

한국교통연구원(KOTI)

딥러닝 기반 도심지 교통혼잡 예측 및 신호제어 솔루션 시스템 개발

2020.10.07 (수)

-Deep Embedded
Clustering(DEC)을 통한
교통 패턴 분류 방법론

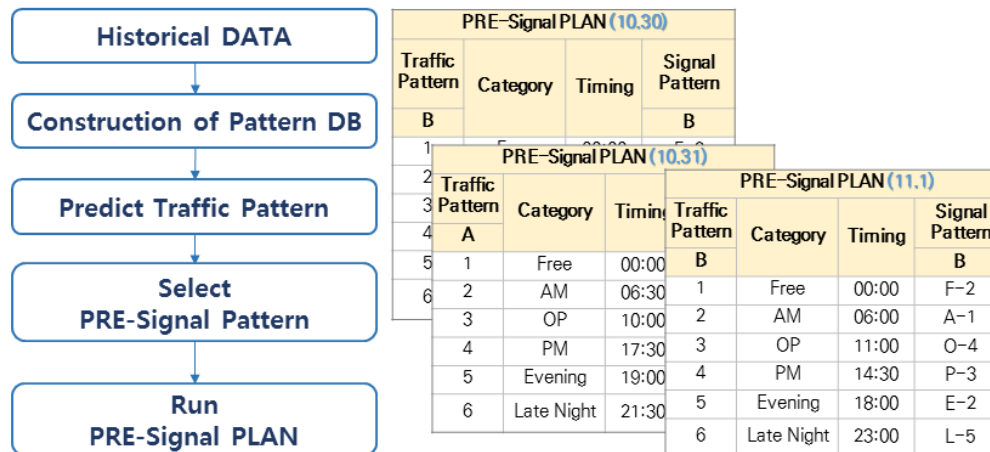


신호최적화를 위한 시뮬레이션 구축

데이터 기반 소통상황 분류 고도화(3차년도)

● 혼잡교차로 신호제어 솔루션 개발을 위한 교통패턴 분류

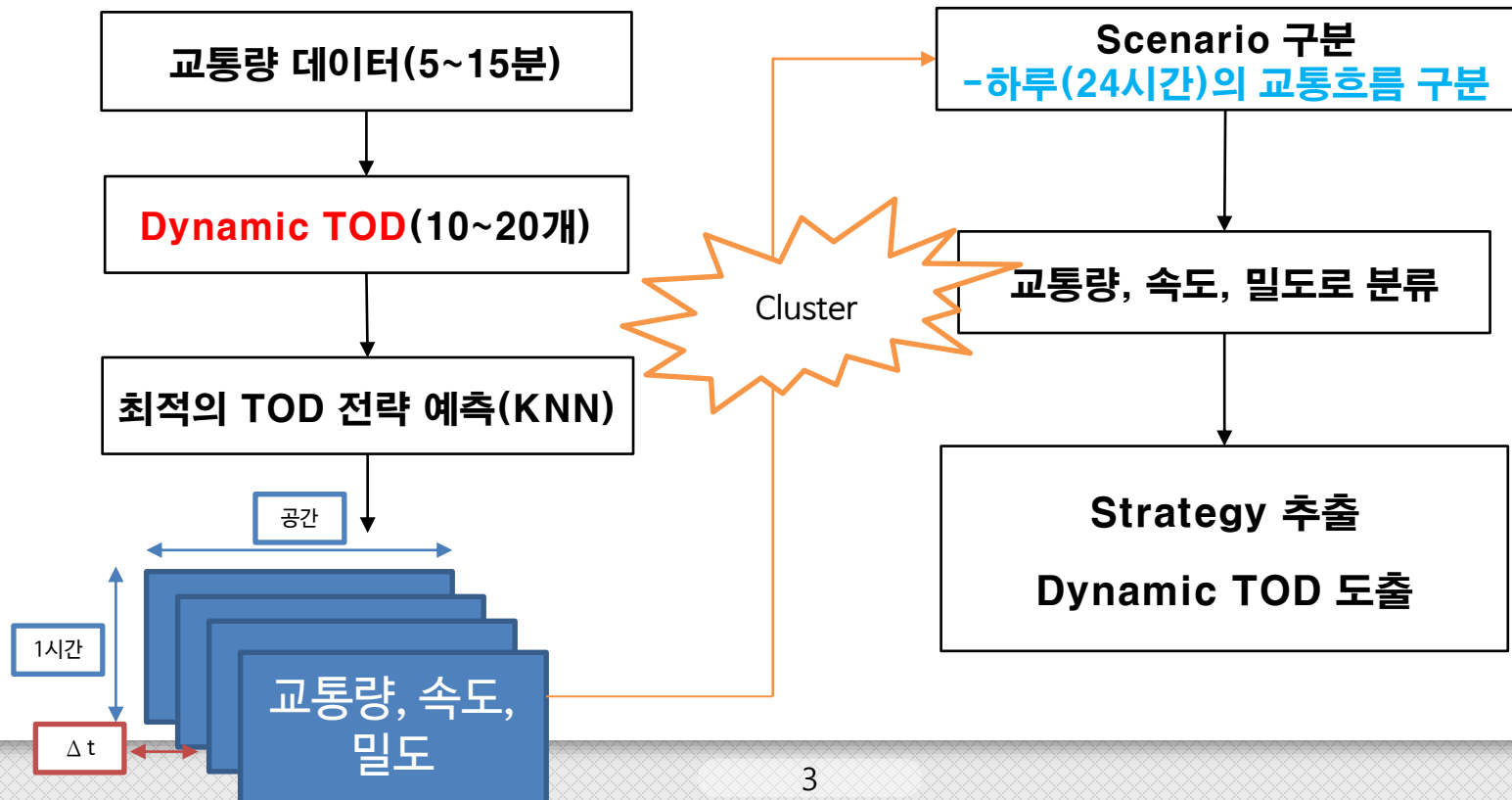
- DSRC 속도 데이터를 활용한 교통패턴 추정
- 교통 데이터를 활용하여 교통 패턴을 분류(레이블링)하고자 함
 - 요일별 분류 / 시간별 분류 / 교통량에 따른 분류
- 교통패턴별 시간대별 교통신호제어전략 설계
- 분류된 교통패턴(시나리오)별 최적교통신호 대안 library 구축



