Research Portfolio

Korea University

School of Electrical Engineering

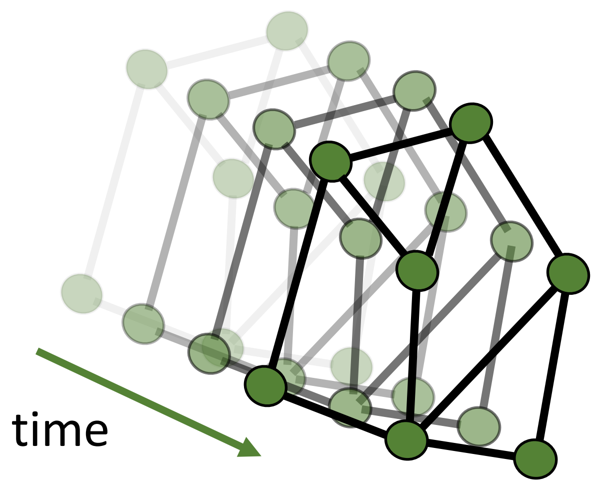
(M.S.) Wonyong Chung, 정원용

CONTENTS

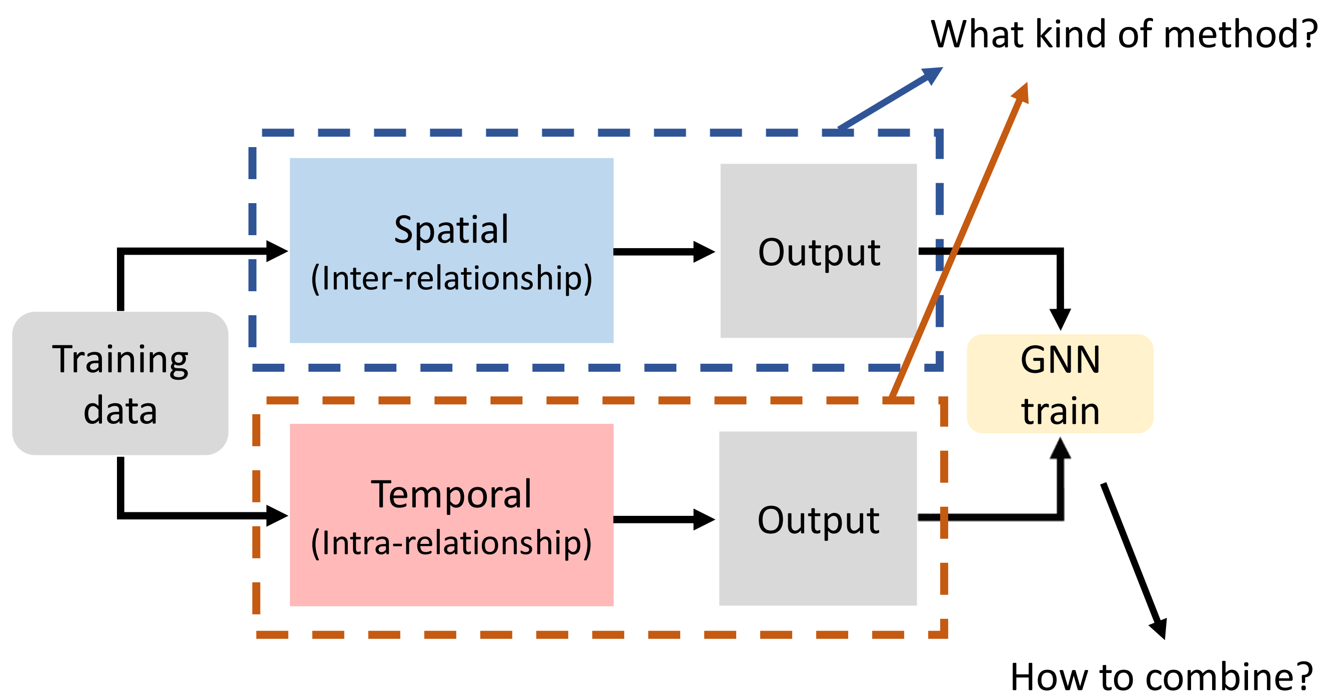
I. Multivariate time series Predicting based on Graph Neural Network

I. Multivariate time series Predicting based on Graph Neural Network

- Predicting the information of multiple nodes configured with time series data.
  - Traffic, Finance, Infectious disease, Electrical system, etc.
- Reflect information from neighboring nodes is important.
  - Existing studies in various ways.
- How to extract the temporal feature of each node data and apply it to the GNN?

