Global Spectrum Monitoring Forum

Future Technology for Spectrum Monitoring
Artificial Intelligence and its Application

r

ETRI

Contents

* Artificial Intelligence
** Spectrum Monitoring

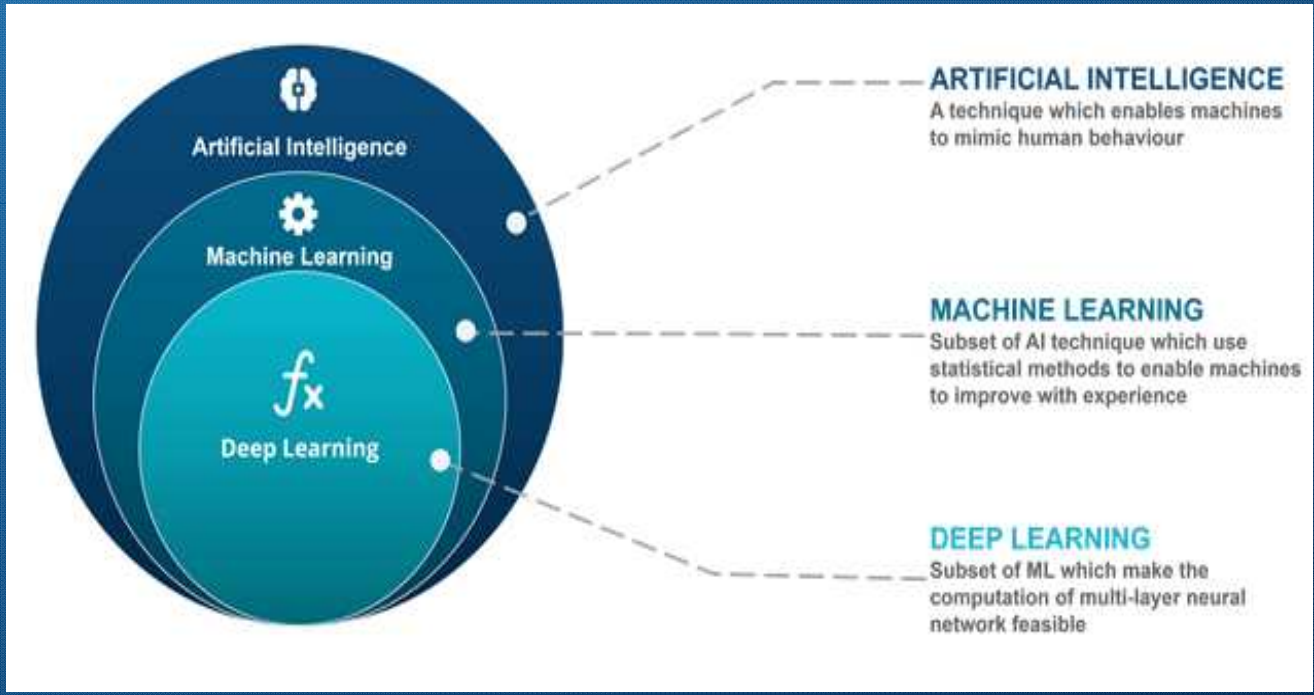
I Introduction to AI

II Modulation Classification by AI

III AI Application for ETRI SM

What is AI

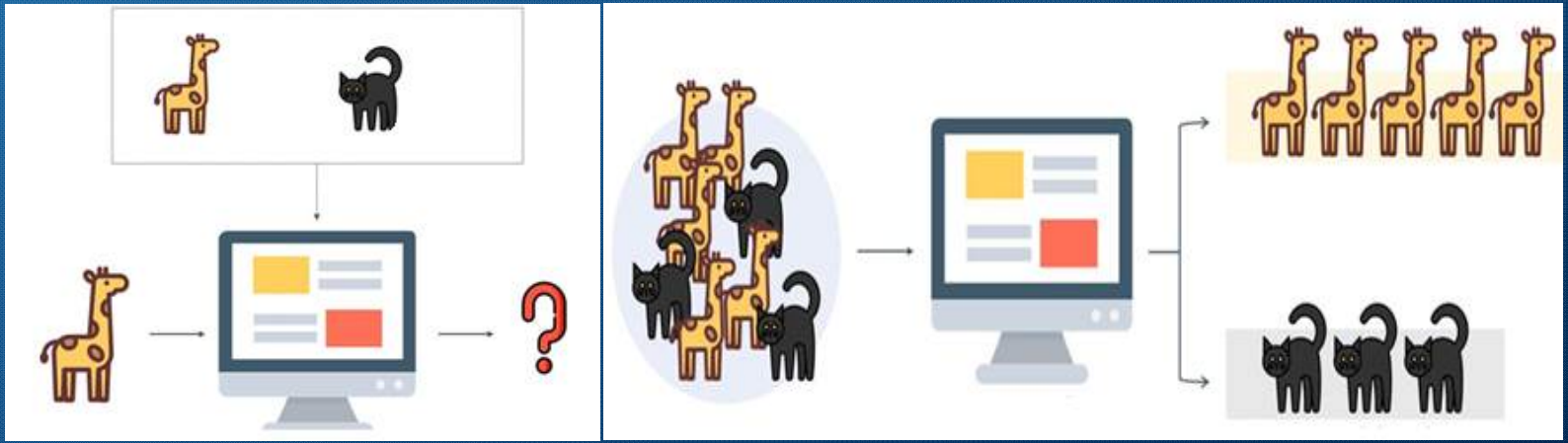
- **Intelligence that uses a computer to implement some or all of human intellectual such as cognition and learning**
  - ✓ Utilize accumulated data to predict patterns of new data
  - ✓ Machine Learning(ML) and Deep Learning(DL)
  - ✓ AI ∋ ML ∋ DL



https://m.blog.naver.com/jevida/221843366216

What can we do with AI technology

- **Classification and Prediction(Left figure)**
  - ✓ Classifying new data or predict categories based on past data and its labeling*
    - Separation of normal/spam e-mail
    - Trend in house prices and sales and profits of company
    - **Modulation identification**
- **Grouping(Right)**
  - ✓ Data grouping based on similarities between data
    - Classification of customer purchasing patterns
    - Determination of defects/failures through by image analysis