TOD 수립

— 현재 신호제어 전략과 Dynamic TOD 설정 방법

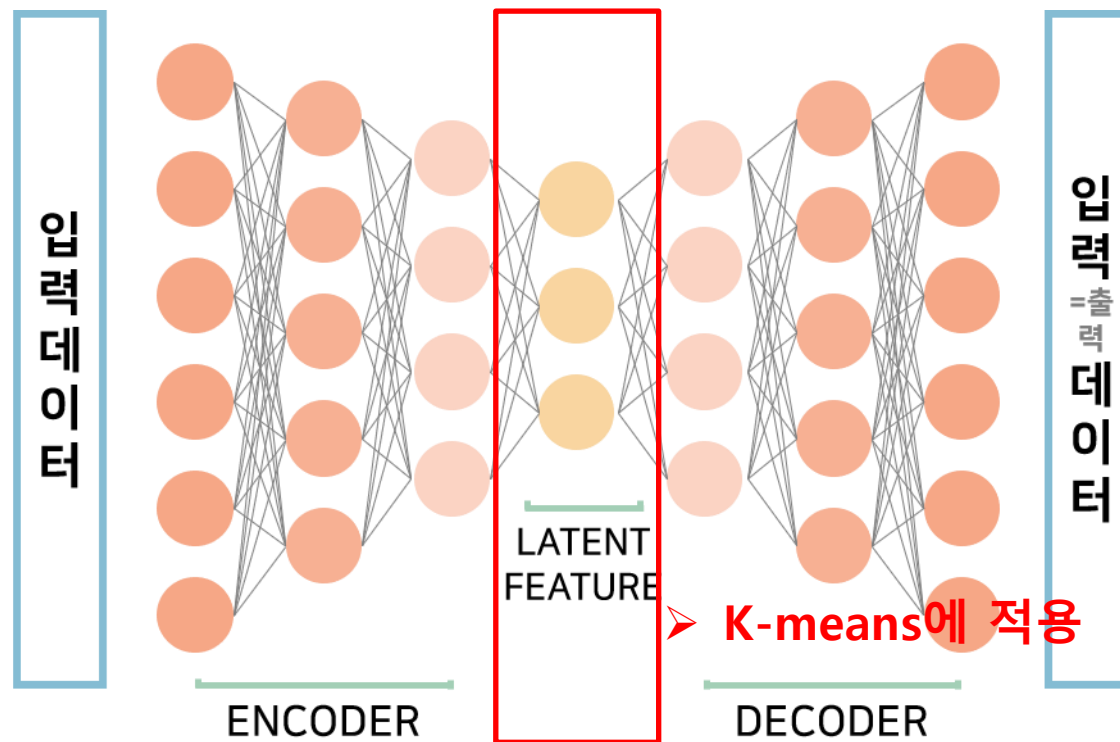
- 현재 신호제어 전략(TOD) : 시간대별 교통량 데이터를 기준으로 TOD를 분류
 - 전문가의 판단에 의해 TOD를 주말/주중의 10~20개(군집화 개수) 시간대로 나눔



Auto encoder를 통한 교통패턴분류 방법론

Auto encoder 적용 방법론 (올해 초 계획)

- 링크별 통행속도에 따른 교통패턴 분류



Auto encoder 방법론

● Auto encoder?

Latent Space란, 특정 차원에서 원하는 정보들이 모여있는 공간이다.

유명한 Mnist의 예시는 아래 그림과 같다.

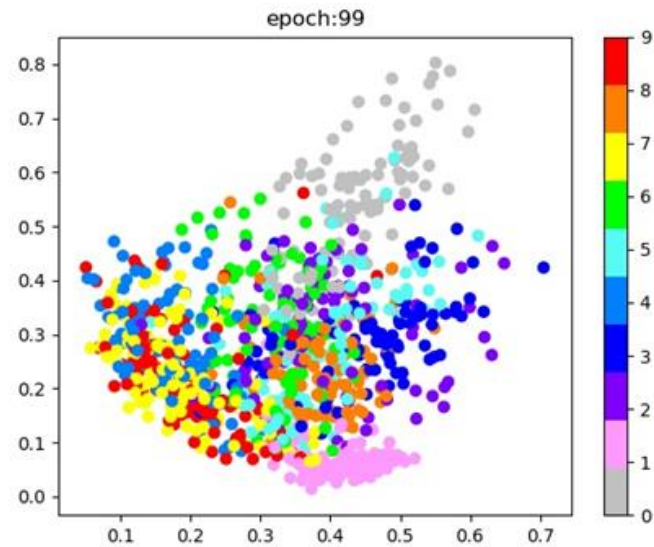


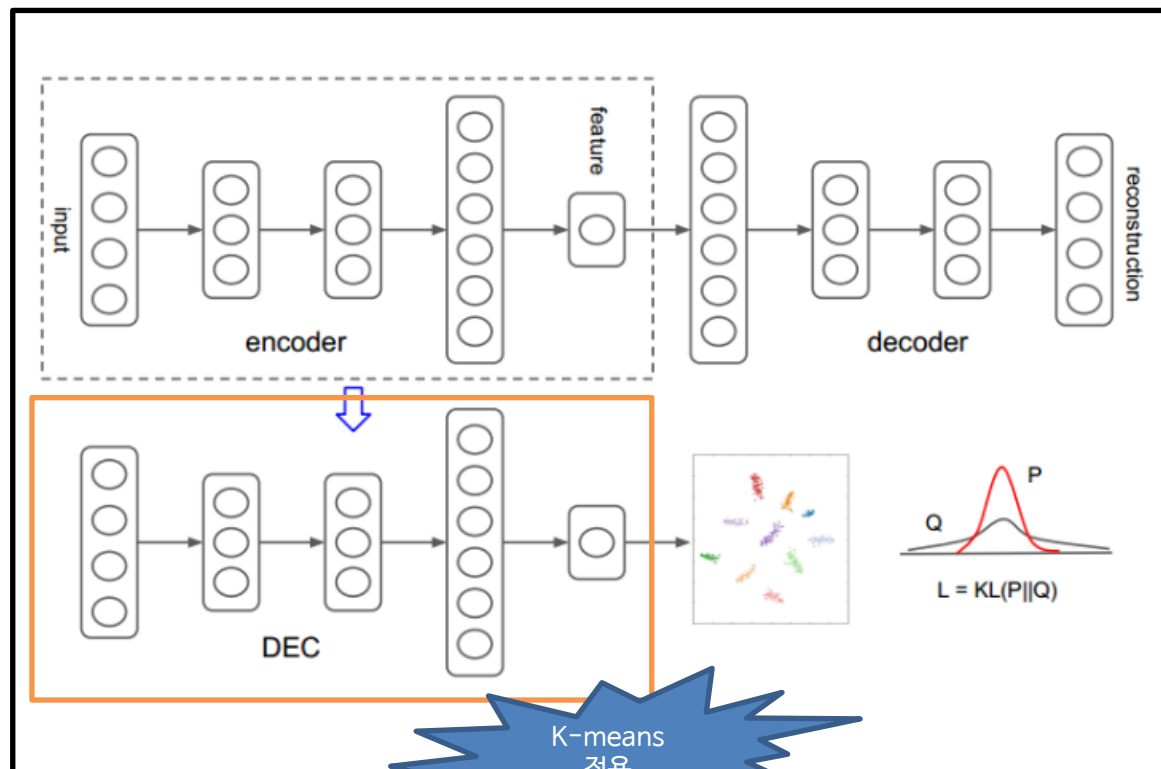
그림4. AE로 구한 Mnist 데이터의 latent space

Auto encoder를 통한 교통패턴분류 방법론

Auto encoder 적용 방법론 보완(1/2)