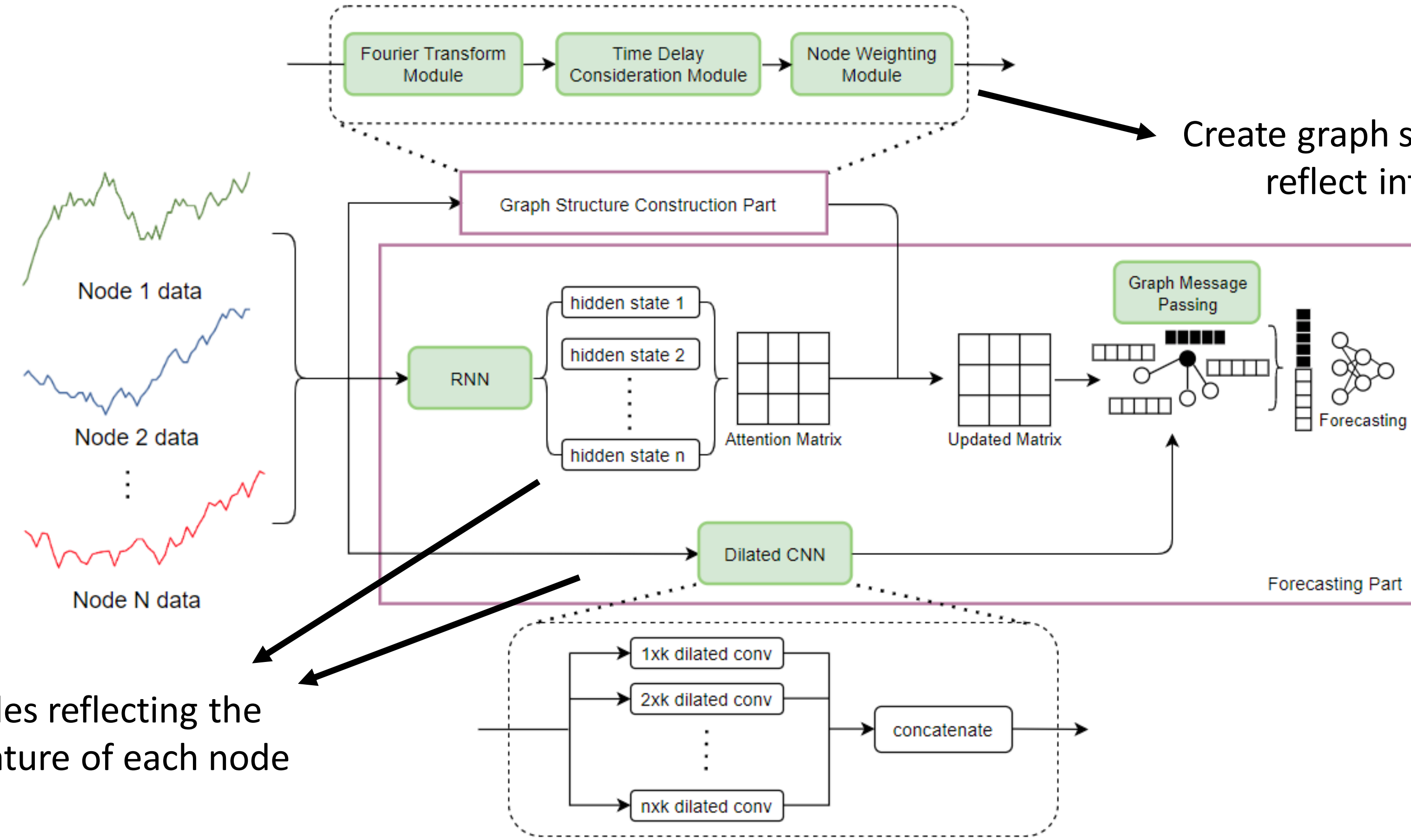


Multivariate time series & GNN



I. Multivariate time series Predicting based on Graph Neural Network

- Propose a GNN based predicting model that combines spatial and temporal information.
  - It is rare to have a pre-defined graph structure. → Graph Structure Construction Part.
  - Predictive learning with the generated graph structure.

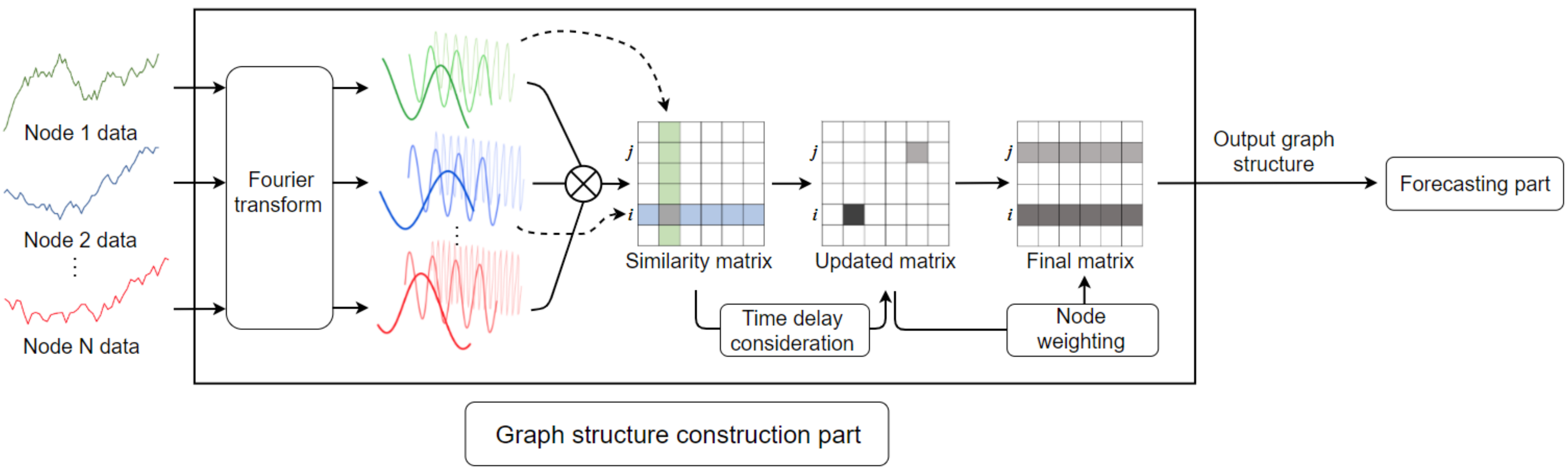


Create graph structure to efficiently reflect inter-relationships

Create nodes reflecting the temporal feature of each node

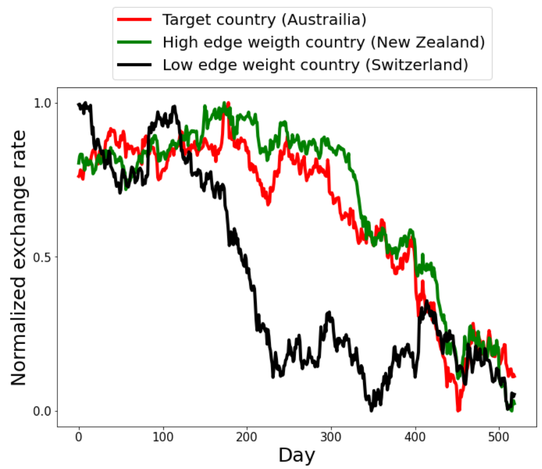
I. Multivariate time series Predicting based on Graph Neural Network

- Generate graph structures only by spectral properties of time series data.
  - Fourier Transform can solve this problem.
  - Remove noise (high frequency spectrum) and extract only low frequency spectrum.
    - Calculate up to $\lfloor \lfloor (\boldsymbol{T}-\mathbf{1})/\mathbf{2} \rfloor * \mathbf{0.2} \rfloor$ frequency when input sequence length $\boldsymbol{T}$.
  - Create a base graph structure by inner producting the amplitude of the two decomposed spectra.
  - The time series similarity in the spectral domain is well reflected in the graph structure.