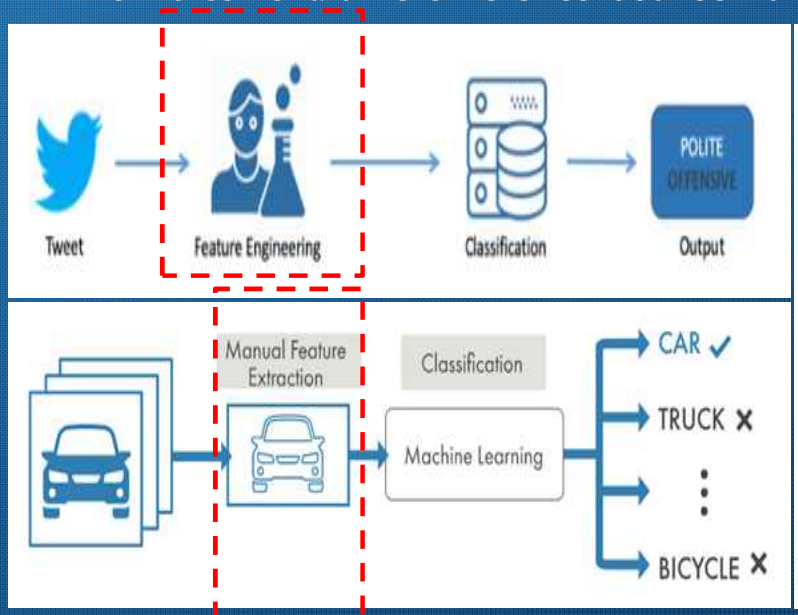


https://www.samsungsds.com/global/ko/support/insights/Generative-adversarial-network-AI.html

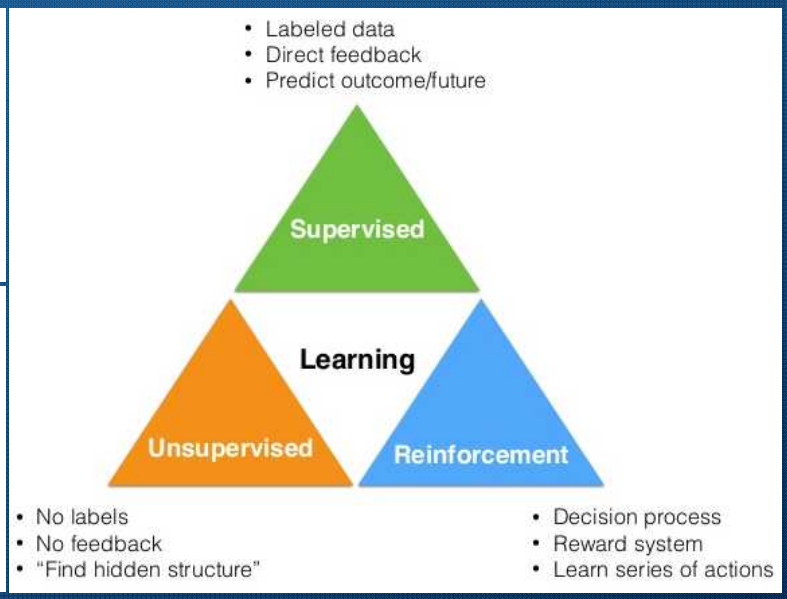
* ML is interested in predicting the future and data mining is interested in solving current problems

Some features of ML

- **Requires feature engineering based on statistics**
  - ✓ The process of reducing data complexity and making patterns more visible in learning algorithm
  - ✓ Requires time and expertise
- **Existence of learning method according to purpose**
  - ✓ Supervised: data with answer -> classification, prediction
  - ✓ Unsupervised: data with no answer -> tendency, grouping
  - ✓ Reinforcement: time difference between the question and the answer -> reward, competition



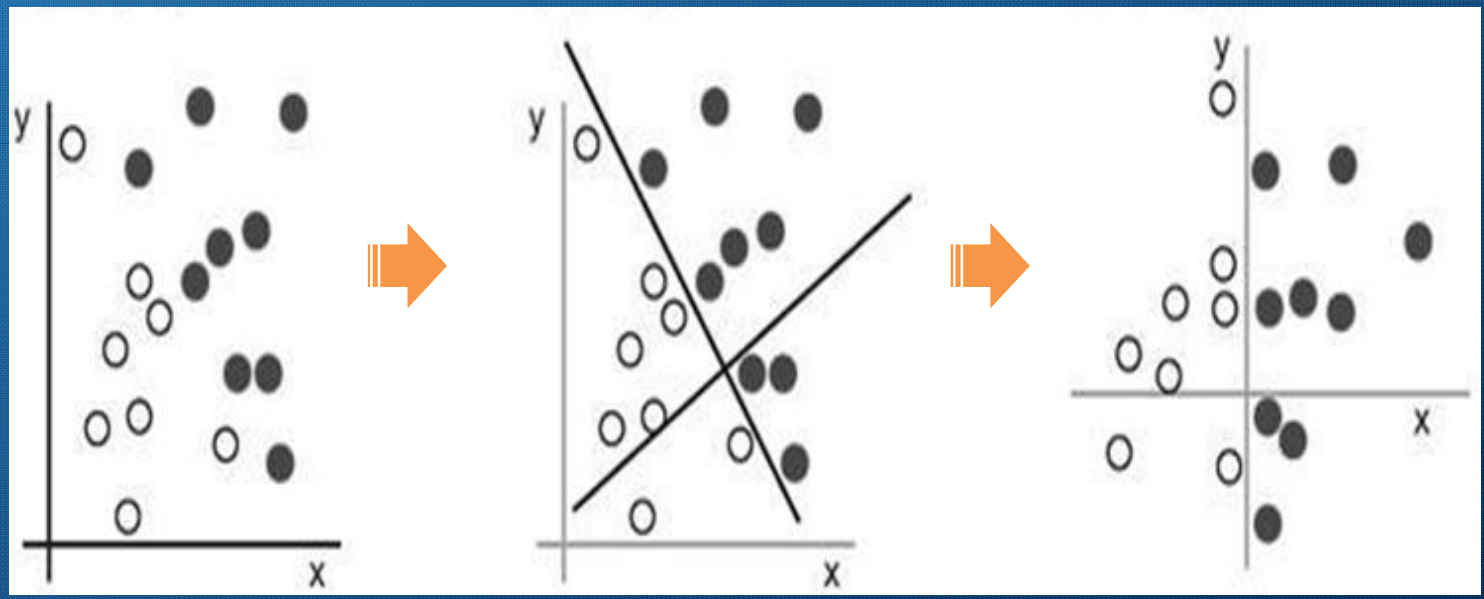
https://blogs.mathworks.com/pick/2017/06/02/deep-learning-tutorial-series



http://solarisailab.com/archives/1785

Meaning of learning in ML

- **The process of finding a better expression**
  - ✓ Input: coordinates of points
  - ✓ Output: colors of points
  - ✓ Criterion of performance evaluation: classification success rate
    - If x > 0, then it is black point and if x < 0, then white point (simple compared with original(left) or coordinate transformation(center)