— Unsupervised Deep Embedding for Clustering(DEC) Analysis 단계

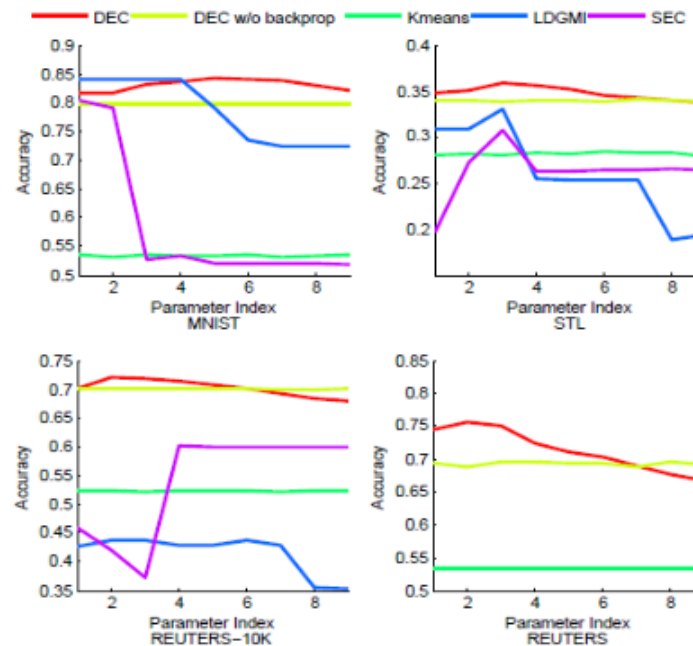
- Clustering a set of n-dimensional points $\{x_i \in X\}_{i=1}^n$ into k clusters, each represented by a centroid $u_j, j = 1, \dots, k$
- $f_\theta : X \rightarrow Z$ 에서 θ : learnable parameters, Z: latent feature space



K-means
적용

DEC 방법론

- Encoding된 값이 일정한 분포를 가질 것으로 가정하고 여러 번의 simulation과 iteration을 통해 최적의 군집화를 도출하는 방법론
- Deep embedding과 Clustering을 결합한 방식으로 Soft assignment를 이용한 새로운 방식의 Iteration을 적용



Auto encoder 적용 방법론 보완(2/2)

—● Unsupervised Deep Embedding for Clustering Analysis 단계

- DEC 방법론: ① parameter initialization with a deep autoencoder ② parameter optimization (i.e., clustering), where we iterate between computing an auxiliary target distribution and minimizing the Kullback–Leibler (KL) divergence
- 아래 i)과 ii) 단계를 반복하여 수렴 기준이 충족될때까지 반복
- i) compute a **soft assignment**(q_{ij} = sample i 가 cluster j 에 속할 확률) between the embedded points and the cluster centroids

$$q_{ij} = \frac{(1 + \|z_i - \mu_j\|^2 / \alpha)^{-\frac{\alpha+1}{2}}}{\sum_{j'} (1 + \|z_i - \mu_{j'}\|^2 / \alpha)^{-\frac{\alpha+1}{2}}},$$

- ii) **KL Divergence Minimization**: update the deep mapping f_θ and refine the cluster centroids by learning from current high confidence assignments using an auxiliary target distribution
- 해당 모델은 soft assignment가 target distribution을 맞출 수 있게 반복됨(훈련됨)

- **Target distribution**: $p_{ij} = \frac{q_{ij}^2 / f_j}{\sum_{j'} q_{ij'}^2 / f_{j'}}$ 여기서, $f_j = \sum_i q_{ij}$ 는 Soft cluster frequencies

- **Loss L**: $L = \text{KL}(P \| Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}.$

대덕대로 구간

시연회 대상 구역 선정

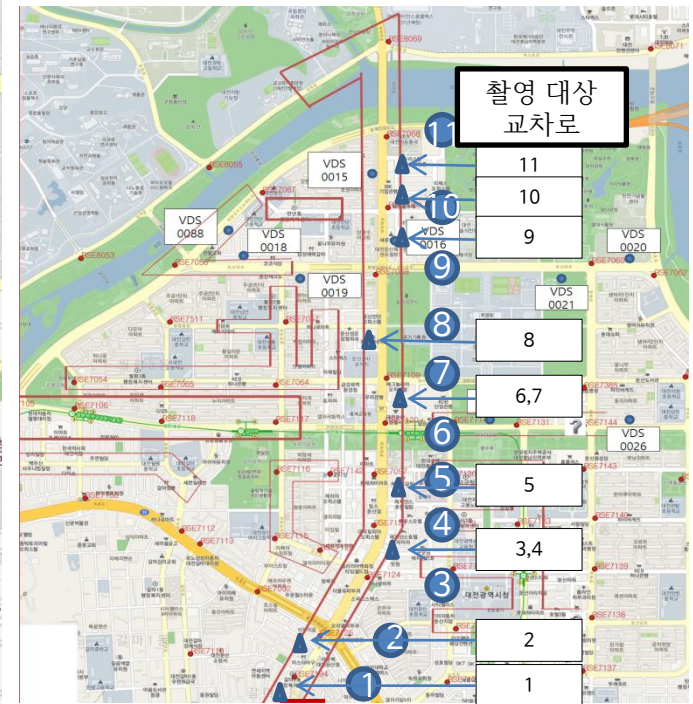
● 대상지 특성

- 대덕대교네거리 및 정부청사역 네거리, 큰마을네거리를 지나는 대덕대로 일부구간 - VDS 2개, DSRC(RSE) 11개 활용 가능
- SA(신호 연동): 주요 12개 교차로를 지나는 구간에 길게 적용 되어 있음 (약 3km)

대덕대교네거리 (엑스포공원 방면, North)