I. Multivariate time series Predicting based on Graph Neural Network

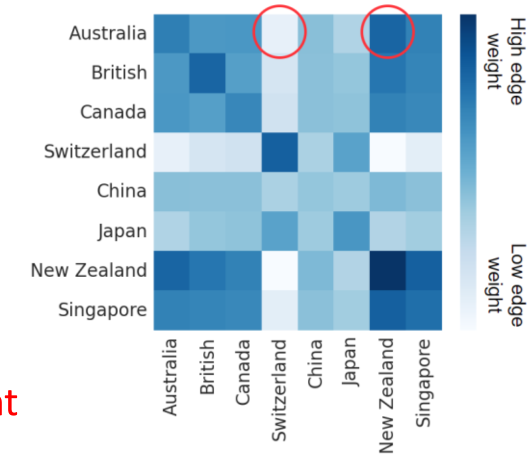
- Spectra based graph structure fits well into real world data.
  - Exchange rate data, US Influenza outbreak data

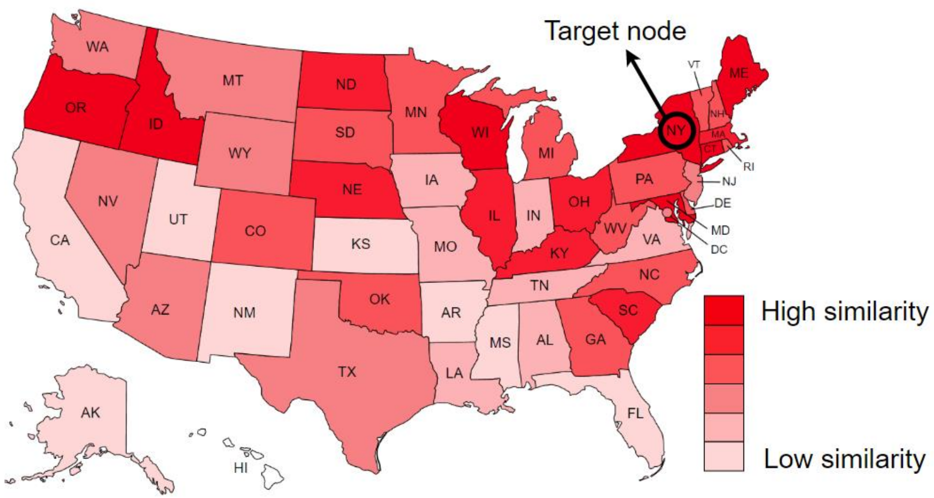


Australia-Switzerland : ↓ edge

Australia-New Zealand : ↑ edge

Lots of information movement





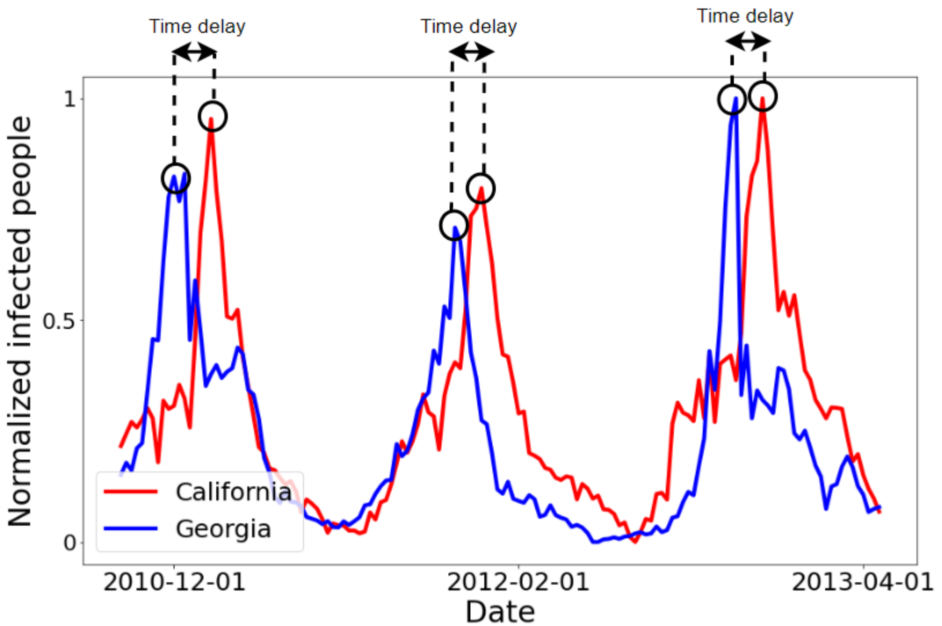
Movement of many infected people

Close to the target node (New York) : ↑ edge

Climate similar to the target node (New York) : ↑ edge

I. Multivariate time series Predicting based on Graph Neural Network

- Update graph structure considering time delay.
  - e.g.) Altcoins follow the trend of bitcoin, Infectious diseases spread from occurred area.
  - The "following data" follows the trend of the "preceding data". → Graph structure is updated by considering both horizon and time delay.
  - The weight of the edge from the preceding data to the following data is higher.