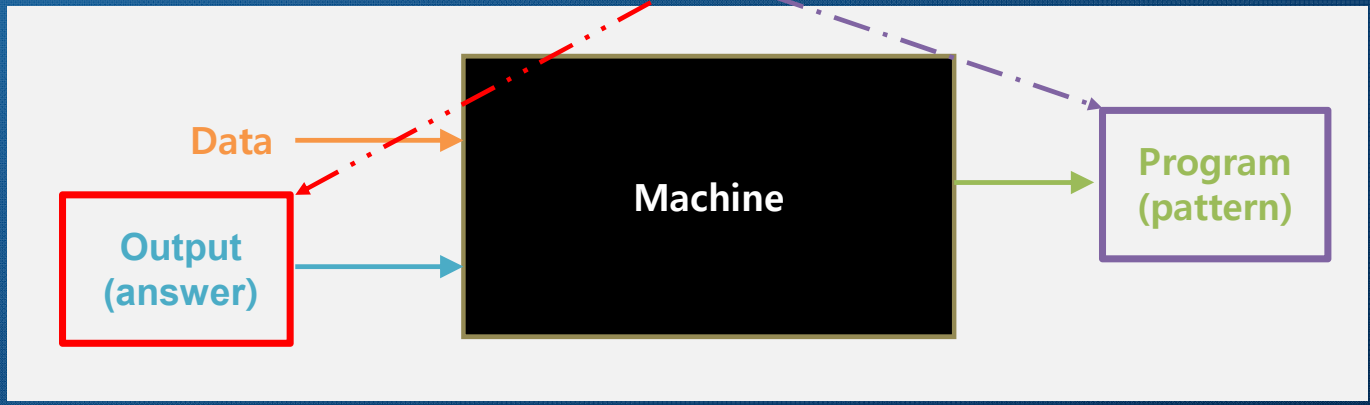


https://www.msit.go.kr/webzine/posts.do?postIdx=337

Programming and ML

- Programming

- ML



What do we need to implement ML in practice

- **Coding with python**
  - ✓ Build a suitable work environment (IDE*, Integrated Development Environment)
  - ✓ Use the required libraries and modules by importing them
- **Select suitable ML models**
  - ✓ Linear
  - ✓ Determination tree / Determination tree ensemble
  - ✓ Kernel Support Vector Machine
  - ✓ K-Nearest Neighbor

```
from FabricateDataset_forLearning import dataset
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
X=dataset.iloc[:,:(len(dataset.columns)-1)]
y=dataset.iloc[:,(len(dataset.columns)-1)]
X_train,X_test,y_train,y_test=train_test_split(X,y,stratify=y,test_size=0.25,random_st
forest=RandomForestClassifier(n_estimators=100,random_state=0,max_depth=20,n_jobs=-1)
forest.fit(X_train, y_train)

import pickle
pickle.dump(forest.fit(X_train,y_train),open('_TrainingModel.h5','wb'))
```

```
import pickle
learned_model=pickle.load(open('../UsingFunction/_TrainingModel.h5','rb'))
prediction=learned_model.predict(X_what)
```

Loading pre-processed data

Loading module

Constructing dataset

Composing training and test data

Performing learning

Saving learning results

Loading learning model

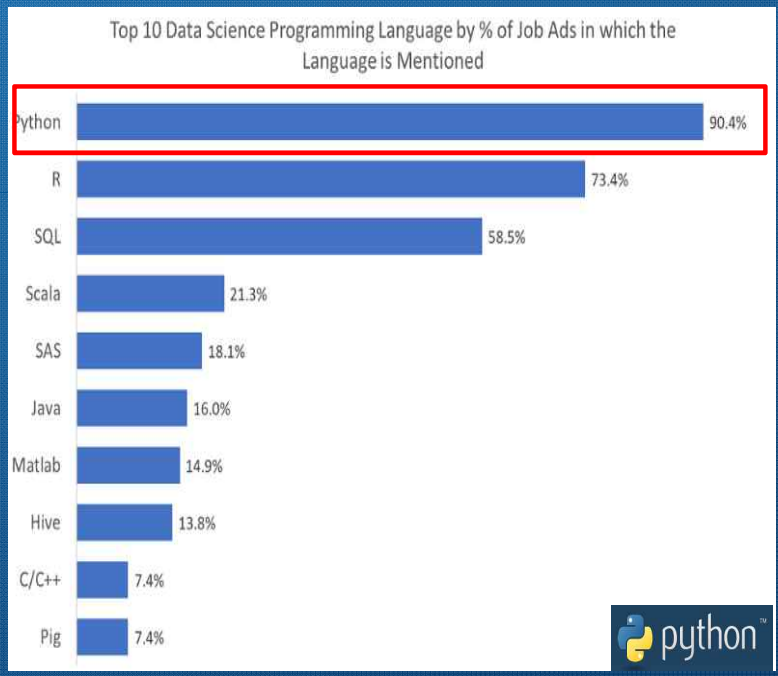
Prediction

* Jupyter notebook, Pycharm, Visual studio code, Spyder, Eclipse, etc

Python since 1989 by Guido van Rossum

- Simple and easy to learn
- Free and open source, Extensive libraries
- High-level language, Interpreted (vs compiler)
- Portable*: Windows, Linux, etc.