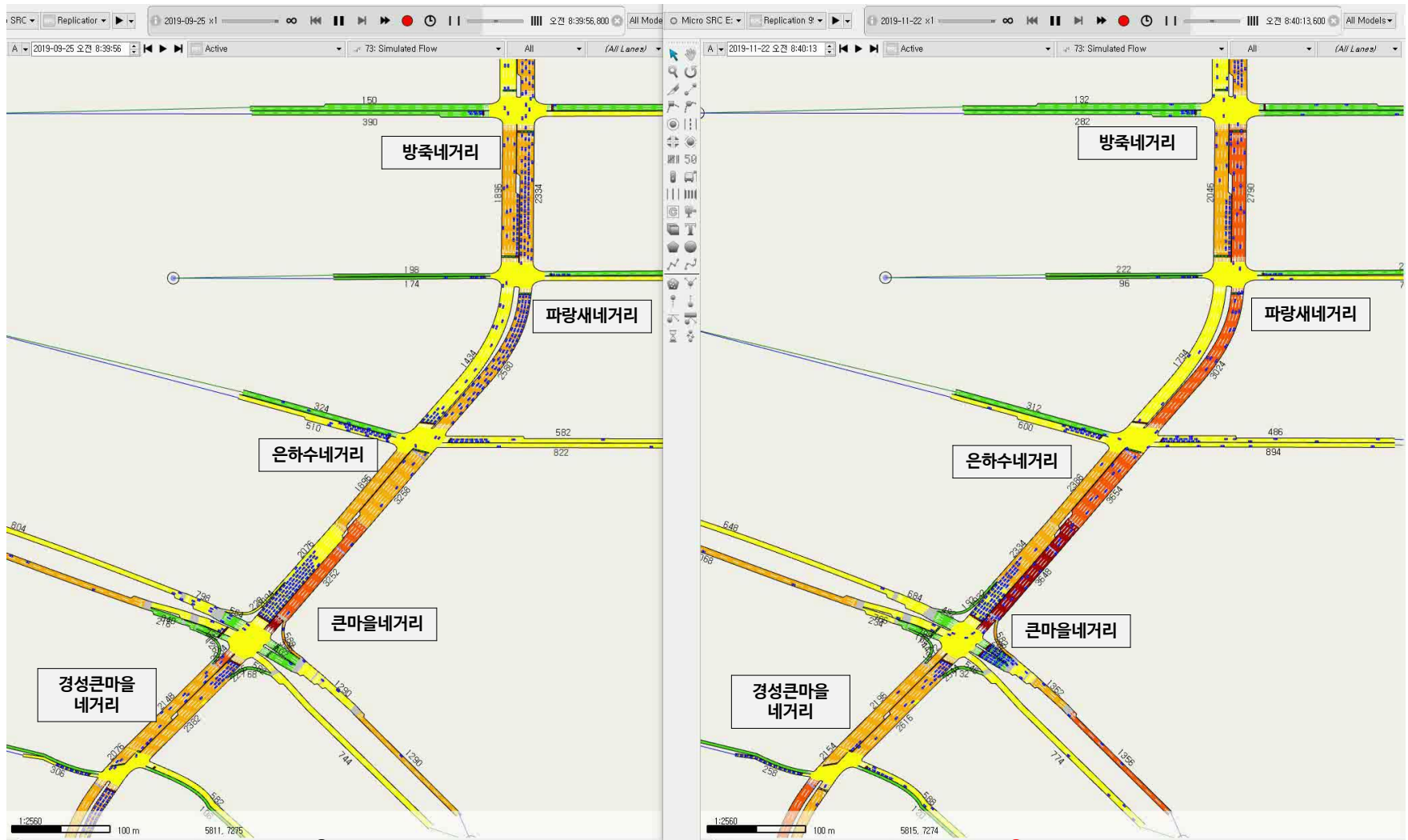
경성큰마을네거리 (갤러리아백화점 방면, South)



장비 설치 위치(세모) 및 촬영 대상 교차로(12개)

신호최적화를 위한 시뮬레이션 구축

신호최적화를 위한 시뮬레이션 구축 (오전 8시_1, 1~5교차로)



Before

After

Input data 구성

Input data 구성

● 대덕대로 Input data

- Input data 구성: 속도, 교통량, 밀도 순으로 데이터가 구성 (3360행 795열)
 - 공간적 범위 : 대덕대로 44개 디텍터 (11개 교차로 X 4개 방향)
 - 시간적 범위 : 19년8월 18일(일) ~ 9월21일(토) - 오전 6시 0분 ~ 오후 9시 50분(10분단위) 속도, 교통량, 밀도
 - 데이터 : ①raw - 실제 데이터, ②normalize - 노말라이즈 수행 데이터(x-min/Max-min)
 - D열 ~ JG열 : 속도(MAX : 62.51, min : 0)
 - JH열 ~ TK열 : 교통량(MAX : 688, min : 0)
 - TL열 ~ ADO열 : 밀도(MAX : 167.84, min : 0)