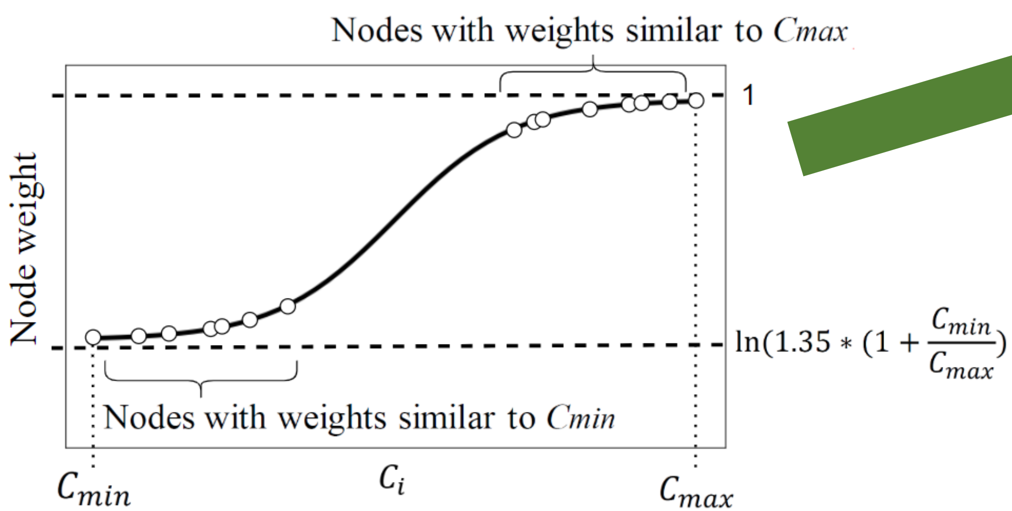


Algorithm 1: Time delay consideration

Input: X_{seq}, G, input length T, moving length m, horizon h
Output: updated G
for each node i of X_{seq} **do**
    for each neighbor j of node i **do**
        for *each starting point of* X_j **do**
            Target data ← X_j [*starting point : starting point* + T]
            Distance ← DTW(Target data, X_i)
            if *Distance* < *min Distance* **do**
                min Distance, min starting point ← *Distance, starting point*
            starting point ← *starting point* + m
        if *min starting point* < *fixed data starting point* **do**
            preceding data, following data ← X_j, X_i
        else do
            preceding data, following data ← X_i, X_j
        time delay ← | X_i *starting point* – *min starting point* |
        G_{ij} ← G_{ij} / | time delay – h + 1 |
        G_{ji} ← G_{ji} / time delay

I. Multivariate time series Predicting based on Graph Neural Network

- Weight nodes based on the amount of information moved. → Nodes with many connections to other nodes.
  - e.g.) Since the number of passengers traveling through hub airports is so large, we increase the influence of hub airports by giving them a large weight.
  - Weighted by the following formula. → Each node's total connection information is taken as a log function defined by C_i.



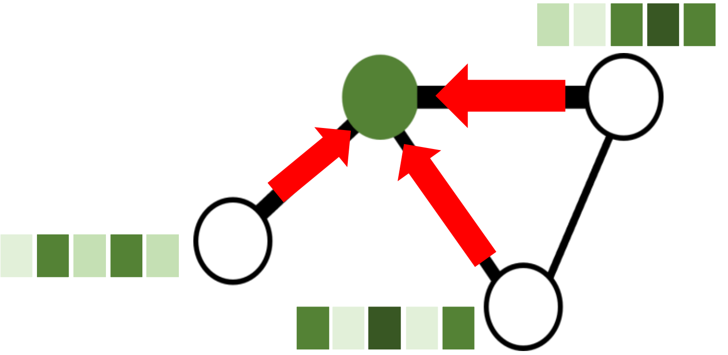
$$f(C_i) = \left(\frac{1}{1 + e^{-a*(C_i - b)}}\right) * c + d$$

$a = \left(\frac{10}{Cmax - Cmin}\right)$	The part keeps the shape of *f*. The numerator value makes a function suitable for the hub node weighting task.
$b = \frac{Cmin + Cmax}{2}$	The part moves *f* along the C_i axis so that the center of the *f* is located at average value of *Cmin* and *Cmax*.
$c = 1 - \ln\left(1.35 * \left(1 + \frac{Cmin}{Cmax}\right)\right)$	The part limits the range of *f* result values.
$d = \ln\left(1.35 * \left(1 + \frac{Cmin}{Cmax}\right)\right)$	The part moves f to the node weight axis so that the result value of *Cmax* converges to 1. If the numerical difference between *Cmin* and *Cmax* is small, $f(Cmin) \approx \ln(2.70) \approx 1.00$.

I. Multivariate time series Predicting based on Graph Neural Network

- The Forecasting part uses graph convolution and MLP.
- The information is combined by <u>graph convolution</u> and <u>concatenated with each node vector</u>.
  - What if graph convolution is replaced by GraphSAGE, Node2vec, etc.?
  - What if the combination part is replaced by mean or average instead of concatenate?

Dataset	Sample rate	Period	# of time-stamps	# of nodes	Total Statistics		
					Average	Max	Standard deviation
ILI US-States	1 week	2010 - 2017	360	49	355.85	11,452	742.89
ILI US-Regions	1 week	2002 - 2017	785	10	1,072.78	30,282	1,949.17
ILI Japan-Prefectures	1 week	2012 - 2019	348	47	615.04	26,635	1620.84
Exchange rate	1 day	1990 -2016	7588	8	0.6947	2.1090	0.4761
US stock market price	1 day	2007-2016	2518	50	103.75	4,984.99	302.97



- Uses five datasets and evaluates with RMSE, PCC.

I. Multivariate time series Predicting based on Graph Neural Network

- Our model showed superior performance over comparative models in almost all horizons of datasets.
  - Relatively higher performance in long-term (horizon = 10, 15) predictions.
  - Relatively higher performance in datasets with many nodes than datasets with fewer nodes.
  - Statistics-based prediction models show significantly lower performance.

