Machine Learning and Process Mining

Most of the lecture slides are made by Prof. van der Aalst.

Overview

AutoPM

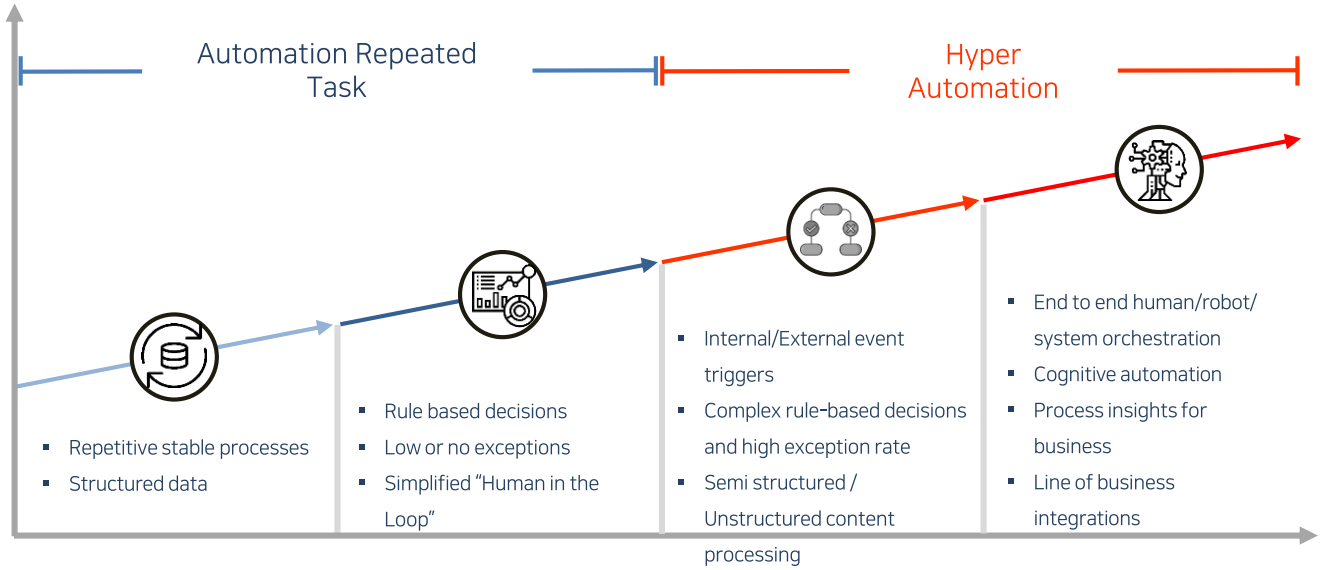
ML과 PM의 적용 필요성

Hyperautomation

Hyperautomation

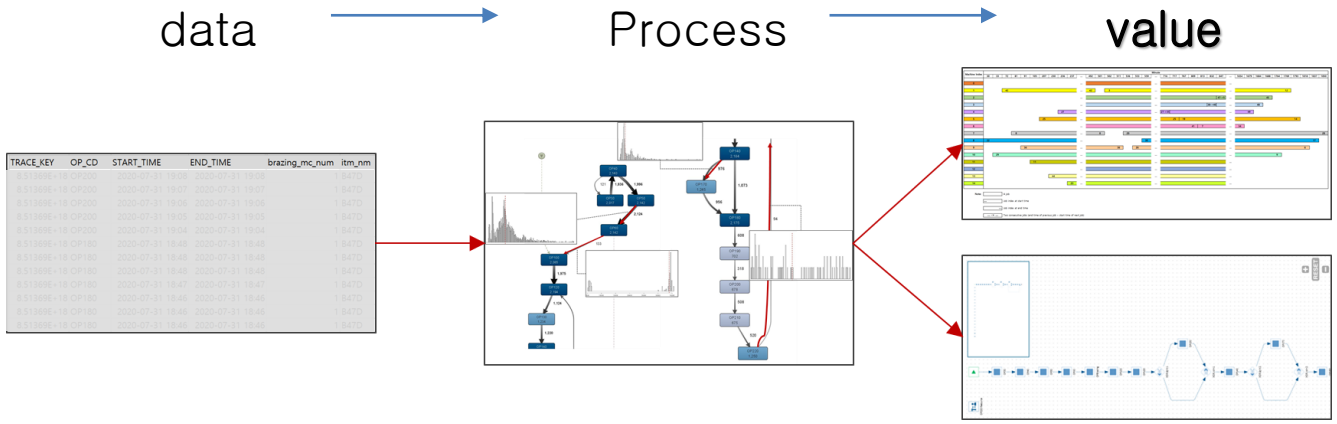
Definition

Combination of multiple digital transformation technologies, starting with Robotic Process Automation (RPA) at its core, and expanding automation capabilities with artificial intelligence (AI), process mining, advanced analytics, and a broader set of advanced tools so that previously *un-automatable tasks can be automated*.