10분 단
위 96개
(16시간)
X 35일(5
주)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	Date	Weekday	Time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
2	20190818	1	600	0.5497	0.5921	0.6909	0.8152	0.8464	0.7088	0.6415	0.836	0.7391	0.7255	0.7941	0.6485	0.7092	0.6418	0.8562	0.538	0.8157	0.6189	0.6952
3	20190818	1	610	0.5353	0.5562	0.7263	0.8128	0.8459	0.6381	0.5895	0.8362	0.7042	0.7212	0.8079	0.5745	0.7652	0.6114	0.8514	0.5505	0.8052	0.5316	0.7002
4	20190818	1	620	0.5534	0.5329	0.6948	0.8076	0.8461	0.6577	0.6437	0.8371	0.7047	0.7959	0.8344	0.714	0.6495	0.6815	0.8821	0.5817	0.7717	0.6231	0.6954
5	20190818	1	630	0.5094	0.5967	0.7204	0.8226	0.8568	0.6631	0.662	0.817	0.7124	0.7504	0.7645	0.6117	0.6524	0.6641	0.8269	0.5687	0.8136	0.6725	0.6231
6	20190818	1	640	0.5582	0.637	0.7535	0.8269	0.8672	0.6239	0.6058	0.8515	0.7026	0.7516	0.8179	0.6653	0.7458	0.5354	0.8461	0.7045	0.7818	0.6479	0.6487
7	20190818	1	650	0.555	0.5615	0.76	0.81	0.8493	0.6956	0.6391	0.8267	0.6325	0.7759	0.7943	0.53	0.6492	0.6087	0.8607	0.5383	0.7567	0.5685	0.5716
8	20190818	1	700	0.5078	0.5414	0.6543	0.8045	0.8194	0.6666	0.6282	0.8219	0.7293	0.7314	0.816	0.6885	0.7031	0.6892	0.8709	0.6364	0.7509	0.6586	0.7016
9	20190818	1	710	0.5058	0.5828	0.7119	0.8223	0.8619	0.6861	0.6628	0.8528	0.7055	0.7324	0.7596	0.5805	0.6034	0.5957	0.8258	0.6335	0.7058	0.7376	0.6682
10	20190818	1	720	0.5574	0.5316	0.7061	0.8352	0.8583	0.6863	0.6429	0.8355	0.7106	0.7504	0.7949	0.5383	0.6311	0.5975	0.8605	0.6255	0.7532	0.6647	0.6007
11	20190818	1	730	0.5343	0.5754	0.762	0.8314	0.8666	0.6613	0.6044	0.81	0.6682	0.7365	0.8045	0.6177	0.6325	0.5993	0.8858	0.6594	0.768	0.6226	0.6409
12	20190818	1	740	0.5174	0.5418	0.7213	0.8299	0.8429	0.6146	0.6319	0.8245	0.6869	0.74	0.7992	0.5828	0.6893	0.6988	0.8699	0.586	0.7479	0.6605	0.701
13	20190818	1	750	0.4959	0.5861	0.7037	0.8229	0.8392	0.6756	0.6468	0.81	0.69	0.7044	0.7564	0.6181	0.5082	0.7031	0.8627	0.653	0.7274	0.6703	0.6834
14	20190818	1	800	0.537	0.5825	0.7124	0.8176	0.8391	0.6642	0.5786	0.8231	0.7093	0.7285	0.7762	0.5823	0.4857	0.698	0.8068	0.6514	0.6361	0.662	0.6807
15	20190818	1	810	0.5837	0.5804	0.7024	0.7935	0.8269	0.6482	0.5935	0.7997	0.6442	0.6908	0.7759	0.5122	0.4492	0.6681	0.8192	0.6399	0.5949	0.7216	0.6244
16	20190818	1	820	0.5169	0.5569	0.7365	0.7927	0.8515	0.6631	0.6009	0.785	0.706	0.7244	0.7644	0.6305	0.4796	0.6855	0.8071	0.6218	0.5847	0.7125	0.7008
17	20190818	1	830	0.507	0.5415	0.7312	0.837	0.868	0.6589	0.6039	0.8186	0.6935	0.6928	0.7831	0.5383	0.4577	0.6841	0.8386	0.638	0.618	0.6588	0.6841
18	20190818	1	840	0.5014	0.5327	0.6839	0.7884	0.8288	0.653	0.6025	0.7778	0.694	0.7074	0.7796	0.5394	0.4566	0.6679	0.8127	0.5881	0.6093	0.7204	0.7308
19	20190818	1	850	0.5262	0.5673	0.6954	0.7901	0.8357	0.6124	0.5837	0.808	0.6277	0.6541	0.7616	0.6421	0.4874	0.6322	0.8136	0.6049	0.602	0.6943	0.7053
20	20190818	1	900	0.5671	0.5641	0.7024	0.769	0.8357	0.6124	0.5837	0.808	0.6277	0.6541	0.7616	0.6421	0.4874	0.6322	0.8136	0.6049	0.602	0.6943	0.7053
21	20190818	1	910	0.5685	0.5706	0.7052	0.7604	0.8357	0.6124	0.5837	0.808	0.6277	0.6541	0.7616	0.6421	0.4874	0.6322	0.8136	0.6049	0.602	0.6943	0.7053
22	20190818	1	920	0.5901	0.5446	0.6888	0.7762	0.8357	0.6124	0.5837	0.808	0.6277	0.6541	0.7616	0.6421	0.4874	0.6322	0.8136	0.6049	0.602	0.6943	0.7053
23	20190818	1	930	0.6057	0.5641	0.6986	0.7946	0.8357	0.6124	0.5837	0.808	0.6277	0.6541	0.7616	0.6421	0.4874	0.6322	0.8136	0.6049	0.602	0.6943	0.7053
24	20190818	1	940	0.5897	0.5513	0.7093	0.7925	0.8357	0.6124	0.5837	0.808	0.6277	0.6541	0.7616	0.6421	0.4874	0.6322	0.8136	0.6049	0.602	0.6943	0.7053
25	20190818	1	950	0.5938	0.5781	0.7362	0.769	0.8357	0.6124	0.5837	0.808	0.6277	0.6541	0.7616	0.6421	0.4874	0.6322	0.8136	0.6049	0.602	0.6943	0.7053
26	20190818	1	1000	0.6183	0.5817	0.7189	0.7519	0.8357	0.6124	0.5837	0.808	0.6277	0.6541	0.7616	0.6421	0.4874	0.6322	0.8136	0.6049	0.602	0.6943	0.7053
27	20190818	1	1010	0.6269	0.5826	0.7152	0.7893	0.8357	0.6124	0.5837	0.808	0.6277	0.6541	0.7616	0.6421	0.4874	0.6322	0.8136	0.6049	0.602	0.6943	0.7053
28	20190818	1	1020	0.6127	0.5532	0.6999	0.7405	0.8357	0.6124	0.5837	0.808	0.6277	0.6541	0.7616	0.6421	0.4874	0.6322	0.8136	0.6049	0.602	0.6943	0.7053
29	20190818	1	1030	0.6082	0.573	0.6845	0.75	0.8357	0.6124	0.5837	0.808	0.6277	0.6541	0.7616	0.6421	0.4874	0.6322	0.8136	0.6049	0.602	0.6943	0.7053
30	20190818	1	1040	0.6317	0.5727	0.7111	0.7948	0.8357	0.6124	0.5837	0.808	0.6277	0.6541	0.7616	0.6421	0.4874	0.6322	0.8136	0.6049	0.602	0.6943	0.7053
31	20190818	1	1050	0.6241	0.5754	0.6991	0.7528	0.8357	0.6124	0.5837	0.808	0.6277	0.6541	0.7616	0.6421	0.4874	0.6322	0.8136	0.6049	0.602	0.6943	0.7053
32	20190818	1	1100	0.6362	0.5399	0.7045	0.7637	0.8357	0.6124	0.5837	0.808	0.6277	0.6541	0.7616	0.6421	0.4874	0.6322	0.8136	0.6049	0.602	0.6943	0.7053
33	20190818	1	1110	0.6186	0.5858	0.6967	0.7581	0.7517	0.665	0.6261	0.8007	0.6111	0.5663	0.7876	0.6159	0.6833	0.6802	0.8037	0.6991	0.6962	0.6369	0.6714
34	20190818	1	1120	0.6209	0.5682	0.7039	0.7645	0.7405	0.6607	0.6501	0.8016	0.6006	0.6066	0.7777	0.5997	0.6845	0.6762	0.8034	0.6896	0.6805	0.6804	0.7149
35	20190818	1	1130	0.6093	0.5802	0.6826	0.7645	0.7252	0.6679	0.6373	0.7831	0.5908	0.5575	0.7839	0.5625	0.6588	0.6573	0.7888	0.6517	0.6882	0.6386	0.6935