< Data science, 2020>



https://blog.simpliv.com/

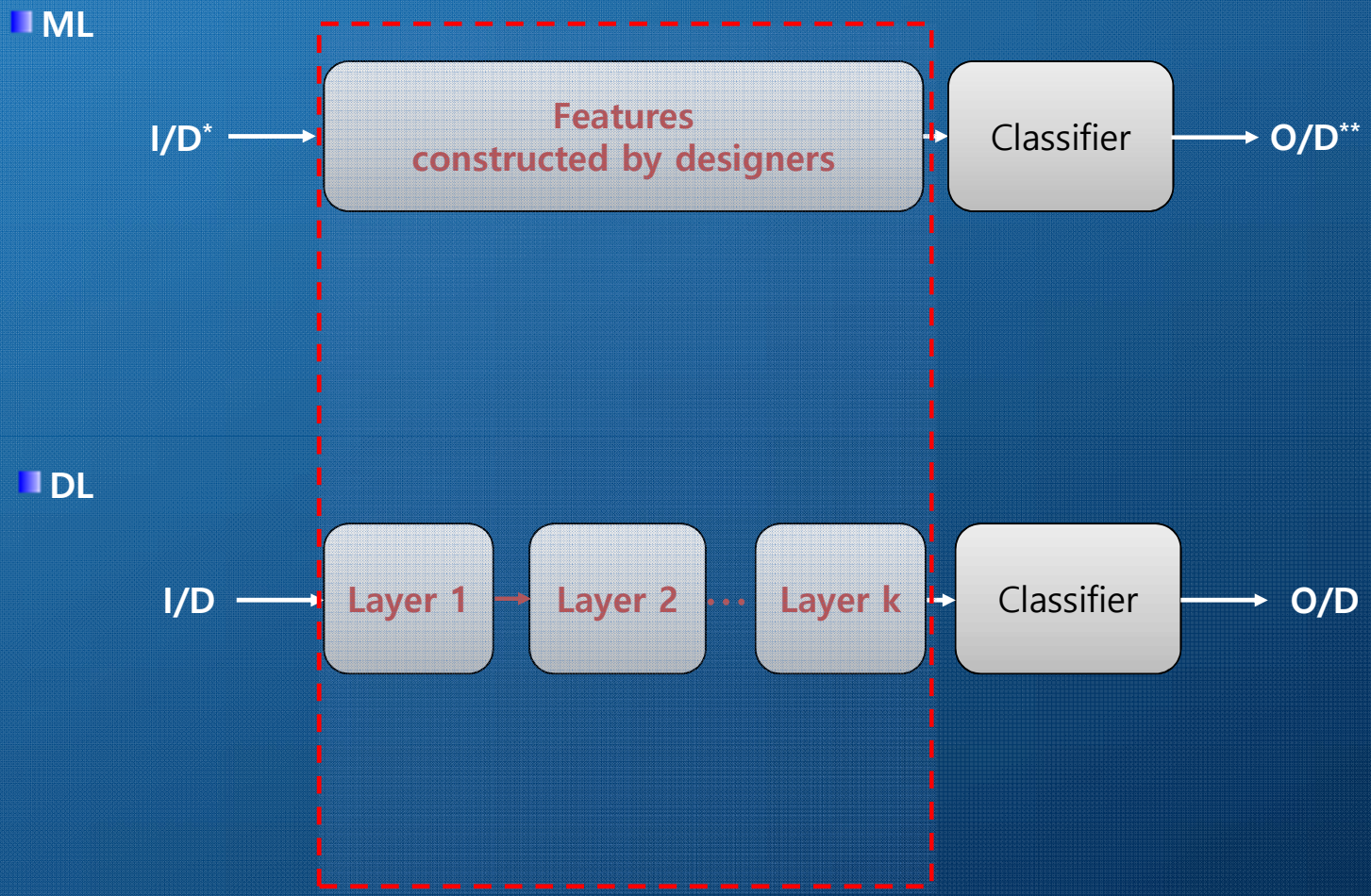
< Programming, 2019/2020 >

Jul 2020	Jul 2019	Change	Programming Language	Ratings	Change
1	2	^	C	16.45%	+2.24%
2	1	v	Java	15.10%	+0.04%
3	3		Python	9.09%	-0.17%
4	4		C++	6.21%	-0.49%
5	5		C#	5.25%	+0.88%
6	6		Visual Basic	5.23%	+1.03%
7	7		JavaScript	2.48%	+0.18%
8	20	^^	R	2.41%	+1.57%
9	8	v	PHP	1.90%	-0.27%
10	13	^	Swift	1.43%	+0.31%
11	9	v	SQL	1.40%	-0.58%
12	16	^^	Go	1.21%	+0.19%
13	12	v	Assembly language	0.94%	-0.45%
14	19	^^	Perl	0.87%	-0.04%
15	14	v	MATLAB	0.84%	-0.24%
16	11	vv	Ruby	0.81%	-0.83%
17	30	^^	Scratch	0.72%	+0.35%
18	33	^^	Rust	0.70%	+0.36%
19	23	^^	PL/SQL	0.68%	-0.01%
20	17	v	Classic Visual Basic	0.66%	-0.35%