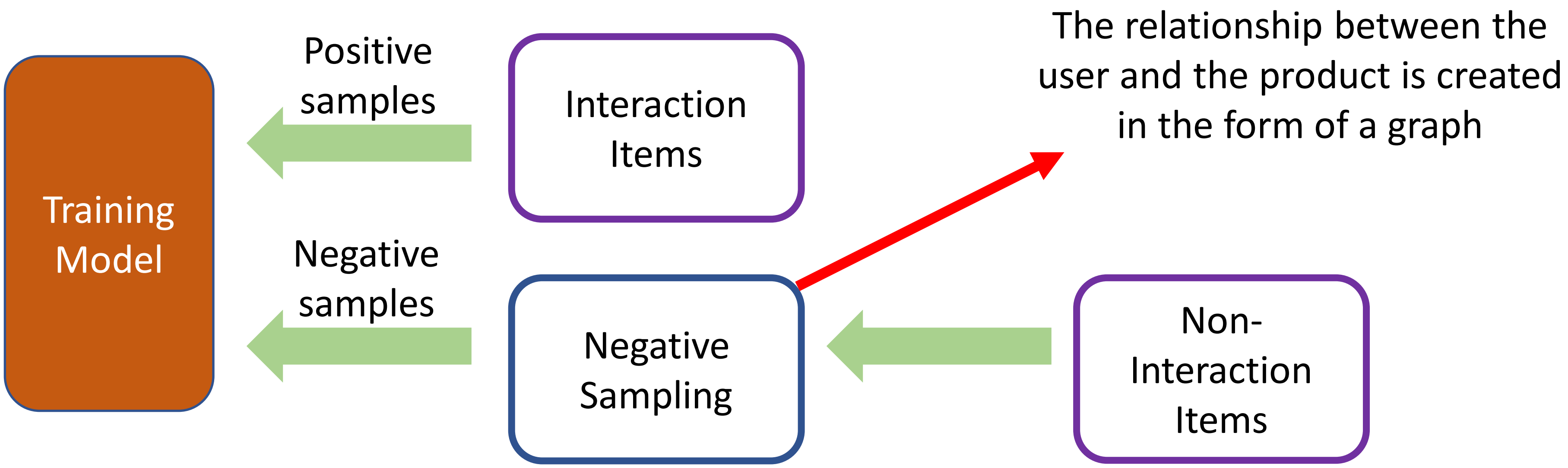
TABLE 3. Experimental results on the ILI US-Regions dataset

Horizon	2		3		5		10		15	
Metric	RMSE	PCC	RMSE	PCC	RMSE	PCC	RMSE	PCC	RMSE	PCC
AR	570	0.927	757	0.878	997	0.792	1330	0.612	1404	0.527
ARMA	560	0.927	742	0.876	989	0.792	1322	0.614	1400	0.520
MLP	524	0.931	701	0.869	974	0.803	1312	0.608	1409	0.531
RNN	513	0.940	689	0.895	896	0.821	1328	0.587	1434	0.499
LSTM	507	0.943	688	0.895	975	0.812	1351	0.586	1477	0.488
LSTNet-skip	554	0.935	801	0.868	998	0.746	1157	0.609	1231	0.533
ST-GCN	697	0.879	807	0.840	1038	0.741	1290	0.644	1286	0.619
Cola-GNN	480	0.940	636	0.909	855	0.835	1134	0.717	1203	0.639
Our model	**478**	**0.946**	**634**	**0.922**	**726**	**0.890**	**932**	**0.815**	**1101**	**0.755**

II. Application of Negative Sampling to Graph Representation Learning

II. Application of Negative Sampling to Graph Representation Learning

- Multiple ways to embed graph nodes. → PinSAGE, GraphSAGE, Node2Vec, etc.
- Negative sampling is used to improve performance in natural language processing (NLP).
  - To include irrelevant words in model training.
  - How can we apply negative sampling to GNN?
- How to perform negative sampling on the recommendation system?

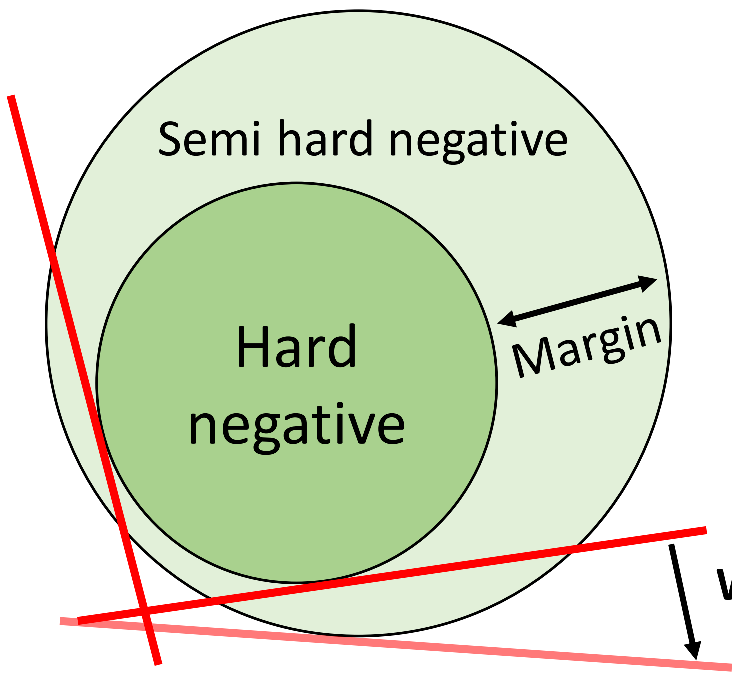


II. Application of Negative Sampling to Graph Representation Learning

- Variety negative sample extraction techniques in graph presentation learning.
  - PinSAGE extracts negative samples through PageRank score. → Hard negative sampling.

Exclude samples that have non-relationship with positive samples. Hard-to-distinguish negative sample.

Easy negative



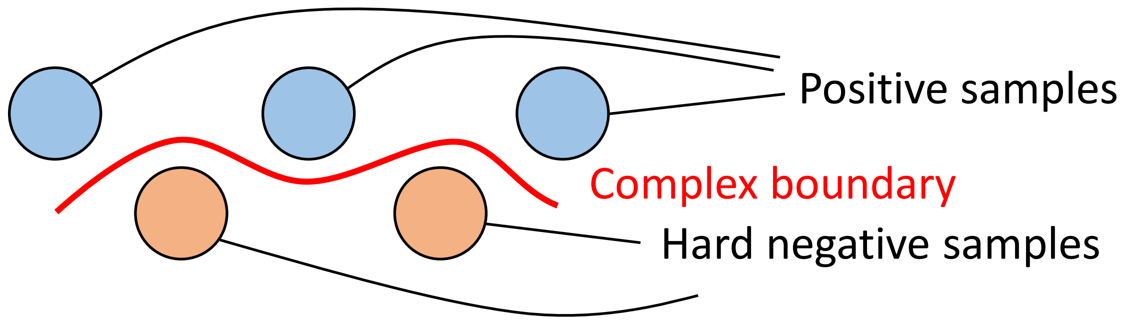
A typical hard negative sampling is not much different from a positive sampling.