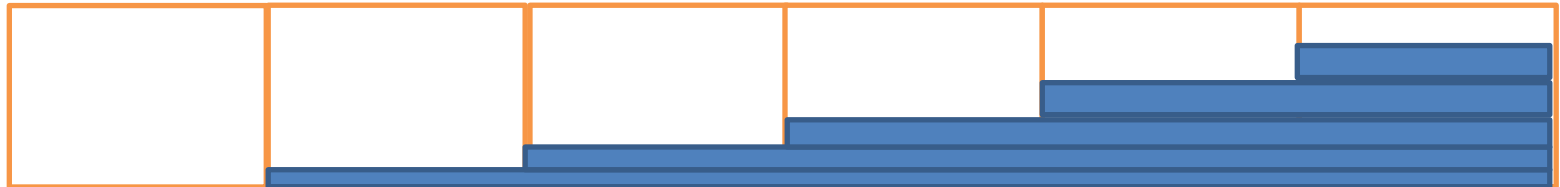
속도 교통량 밀도별
X 6개 SET
X 44개 디텍터

Input data 구성

Input data 구성

대덕대로 Input data

- Input data 구성: 지표별로 6개 SET를 구성하여 DEC가 가능할 수 있도록 데이터를 구성



10분 단
위 96개
(16시간)
X 35일 (5
주)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	Date	Weekday	Time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
2	20190818	1	600	0.5497	0.5921	0.6909	0.8152	0.8464	0.7088	0.6415	0.836	0.7391	0.7255	0.7941	0.6485	0.7092	0.6418	0.8562	0.538	0.8157	0.6189	0.6952
3	20190818	1	610	0.5353	0.5562	0.7263	0.8128	0.8459	0.6381	0.5895	0.8362	0.7042	0.7212	0.8079	0.5745	0.7652	0.6114	0.8514	0.5505	0.8052	0.5316	0.7002
4	20190818	1	620	0.5534	0.5329	0.6948	0.8076	0.8461	0.6577	0.6437	0.8371	0.7047	0.7959	0.8344	0.714	0.6495	0.6815	0.8821	0.5817	0.7717	0.6231	0.6954
5	20190818	1	630	0.5094	0.5967	0.7204	0.8226	0.8568	0.6631	0.662	0.817	0.7124	0.7504	0.7645	0.6117	0.6524	0.6641	0.8269	0.5687	0.8136	0.6725	0.6231
6	20190818	1	640	0.5582	0.637	0.7535	0.8269	0.8672	0.6239	0.6058	0.8515	0.7026	0.7516	0.8179	0.6653	0.7458	0.5354	0.8461	0.7045	0.7818	0.6479	0.6487
7	20190818	1	650	0.555	0.5615	0.76	0.81	0.8493	0.6956	0.6391	0.8267	0.6325	0.7759	0.7943	0.53	0.6492	0.6087	0.8607	0.5383	0.7567	0.5685	0.5716
8	20190818	1	700	0.5078	0.5414	0.6543	0.8045	0.8194	0.6666	0.6282	0.8219	0.7293	0.7314	0.816	0.6885	0.7031	0.6892	0.8709	0.6364	0.7509	0.6586	0.7016
9	20190818	1	710	0.5058	0.5828	0.7119	0.8223	0.8619	0.6861	0.6628	0.8528	0.7055	0.7324	0.7596	0.5805	0.6034	0.5957	0.8258	0.6335	0.7058	0.7376	0.6682
10	20190818	1	720	0.5574	0.5316	0.7061	0.8352	0.8583	0.6863	0.6429	0.8355	0.7106	0.7504	0.7949	0.5383	0.6311	0.5975	0.8605	0.6255	0.7532	0.6647	0.6007
11	20190818	1	730	0.5343	0.5754	0.762	0.8314	0.8666	0.6613	0.6044	0.81	0.6682	0.7365	0.8045	0.6177	0.6325	0.5993	0.8858	0.6594	0.768	0.6226	0.6409
12	20190818	1	740	0.5174	0.5418	0.7213	0.8299	0.8429	0.6146	0.6319	0.8245	0.6869	0.74	0.7992	0.5828	0.6893	0.6988	0.8699	0.586	0.7479	0.6605	0.701
13	20190818	1	750	0.4959	0.5861	0.7037	0.8229	0.8392	0.6756	0.6468	0.81	0.69	0.7044	0.7564	0.6181	0.5082	0.7031	0.8627	0.653	0.7274	0.6703	0.6834
14	20190818	1	800	0.537	0.5825	0.7124	0.8176	0.8391	0.6642	0.5786	0.8231	0.7093	0.7285	0.7762	0.5823	0.4857	0.698	0.8068	0.6514	0.6361	0.662	0.6807
15	20190818	1	810	0.5837	0.5804	0.7024	0.7935	0.8269	0.6482	0.5935	0.7997	0.6442	0.6908	0.7759	0.5122	0.4492	0.6681	0.8192	0.6399	0.5949	0.7216	0.6244
16	20190818	1	820	0.5169	0.5569	0.7365	0.7927	0.8515	0.6631	0.6009	0.785	0.706	0.7244	0.7644	0.6305	0.4796	0.6855	0.8071	0.6218	0.5847	0.7125	0.7008
17	20190818	1	830	0.507	0.5415	0.7312	0.837	0.868	0.6589	0.6039	0.8186	0.6935	0.6928	0.7831	0.5383	0.4577	0.6841	0.8386	0.638	0.618	0.6588	0.6841
18	20190818	1	840	0.5014	0.5327	0.6839	0.7884	0.8288	0.653	0.6025	0.7778	0.694	0.7074	0.7796	0.5394	0.4566	0.6679	0.8127	0.5881	0.6093	0.7204	0.7308
19	20190818	1	850	0.5262	0.5673	0.6954	0.7901	0.8357	0.6124	0.5837	0.808	0.6277	0.6541	0.7616	0.6421	0.4874	0.6322	0.8136	0.6049	0.602	0.6943	0.7053
20	20190818	1	900	0.5671	0.5641	0.7024	0.769	0.769	0.769	0.769	0.769	0.769	0.769	0.769	0.769	0.769	0.6841	0.8296	0.6725	0.6409	0.674	0.6444
21	20190818	1	910	0.5685	0.5706	0.7052	0.7604	0.7604	0.7604	0.7604	0.7604	0.7604	0.7604	0.7604	0.7604	0.7604	0.657	0.8565	0.6508	0.6377	0.6951	0.6445
22	20190818	1	920	0.5901	0.5446	0.6888	0.7762	0.7762	0.7762	0.7762	0.7762	0.7762	0.7762	0.7762	0.7762	0.7762	0.6554	0.7916	0.6359	0.6375	0.7146	0.6949
23	20190818	1	930	0.6057	0.5641	0.6986	0.7946	0.7946	0.7946	0.7946	0.7946	0.7946	0.7946	0.7946	0.7946	0.7946	0.665	0.8327	0.6687	0.7415	0.6394	0.6935
24	20190818	1	940	0.5897	0.5513	0.7093	0.7925	0.7925	0.7925	0.7925	0.7925	0.7925	0.7925	0.7925	0.7925	0.7925	0.6329	0.8152	0.6917	0.7394	0.6097	0.7087
25	20190818	1	950	0.5938	0.5781	0.7362	0.769	0.769	0.769	0.769	0.769	0.769	0.769	0.769	0.769	0.769	0.6786	0.832	0.6873	0.7562	0.6447	0.722
26	20190818	1	1000	0.6183	0.5817	0.7189	0.7519	0.7519	0.7519	0.7519	0.7519	0.7519	0.7519	0.7519	0.7519	0.7519	0.6674	0.8144	0.6968	0.7314	0.637	0.6898
27	20190818	1	1010	0.6269	0.5826	0.7152	0.7893	0.7893	0.7893	0.7893	0.7893	0.7893	0.7893	0.7893	0.7893	0.7893	0.6565	0.8098	0.6618	0.6487	0.6458	0.6983
28	20190818	1	1020	0.6127	0.5532	0.6999	0.7405	0.7405	0.7405	0.7405	0.7405	0.7405	0.7405	0.7405	0.7405	0.7405	0.6303	0.8351	0.6785	0.7119	0.6601	0.6669
29	20190818	1	1030	0.6082	0.573	0.6845	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.6562	0.8184	0.6741	0.6527	0.6444	0.7228
30	20190818	1	1040	0.6317	0.5727	0.7111	0.7948	0.7948	0.7948	0.7948	0.7948	0.7948	0.7948	0.7948	0.7948	0.7948	0.6498	0.8076	0.6956	0.6497	0.63	0.7264
31	20190818	1	1050	0.6241	0.5754	0.6991	0.7528	0.7528	0.7528	0.7528	0.7528	0.7528	0.7528	0.7528	0.7528	0.7528	0.6613	0.7957	0.6545	0.7501	0.6527	0.696
32	20190818	1	1100	0.6362	0.5399	0.7045	0.7637	0.7637	0.7637	0.7637	0.7637	0.7637	0.7637	0.7637	0.7637	0.7637	0.6594	0.7744	0.6919	0.6681	0.6253	0.6916
33	20190818	1	1110	0.6186	0.5858	0.6967	0.7581	0.7581	0.7581	0.7581	0.7581	0.7581	0.7581	0.7581	0.7581	0.7581	0.6802	0.8037	0.6991	0.6962	0.6369	0.6714
34	20190818	1	1120	0.6209	0.5682	0.7039	0.7645	0.7645	0.7645	0.7645	0.7645	0.7645	0.7645	0.7645	0.7645	0.7645	0.6762	0.8034	0.6896	0.6805	0.6604	0.7149
35	20190818	1	1130	0.6093	0.5802	0.6826	0.7645	0.7252	0.6679	0.6373	0.7831	0.5908	0.5575	0.7839	0.5625	0.6588	0.6573	0.7888	0.6517	0.6882	0.6386	0.6935