https://www.tiobe.com/tiobe-index/

* Easy to transplant to other S/W

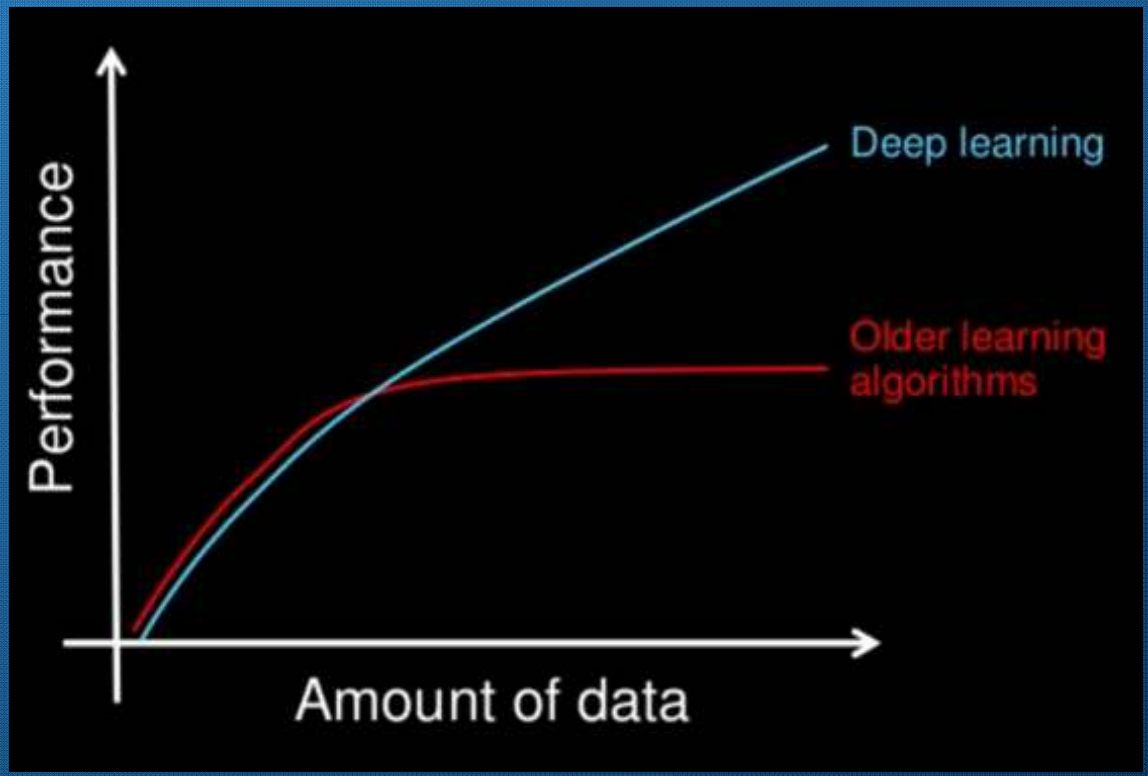
Difference between ML and DL



* Input data
** Output data

Importance of data

- It should be of good qualities data
- Whether we use ML or DL depends on the amount of data and features*
- DL is not necessarily better than ML



https://brunch.co.kr/@itschloe1/8

* Image and speech recognition is already well known for its superior DL

I Introduction to AI

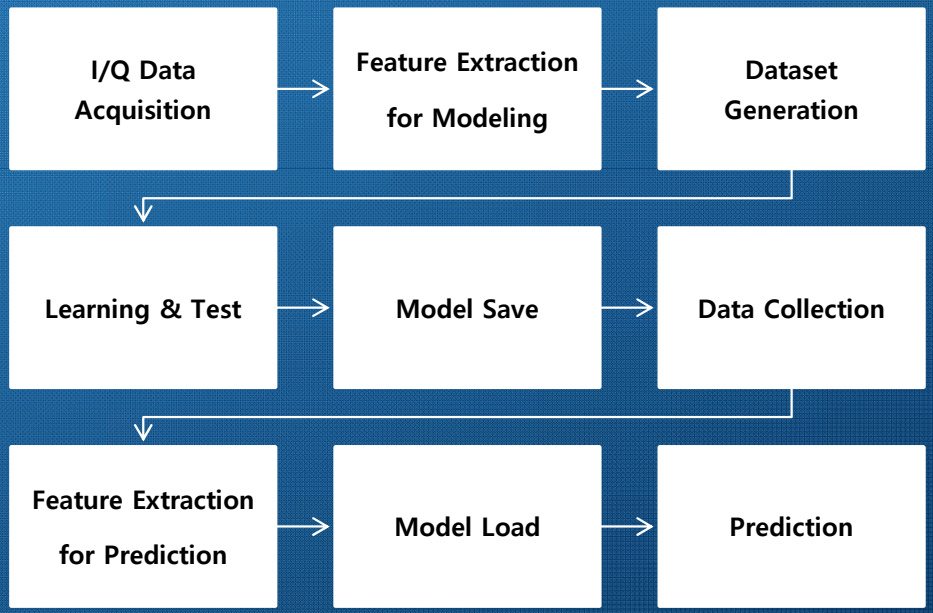
II Modulation Classification by AI

III AI Application on sM

Process of classifying signals based on ML

- **Extracted features calculated from I/Q data* of major broadcasting and communication signals are used as input**
  - Feature value is a numerical value of the unique spectral power of each modulated signal, the standard deviation of amplitude and phase, spectrum symmetry, etc.
  - A random forest, one of the supervised learning models, is applied

$$e^{jx} = I + Q$$

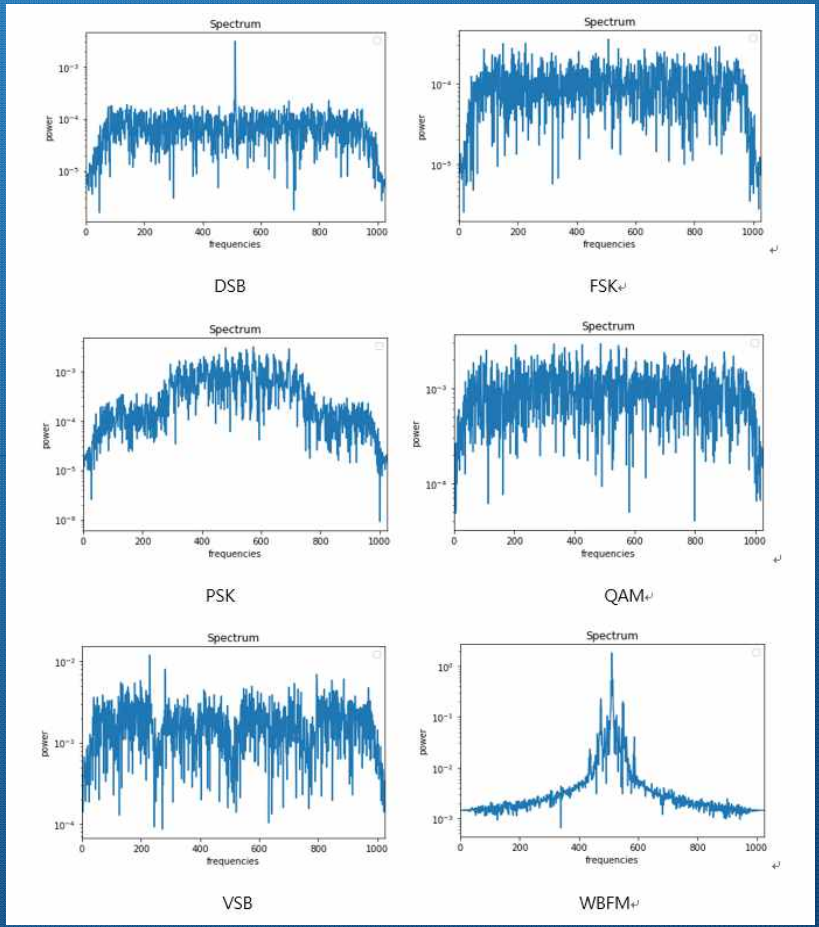


'gamma_max', 'sigma_ap', 'sigma_dp', 'P', 'sigma_aa', 'sigma_af', 'sigma_a', 'mu_a42', 'mu_f42', 'beta', 'v20', 'K', 'S', 'PA', 'PR', 'sigma_v', 'm_a', 'sigma_f', 'sigma_inst_a', 'gamma_maxf', 'gamma_maxa', 'mu_aa', 'v_phs', C20', 'C21', 'C40', 'C41', 'C42', 'C60', 'C61', 'C62', 'C63', 'C80', 'C81', 'C82', 'C83', 'C84', 'C100', 'C101', 'C102'

