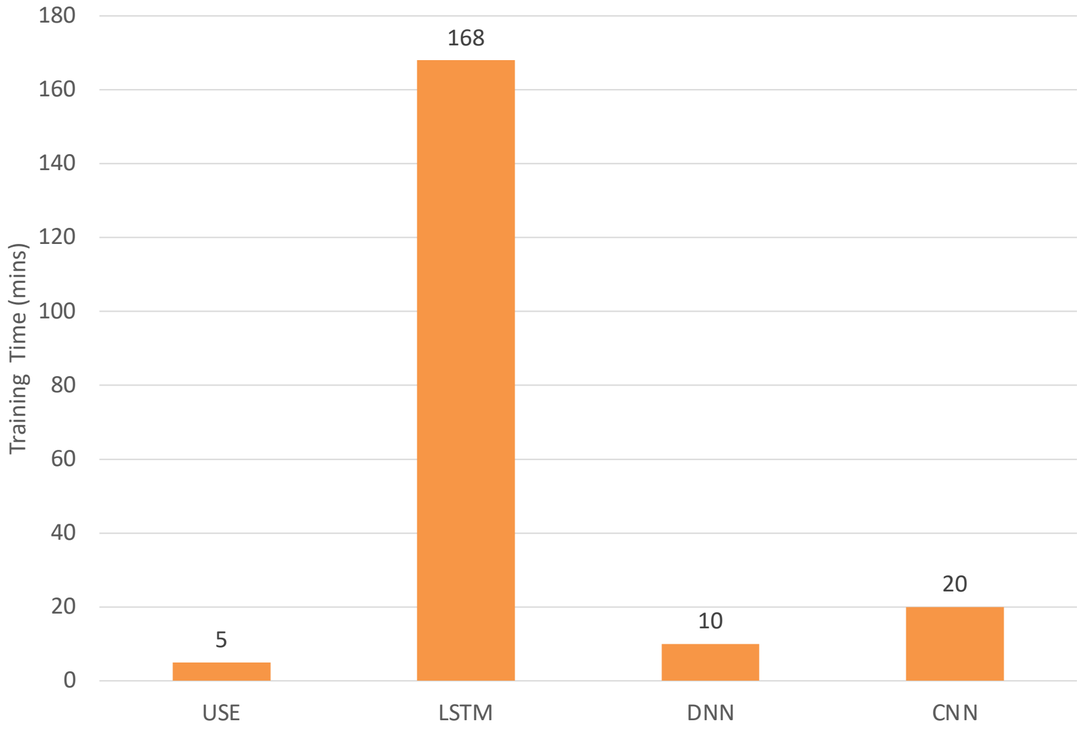


	Precision	Recall	F1-Score	Support
กีดขวาง	0.990	0.994	0.992	511
คลอง	0.993	0.993	0.993	298
ความปลอดภัย	0.996	0.996	0.996	530
ความสะอาด	0.994	1.000	0.997	836
จราจร	1.000	0.993	0.996	423
ถนน	0.999	0.999	0.999	3546
ทางเท้า	0.997	0.997	0.997	1164
ท่อระบายน้ำ	1.000	0.991	0.996	448
น้ำท่วม	0.999	0.999	0.999	1508
สะพาน	0.995	0.997	0.996	375
สัตว์จรจัด	1.000	0.992	0.996	120
สายไฟ	0.996	0.999	0.997	723
เสนอแนะ	1.000	1.000	1.000	37
แสงสว่าง	0.997	0.995	0.996	1062

Accuracy	95.60%
Marco Average	94.60%
Weighted Average	95.60%

04 RESULTS


100.00%
90.00%
80.00%
70.00%
60.00%
50.00%
40.00%
30.00%
20.00%
10.00%
0.00%
%
90.10%
99.70%
52.80%
95.60%
86.30%
99.60%
42.10%
94.60%
90.00%
99.70%
50.20%
95.60%
Accuracy
Marco Average
Weighted Average
USE
LSTM
DNN
CNN


180
160
140
120
100
80
60
40
20
0
Training Time (mins)
5
168
10
20
USE
LSTM
DNN
CNN

05 CONCLUSIONS

CONCLUSIONS

	Accuracy	Marco Average	Weighted Average	Time
USE	90.10%	86.30%	90.00%	5 min
LSTM	99.70%	99.60%	99.70%	168 min
DNN	52.80%	42.10%	50.20%	10 min
CNN	95.60%	94.60%	95.60%	20 min

Long Short-Term Memory (LSTM)

Highest Accuracy

Universal Sentence Encoder (USE)

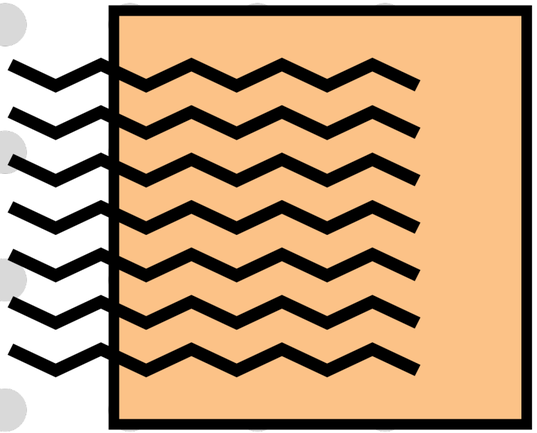
Least time for training

FUTURE WORK

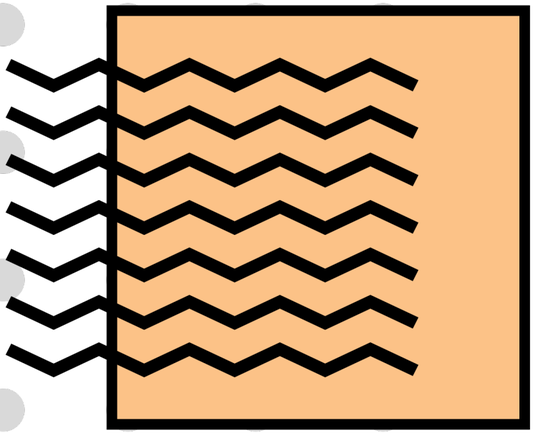
USE ANOTHER WORD SEGMENTATION

- attacut
- deepcut

> PRIORITY ORGANIZATION
> USE SOCIAL LISTENING FOR IDENTIFY
> PRIORITY ORGANIZATION



Q &A



THANKS, YOU