속도 교통량 밀도별
X 6개 SET
X 44개 디텍터

Input 대안 수립 - 총 17개

속도, 교통량, 밀도 > 열#	열의 개수	Data 종류	clustering 개수	저장파일
속도 3-267	1개:264	raw	5	#1~
교통량 267-531	2개:528		nor	
밀도 531-795	3개:792			



대안#	속도, 교통량, 밀도	Data 종류	clustering 개수(n_clusters =)	저장파일
1	3	nor	7	#1
2	3	nor	5	#2
3	3	nor	10	#3
4	3	nor	15	#4
5	3	nor	20	#5
6	속도	nor	10	#6
7	교통량	nor	10	#7

대안#	속도, 교통량, 밀도	Data 종류	clustering 개수(n_clusters =)	저장파일
8	밀도	nor	10	#8
9	속도	nor	10	#9
10	교통량	nor	10	#10
11	속도 밀도	nor	10	#11
12	속도 밀도	nor	5	#12
13	속도 교통량	nor	5	#13
14	교통량 밀도	nor	5	#14
15	속도	nor	5	#15
16	교통량	nor	5	#16
17	밀도	nor	5	#17

Output data 구성

Output data 구성

DEC Output data

- Output data 구성:
일별 시간별 Cluster
ID 부여
- Clustering ID가 바
르게 부여됐는지 확인
하는 과정 필요

	A	B	C	D
1	year	weekday	TRUE	cluster_id
2	20190818	1	600	1
3	20190818	1	610	1
4	20190818	1	620	1
5	20190818	1	630	1
6	20190818	1	640	1
7	20190818	1	650	1
8	20190818	1	700	1
9	20190818	1	710	1
10	20190818	1	720	1
11	20190818	1	730	1
12	20190818	1	740	1
13	20190818	1	750	1
14	20190818	1	800	1
15	20190818	1	810	1
16	20190818	1	820	1
17	20190818	1	830	1
18	20190818	1	840	1
19	20190818	1	850	1
20	20190818	1	900	1
21	20190818	1	910	4
22	20190818	1	920	4
23	20190818	1	930	4
24	20190818	1	940	4
25	20190818	1	950	4
26	20190818	1	1000	4
27	20190818	1	1010	4