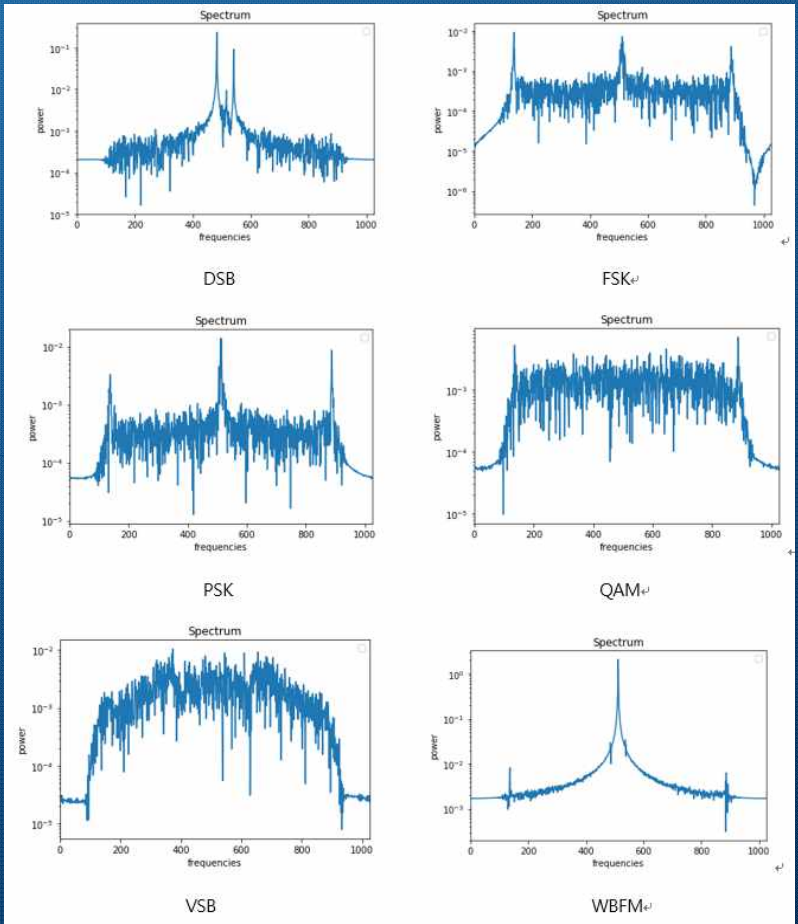
< Work flow and features >

* in-phase and quadrature components, complex signal

Signal shape example



<Original signals>



<Sampled signals>

Learning

```
from CatchIQ_forLearning import mod_n_freq,signal_n_bw,modulation_name
```

Load main parameters

```
feature_file_string=modulation_name
from os import listdir
dataset=None
for mod_type in range(len(feature_file_string)):
    feature_file_saved_dir='../UsingFunction'
    feature_files=[]
    for ffl in listdir(feature_file_saved_dir):
        if ffl.find('forLearning_signal feature_of_real_'+feature_file_string[mod_type])!=-1:
            feature_files.append(ffl)

    from scipy.io import loadmat
    from pandas import DataFrame
    import pandas as pd
    X_data=None
    for ff in range(len(feature_files)):
        _X_data=DataFrame(loadmat(feature_files[ff])['signal_feature'])
        X_data=pd.concat([X_data,_X_data],axis=0,ignore_index=True)

    _y_data=[]
    for Xds in range(0,X_data.shape[0]):
        _y_data.append((modulation_name.index(feature_file_string[mod_type])))
    y_data=pd.Series(_y_data)

    _dataset=[]
    _dataset=pd.concat([X_data,y_data],axis=1,ignore_index=True)
    dataset=pd.concat([dataset,_dataset],axis=0,ignore_index=True)
```

Dataset construction

List up feature-values

```
features=['gamma_max','sigma_ap','sigma_dp','P','sigma_aa','sigma_af','sigma_a','mu_a42','mu_f42','beta','v20',
          'K','S','PA','PR','sigma_v','m_a','sigma_f','sigma_inst_a','gamma_maxf','gamma_maxa','mu_aa','v_phs',
          'c20','c21','c40','c41','c42','c60','c61','c62','c63','c80','c81','c82','c83','c84','c100','c101','c102']
features_name=[]
for fl in range(len(features)):
    features_name.append(features[fl])
features_name.append('modulation type')
dataset.columns=features_name
dataset
```

Prediction

```
import CatchIQ_forPrediction
import ExtractFeatures_forPrediction
import FabricateDataset_forPrediction
```

Load main module

```
from CatchIQ_forPrediction import mod_n_freq,signal_n_bw,modulation_name
from FabricateDataset_forPrediction import X_what
import pickle
learned_model=pickle.load(open('../UsingFunction/_TrainingModel.h5','rb'))
prediction=learned_model.predict(X_what)
```

Load learning model and predict

```
import pandas as pd
from pandas import Series,DataFrame
temp_obj=Series(prediction)
_recog=sorted(temp_obj.unique())

modulation_name=list(sorted(signal_n_bw.keys()))
ln_pair=dict(zip(range(len(modulation_name)),modulation_name))
recog_numbers=DataFrame(temp_obj.value_counts(),index=[list(sorted(ln_pair.keys()))],columns=['# of recognition']).fillna(value=0.0)
for _r in range(len(_recog)):
    recog_numbers=recog_numbers.rename(index={_recog[_r]:ln_pair[_recog[_r]]})
recog_percent=DataFrame(recog_numbers/len(X_what)).fillna(value=0.0);recog_percent.columns=['% of recognition']

f=open('_Score_of_prediction.txt','w')
guess=pd.concat([recog_numbers,recog_percent*100],axis=1)
print(guess,file=f)
f.close();print(guess)
```