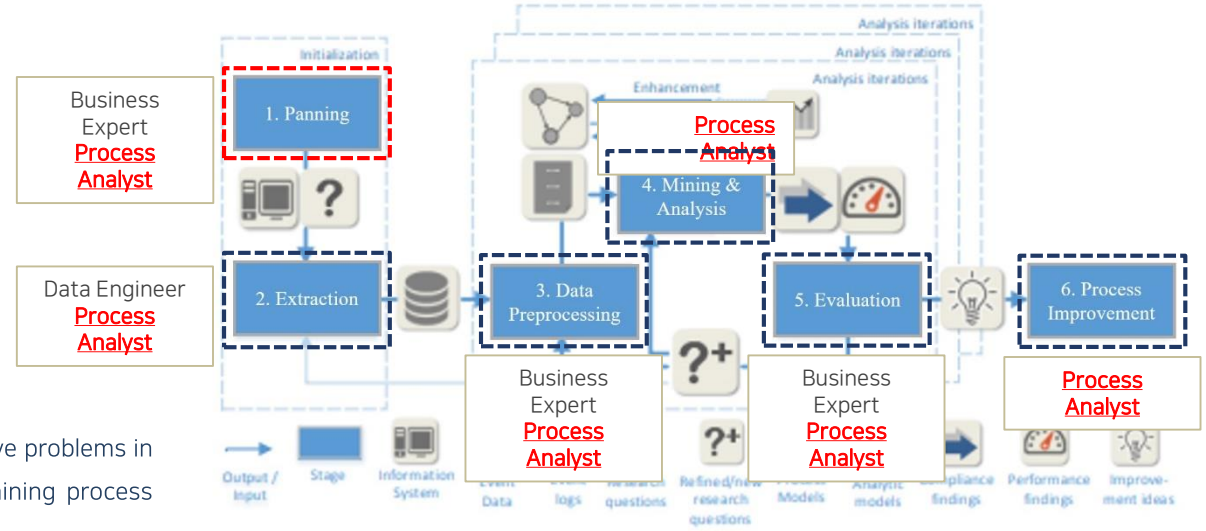
The key components of hyperautomation

- Robotic Process Automation
- Process Mining
- Artificial Intelligence
- Advanced Analytics



The key components of hyperautomation

- Robotic Process Automation
- Process Mining
- Artificial Intelligence
- Advanced Analytics