→ **“Different but similar”**

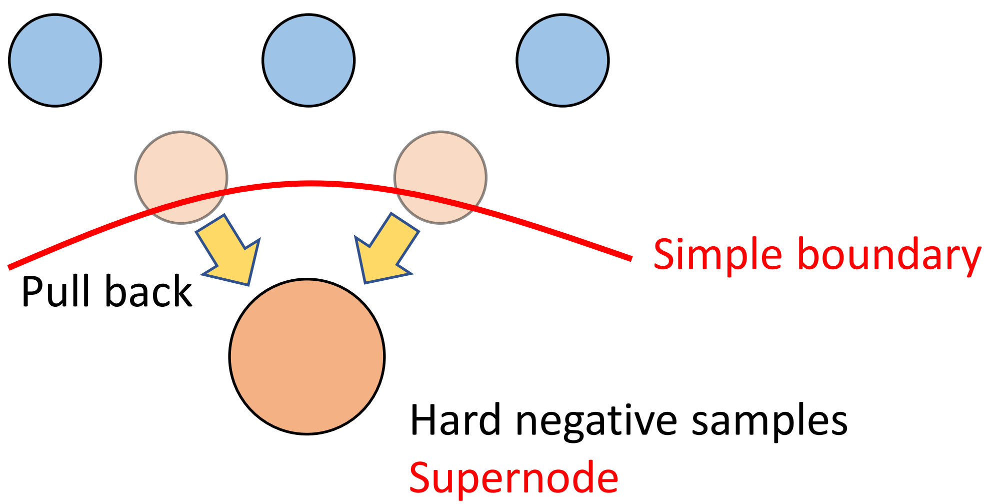
What I focus on: Get a little out of the hard negative

II. Application of Negative Sampling to Graph Representation Learning

- Overcoming the disadvantages of excessive hard negatives.
  - Complex boundary between positive and hard negative samples.



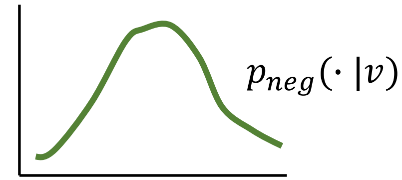
Complex boundary causes the <u>False negative</u> or <u>False positive.</u>

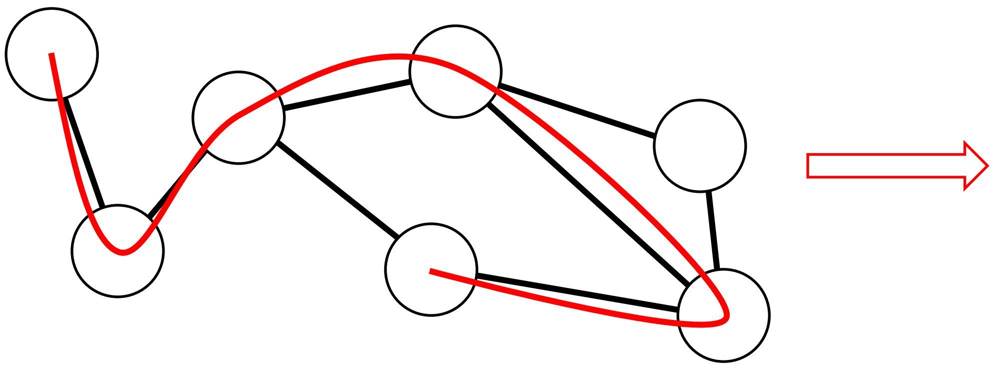


Configure super nodes by grouping hard negative nodes with message passing.
<u>→ Super nodes pull negative samples back</u>

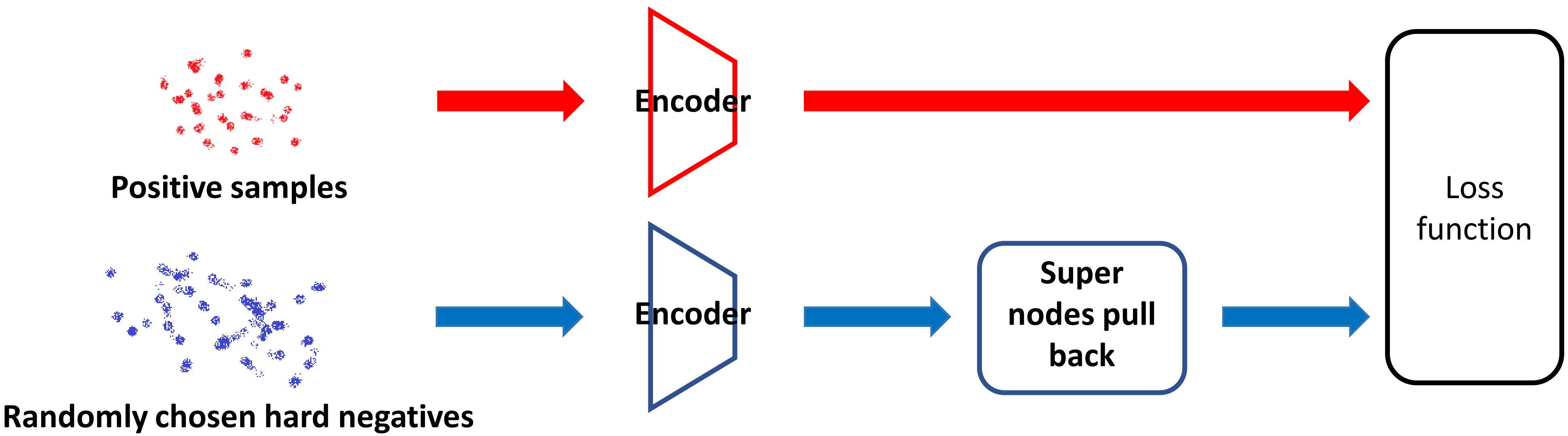
II. Application of Negative Sampling to Graph Representation Learning

- Where should a super node be created?
  - Which message passing method should I use?