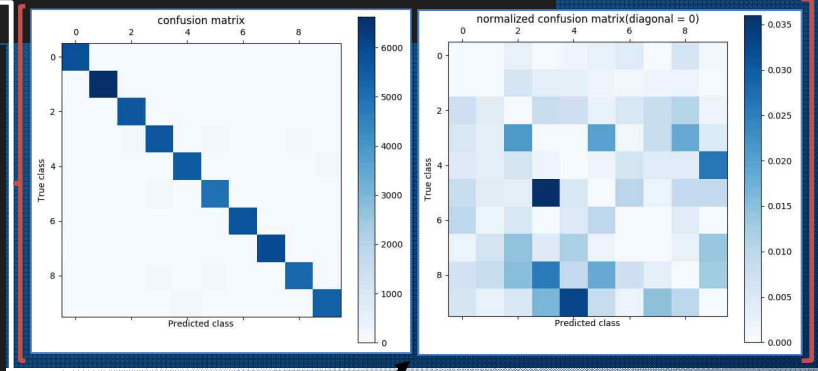
Output format

```
import Figure_forPrediction
```

```
import pickle
learned_model=pickle.load(open('../UsingFunction/_TrainingModel.h5','rb'))
prediction=learned_model.predict(X_what)

from pandas import Series,DataFrame
temp_obj=Series(prediction)
_recog=sorted(temp_obj.unique())
ln_pair=dict(zip(range(len(modulation_name)),modulation_name))
recog_numbers=DataFrame(temp_obj.value_counts(),index=[list(sorted(ln_pair.keys()))],columns=['# of recognition']).f
for _r in range(len(_recog)):
    recog_numbers=recog_numbers.rename(index={_recog[_r]:ln_pair[_recog[_r]]})
recog_percent=DataFrame(recog_numbers/len(X_what)).fillna(value=0.0);recog_percent.columns=['% of recognition']

expect=pd.concat([recog_numbers,recog_percent*100],axis=1)
heatmap_num[mn]=expect['# of recognition']
heatmap_per[mn]=expect['% of recognition']
```



visualization

I Introduction to AI

II Modulation Classification by AI

III AI Application on SM

Current status of SM technology development

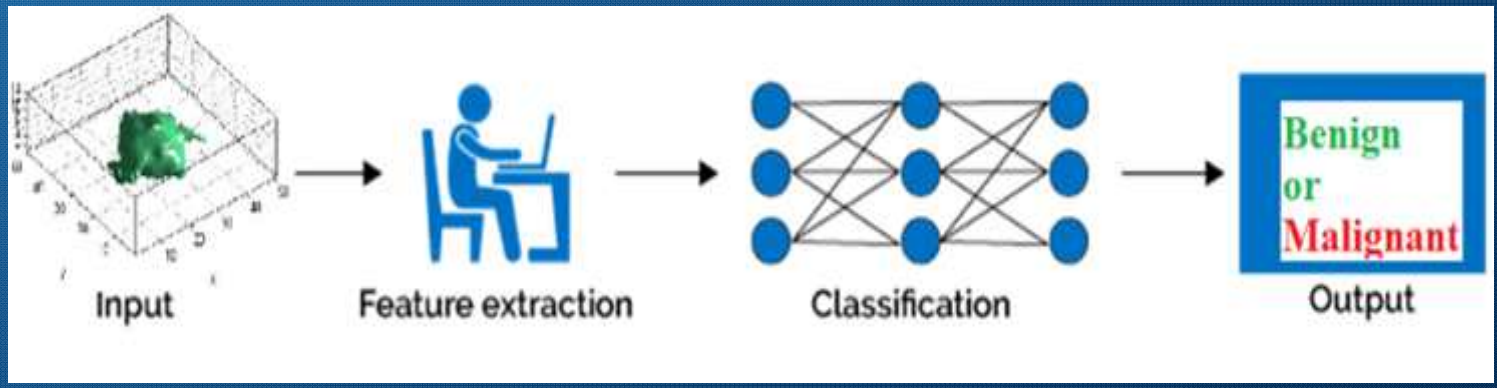
- **Monitoring unlicensed frequency to prevent interference**
- **Evolving platform**
  - ✓ Handy -> Drone -> Sensor
- **Being apply AI technique**
- **Considering 5G frequency (28 GHz) monitoring**
- **Reducing size of SM system**
  - ✓ Fixed -> Mobile -> Handy

< Fixed SM system >

< Mobile SM system >

Advantages of AI based SM technology

- **Automation**
  - ✓ Classification of the signals collected from in real time through various platforms such as fixed stations and mobile stations, drones, sensor, etc.
- **Accuracy**
  - ✓ Better classification rate than manual classification based on the experience of workers
  - ✓ Block the possibility of errors due to manual classification
- **Maximum SM effects**
  - ✓ Reducing data acquisition time
  - ✓ Real-time SM and analysis



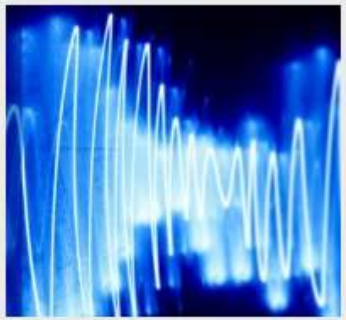
< http://semiengineering.com >

Standard of AI based SM technology

ITU's Areas and Actions*: Future plan

Telecommunications/ICTs are a key enabler to achieve the Sustainable Development Goals (SDGs) and to build a world where social, economic, environmental and technological development is sustainable and available for everyone, everywhere. The fourth industrial revolution, digital economy and society, Internet of Things, connected cars and cities all depend on telecommunication networks, services and applications, and increasingly rely on radiocommunications to provide the basis for ubiquitous connectivity. The ITU Radiocommunication Sector plays a vital role in this ecosystem: managing frequency spectrum and satellite orbits, as well as developing globally harmonized regulations and standards, are fundamental in order to ensure accessible and affordable telecommunications to all. In doing so, artificial intelligence acts as an enabler to enhance emerging radio technologies.