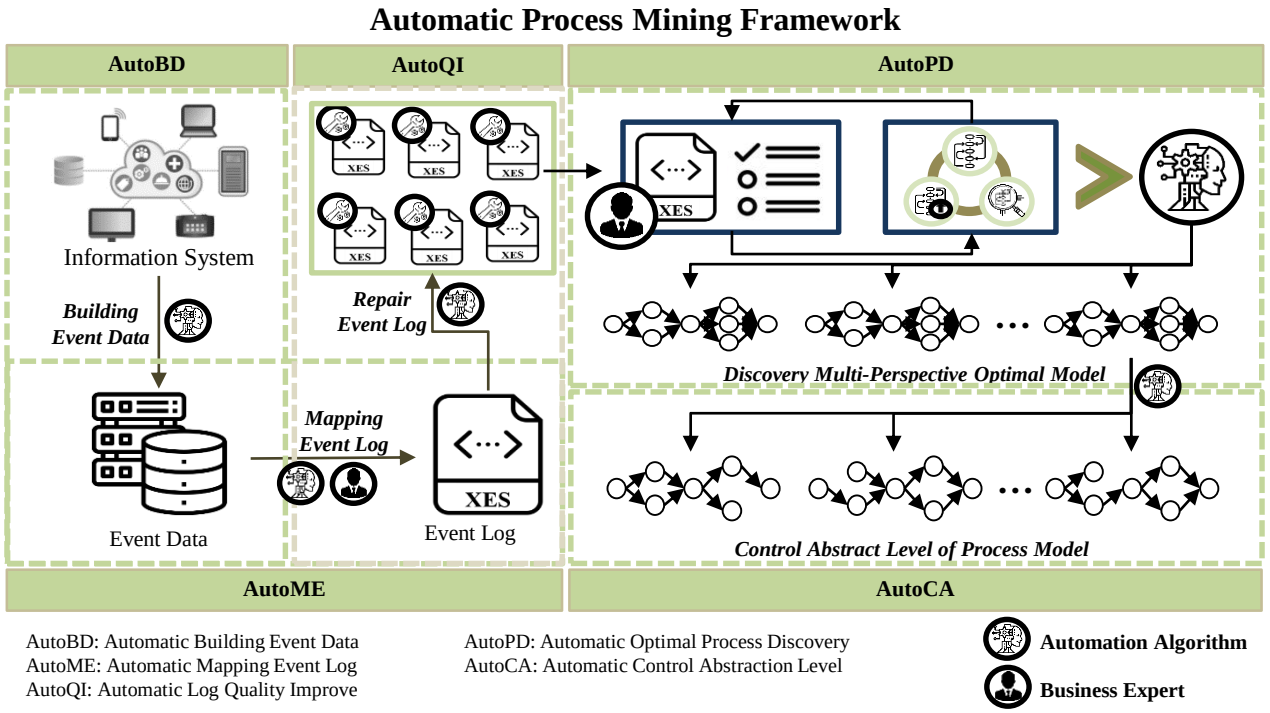
An overview of the PM² methodology

Problem

It is not easy for Domain Experts to solve problems in the field by performing the process mining process without the help of Process Analyst

AutoPM (inspired by AutoML)

(Goal) A methodology that allows business experts to easily perform process mining without the help of a Process Analyst.



Automatic Mapping Event Log to Event Data

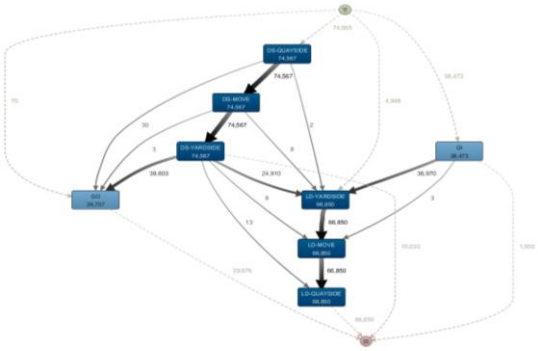
- I can understand 'timestamp'. But, what are 'Event', 'Activity', 'Trace', 'Resource', 'CaseID', and 'Originator'?

Event Data

Container No	Job Name	Machine No	Machine Type	Container Type	Complete Time	Pod	Block	Bay
C01	DS QuaySide	GC101	GC	M	2018-01-31 09:49:31	KR PUS	2M	75
C01	DS Move	YT555	YT	M	2018-01-31 09:54:31	KR PUS	2M	75
C01	DS YardSide	RS309	RS	M	2018-01-31 09:55:31	KR PUS	2M	75
C02	DS QuaySide	GC102	GC	F	2018-01-31 05:24:11	CN XMN	1B	12
C02	DS Move	YT512	YT	F	2018-01-31 05:34:51	CN XMN	1B	12
C02	DS YardSide	YC213	YC	F	2018-01-31 05:40:00	CN XMN	1B	12
C02	Re Handling	YC216	YC	F	2018-01-31 11:23:31	CN XMN	3C	27
C02	GateOut YardSide	YC213	YC	F	2018-01-31 00:38:03	CN XMN	3C	27
C02	GateOut	RT	RT	F	2018-01-24 00:45:00	CN XMN	3C	27
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Mapping

Column	Mapping result	Type
Container No	case:concept:name	Case identifier
Job Name	concept:name	Activity label
Machine No	org:resource	Originator label
Complete Time	time:timestamp	Timestamp



Research Results: When the training data is large enough

Event density embedding: A new embedding method that is easy to learn the event log structure.