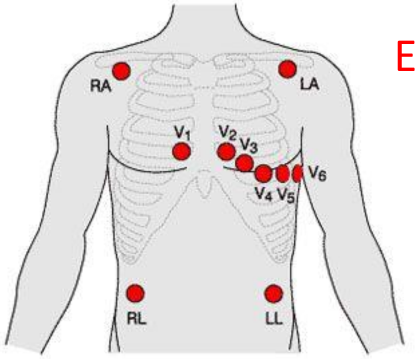


(MCMC algorithm is also available)
Sampling random walks →
Training skip-gram model →
Computing embeddings



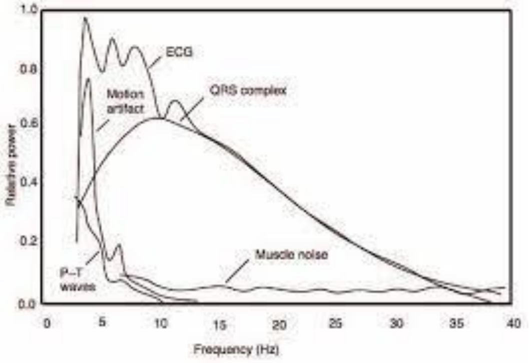
III. Deep Learning / Machine Learning based Projects (1)

- Deep Learning-based ECG signal classification
- How to classify heart disease through ECG signal?
  - The most important consideration in ECG signal data → Too much noise
  - e.g.) QRS, Motion artifact, Muscle noise, etc. → Should be eliminated.
  - Fourier transform decomposes into spectral domains, eliminates frequencies corresponding to noise
  - Only the signals in the remaining frequency range are combined.



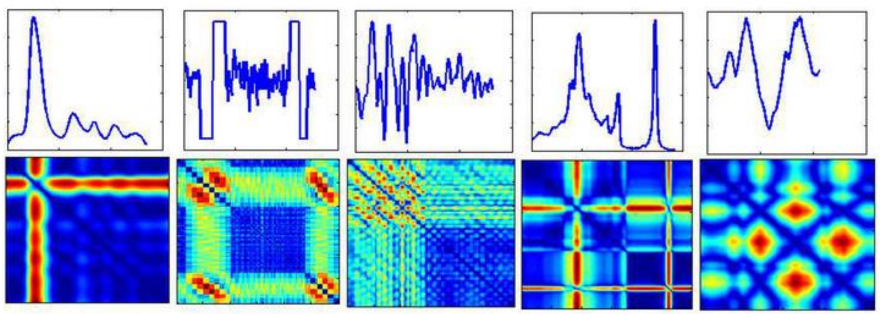
Electrical signal at each measurement point

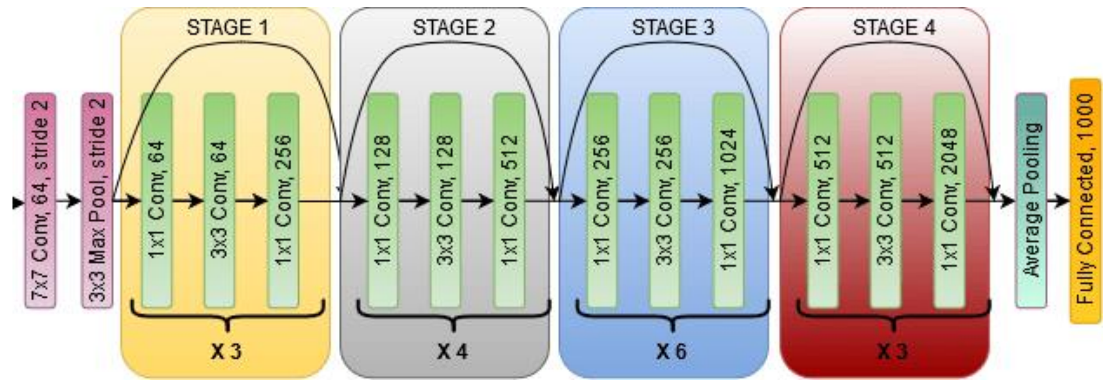
Eliminate unnecessary frequencies



III. Deep Learning / Machine Learning based Projects (1)

- Imaging the noise-cancelled electrical signals. → To pass to CNN layer.
  - Electrical signals converted into images pass through Resnet-50.





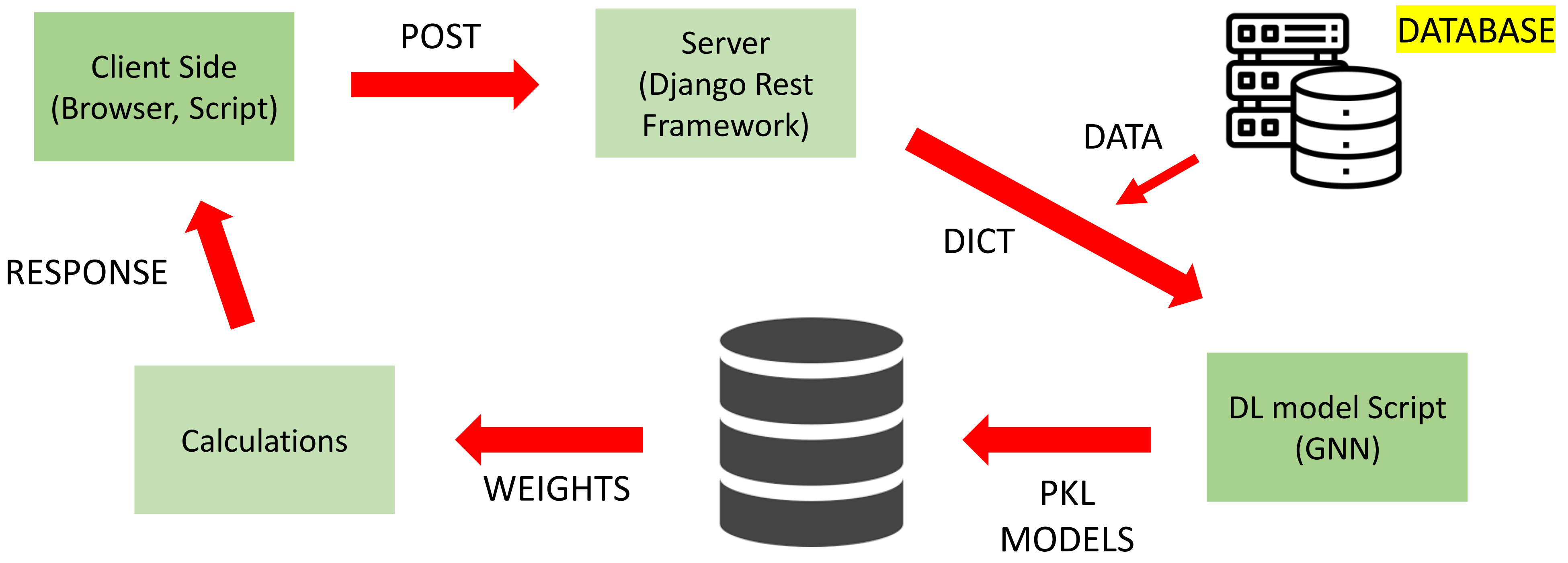
- Applied various CNN models. → Residual block is required.
- Overfitting must be prevented
- Data class <u>imbalance problem</u> in medical data. → F1-score was used as an evaluation metric.