AI AND SPECTRUM MONITORING



Spectrum monitoring has always been the eyes and ears of spectrum management processes to facilitate planning, maximize efficiency, minimize interference and eliminate unauthorized and improper use of the spectrum. AI may be the next computer-aided techniques enhancing the automation of spectrum monitoring tasks which become more complex with the development of new radio technologies. As such, AI may bring new solutions and opportunities for instance for the signal recognition, the real-time monitoring of multiples auto-signaling equipment and devices, and/or the identification of sources of interference.

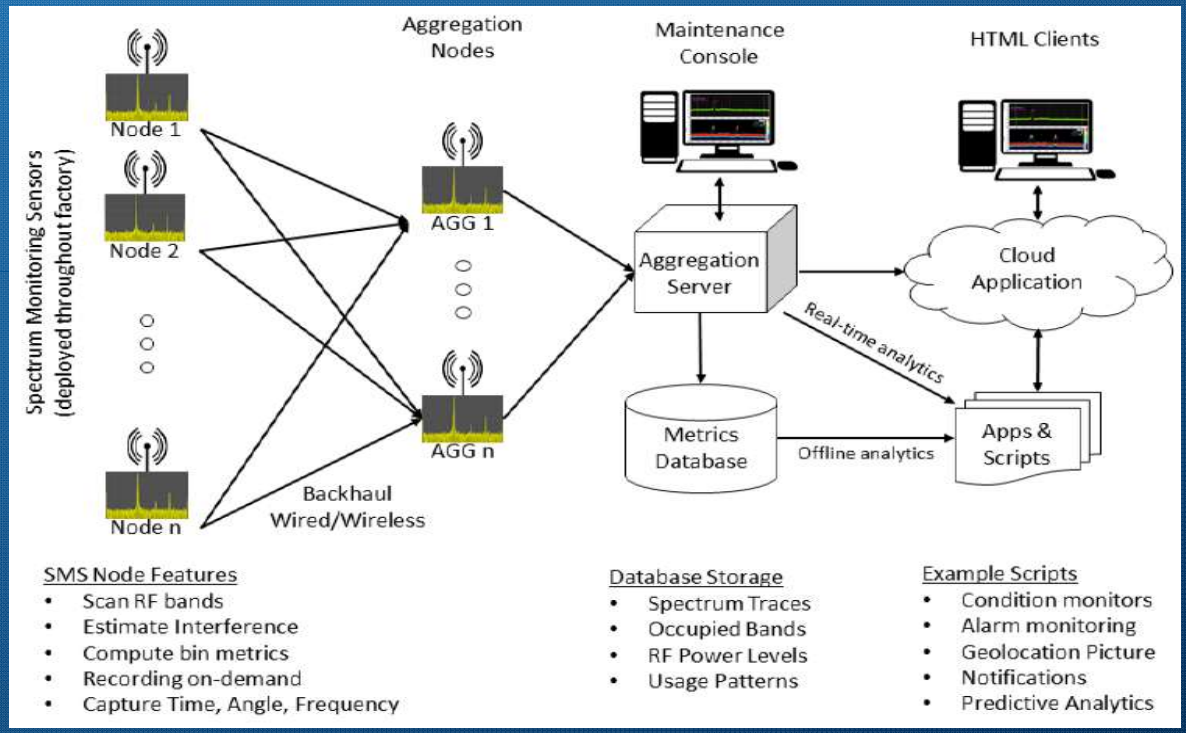
Signal identification, Interference, Real time SM

< AI application in SM >

* https://www.itu.int/en/action/ai/emerging-radio-technologies/Pages/default.aspx

Future view of AI based SM technology

- **Cloud-based signal analysis for efficient data collection and real-time processing**
  - ✓ Utilizing the database in central server
    - Based on various real-world data,
    - Prediction error is minimized, and bias of past data is excluded

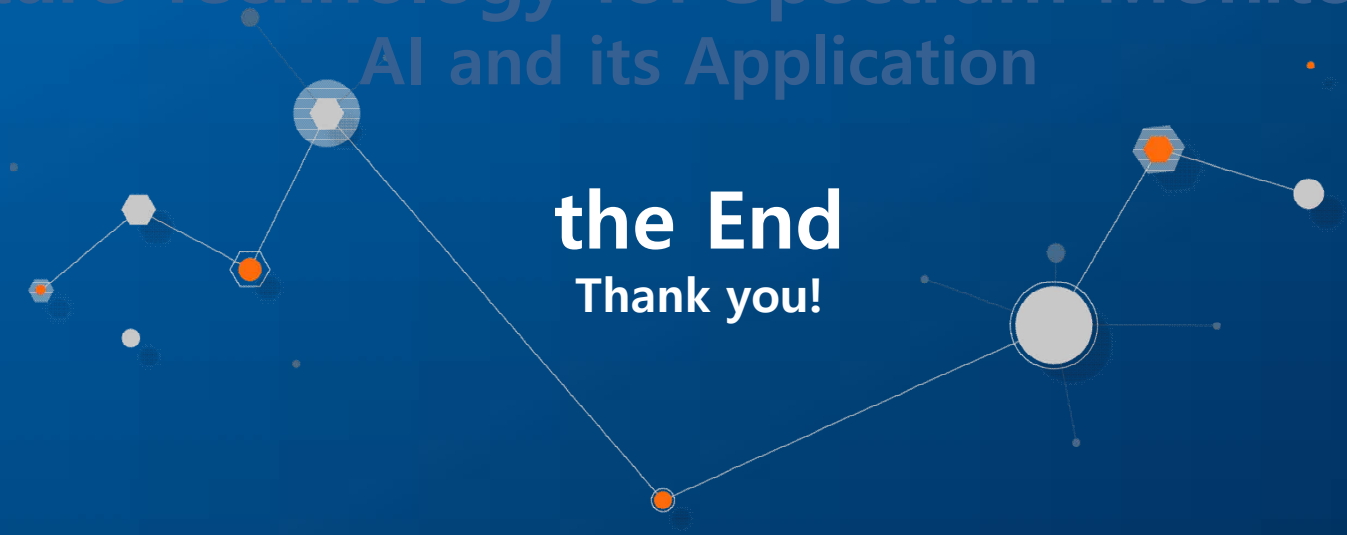


< Future SM and Processing Architecture* >

* NIST Report, "Requirements for Spectrum Monitoring in Industrial Environments," Nov. 2017.

Future Technology for Spectrum Monitoring

AI and its Application



the End

Thank you!