Input 데이터

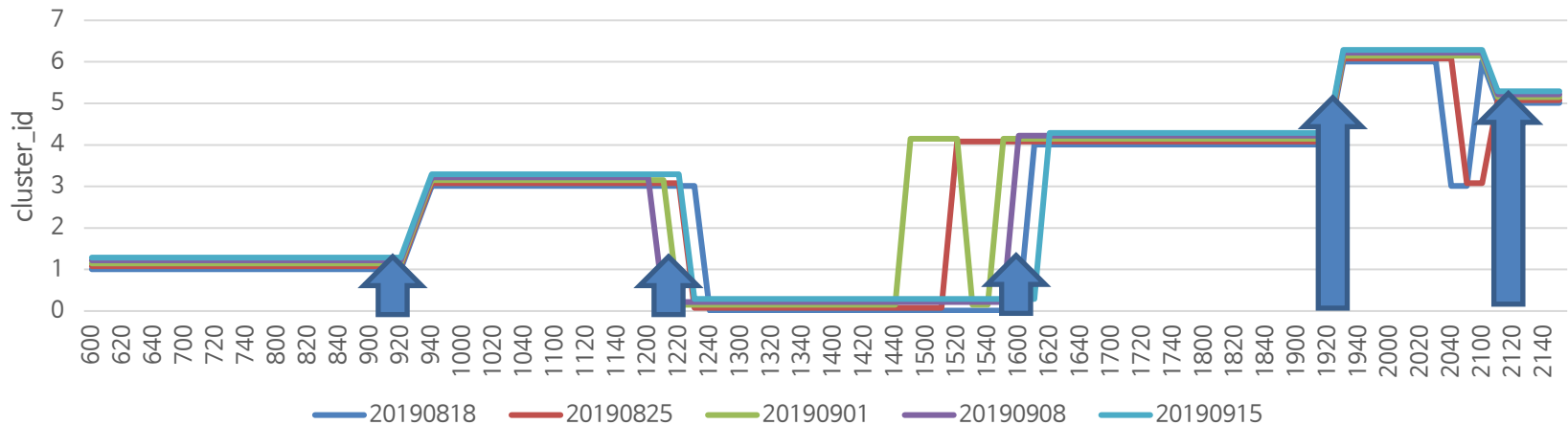
	A	B	C	D	E	F	G
1	time	speed	traffic	density	cluster_id	pcolor	
2	6:00	43.87725	38.4899	40.41083	2	mediumpurple1	
3	6:10	43.92793	42.25505	42.97629	2	mediumpurple1	
4	6:20	43.86104	46.77525	46.34341	2	mediumpurple1	
5	6:30	43.71811	52.55808	49.86571	2	mediumpurple1	
6	6:40	43.61795	59.90404	53.56551	2	mediumpurple1	
7	6:50	43.4751	68.39394	57.46886	2	mediumpurple1	
8	7:00	43.22889	77.81061	61.97616	2	mediumpurple1	
9	7:10	42.81038	87.92424	67.02199	2	mediumpurple1	
10	7:20	42.47783	97.63131	71.28606	2	mediumpurple1	
11	7:30	42.25348	107.0076	75.00851	2	mediumpurple1	
12	7:40	42.04066	116.0051	78.40313	2	mediumpurple1	
13	7:50	41.91846	124.2323	80.97232	2	mediumpurple1	
14	8:00	41.92525	132.3056	82.32596	2	mediumpurple1	
15	8:10	42.10381	140.7096	82.92755	2	mediumpurple1	
16	8:20	42.33258	150.2904	83.79995	2	mediumpurple1	
17	8:30	42.54798	159.5581	85.18119	2	mediumpurple1	
18	8:40	42.77616	168.6869	86.58417	2	mediumpurple1	
19	8:50	42.96298	178.399	88.27432	2	mediumpurple1	
20	9:00	43.10889	188.351	90.45427	2	mediumpurple1	
21	9:10	43.11558	197.5884	93.04184	2	mediumpurple1	
22	9:20	43.03904	204.2071	95.49843	2	mediumpurple1	
23	9:30	42.94429	210.4293	97.56909	6	steelblue	
24	9:40	42.87437	214.8283	99.16159	3	slateblue1	
25	9:50	42.79018	217.0985	100.3865	3	slateblue1	
26	10:00	42.69586	217.2449	101.1416	3	slateblue1	
27	10:10	42.6399	216.0657	101.1994	3	slateblue1	

Output data 활용

● 교통패턴 분류

- Output data 구성: 일별 시간별 Cluster ID를 통해 교통패턴(흐름)이 달라지는 시간대 분별
- 해당 부분은 오전첨두시간/오전비첨두시간/오후첨두시간/오후비첨두시간이 잘 구분되는 것으로 확인
- Clustering 개수(K-means 설정 개수)에 따라 결과가 상이함

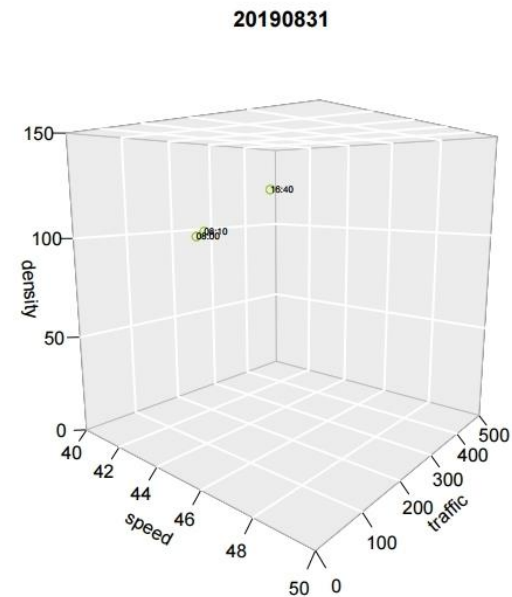
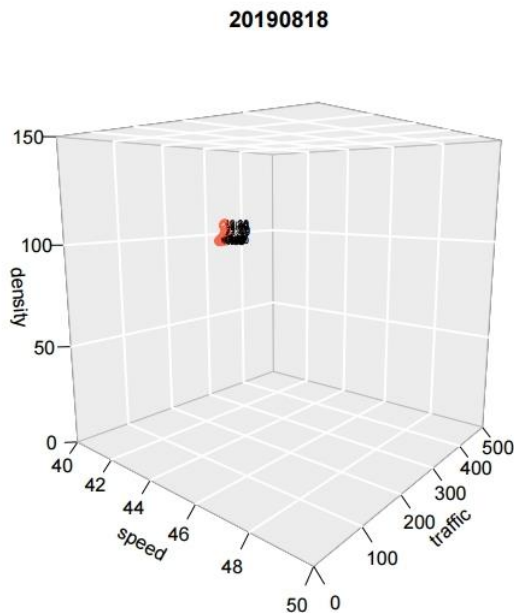
1_일요일



Output data 검증

DEC 결과 Plotting

- 3차원(속도/교통량/밀도) 그래프 Plotting을 통해 원시데이터의 1시간별 지표별 방향별(상행, 하행) 평균값과 비교
- Clustering된 점을 통해 개별 ID별 속도, 교통량, 밀도 간 차이 값 분별
- 추가적으로 일별 Clustering ID를 전부 Plotting해서 ID별로 분포가 다른지 확인



Q&A