  Data resampling techniques?

III. Deep Learning / Machine Learning based Projects (2)

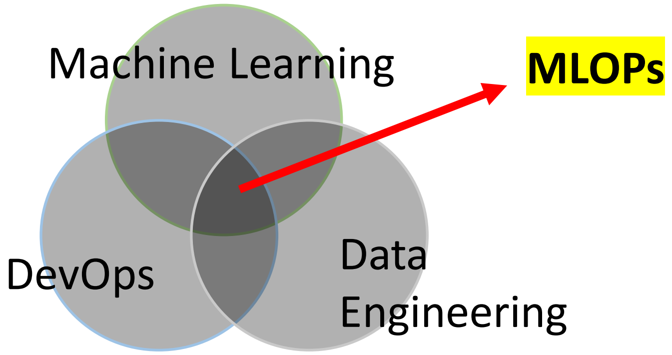
III. Deep Learning / Machine Learning based Projects (2)

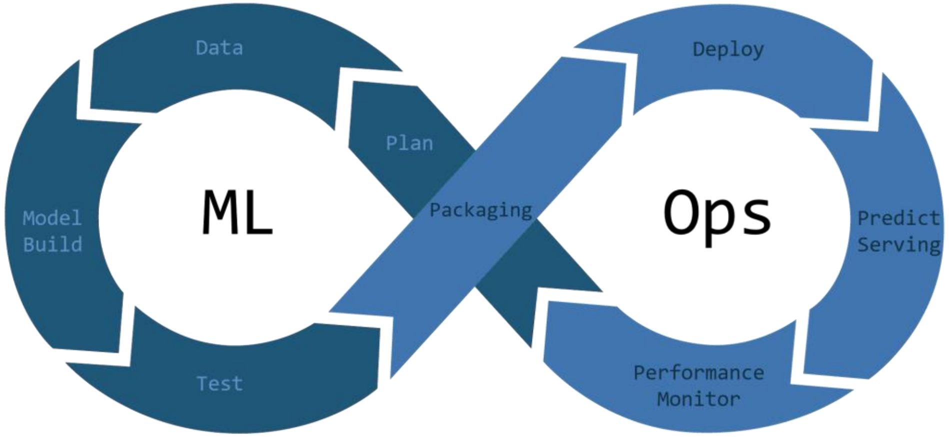
- GNN-Based Stock Price Forecasting and Web Service Implementation.
  - Implement the stock price forecasting web service through the Django framework.
  - Deep learning required for stock prediction is implemented with GNN.
  - When the user enters the desired company (stock) and the forecast point, the forecast is shown.



III. Deep Learning / Machine Learning based Projects (2)

- About MLOPs (Machine Learning Operations).
  - Not just research deep learning models.
  - Data preprocessing → model training → database management → model storage → implementation as web services.
  - Be familiar with the whole series of courses.
  - Research and analyze the optimal course. → ML Design Patterns ?
- Store the list of major corporations and stock prices by date in the database.
  - Retrieve them from the database whenever necessary.
  - Building a database is more efficient.

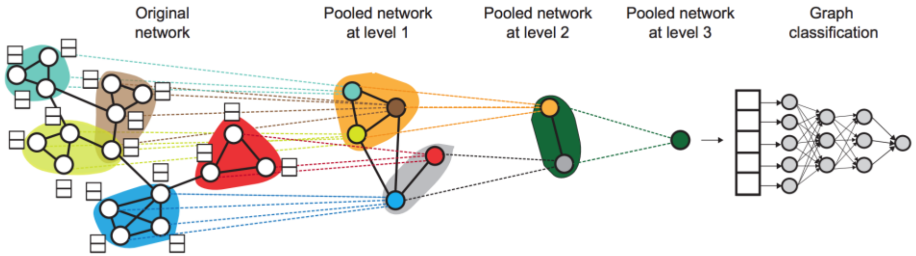




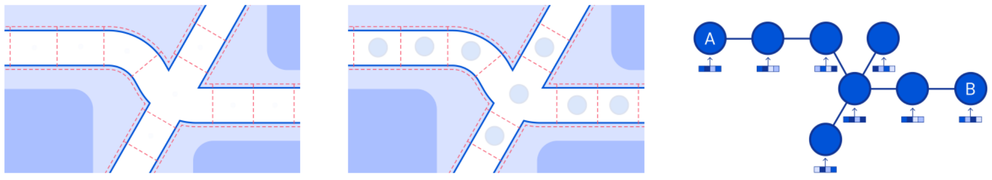
IV. Future Research

IV. Future Research

- Time series prediction for large graph datasets.
  - Reduces graph size with Graph pooling. → Graph U-net, Diffpool, Eigenpool, etc.
  - Usually used for tasks such as node classification, graph classification.
  - How to apply graph pooling to time series prediction? → Google map traffic prediction



Diffpool overview



Google map

- Plan to study times series prediction by referring to the idea of existing graph pooling techniques.
  - Gradually reduce the graph by grouping several nodes into supercells

E.O.D.

Korea University

School of Electrical Engineering

(M.S.) Wonyong Chung, 정원용