Container No	Job Name	Machine No	Machine Type	Container Type	Complete Time	Pod	Block	Bay
C01	DS QuaySide	GC101	GC	M	2018-01-31 09:49:31	KR PUS	2M	75
C01	DS Move	YT555	YT	M	2018-01-31 09:54:31	KR PUS	2M	75
C01	DS YardSide	RS309	RS	M	2018-01-31 09:55:31	KR PUS	2M	75
C02	DS QuaySide	GC102	GC	F	2018-01-31 05:24:11	CN XMN	1B	12
C02	DS Move	YT517	YT	F	2018-01-31 05:34:51	CN XMN	1B	12
C02	DS YardSide	YC213	YC	F	2018-01-31 05:40:00	CN XMN	1B	12
C02	Re Handling	YC216	YC	F	2018-01-31 11:23:31	CN XMN	3C	27
C02	GateOut YardSide	YC216	YC	F	2018-01-31 00:38:03	CN XMN	3C	27
C02	GateOut	RT	RT	F	2018-01-24 00:45:00	CN XMN	3C	27
C03	DS QuaySide	GC102	GC	F	2018-01-31 05:24:11	CN XMN	2B	05
C03	DS Move	YT512	YT	F	2018-01-31 05:34:51	CN XMN	2B	05
C03	DS YardSide	YC213	YC	F	2018-01-31 05:40:00	CN XMN	2B	05
C03	GateOut YardSide	YC213	YC	F	2018-01-31 00:38:03	CN XMN	3C	27
C03	GateOut	RT	RT	F	2018-01-24 00:45:00	CN XMN	3C	27
C04	DS QuaySide	GC102	GC	M	2018-01-31 09:51:33	KR PUS	6A	21
C04	DS Move	YT556	YT	M	2018-01-31 09:51:43	KR PUS	6A	21
C04	DS YardSide	RS307	RS	M	2018-01-31 09:55:36	KR PUS	6A	21
C05	DS QuaySide	GC102	GC	M	2018-01-31 09:53:20	KR PUS	7B	56
C05	DS Move	YT555	YT	M	2018-01-31 09:55:50	KR PUS	7B	56
C05	DS YardSide	RS309	RS	M	2018-01-31 10:01:17	KR PUS	7B	56

Event Data

The reference column: ***Container No***

Container No	Job Name	Machine No	Machine Type	Container Type	Pod	Block	Bay
C01	3	3	3	1	1	1	1
C02	4	6	3	1	1	1	1
C03	3	3	3	1	1	1	1
C04	3	3	3	1	1	1	1

The reference column: ***Job Name***

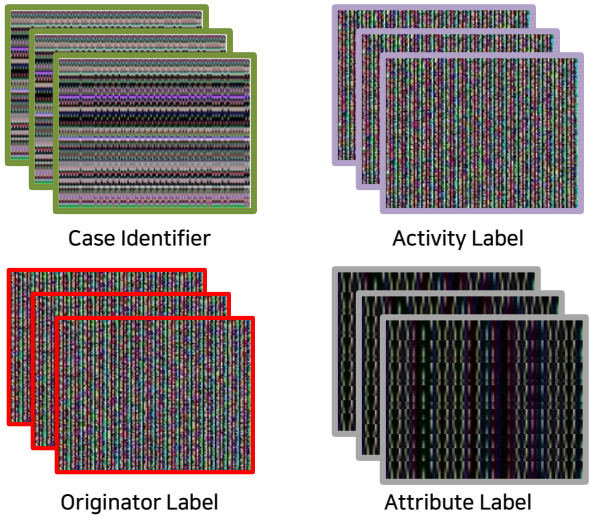
Job Name	Container No	Machine No	Machine Type	Container Type	Pod	Block	Bay
DS QuaySide	5	2	1	2	2	5	5
DS Move	5	4	1	2	2	5	5
DS YardSide	5	4	2	2	2	5	5
Re Handling	1	1	1	1	1	1	1
GateOut YardSide	1	2	1	1	1	2	2
GateOut	1	1	1	1	2	2	2

The reference column: ***Pod***

Pod	Container No	Job Name	Machine No	Machine Type	Container Type	Block	Bay
KR PUS	3	3	6	3	1	3	3
CN XMN	2	6	6	4	1	4	4

The reference column: ***Container Type***

Container Type	Container No	Job Name	Machine No	Machine Type	Pod	Block	Bay
M	3	3	6	3	1	3	3
F	2	6	6	4	1	4	4

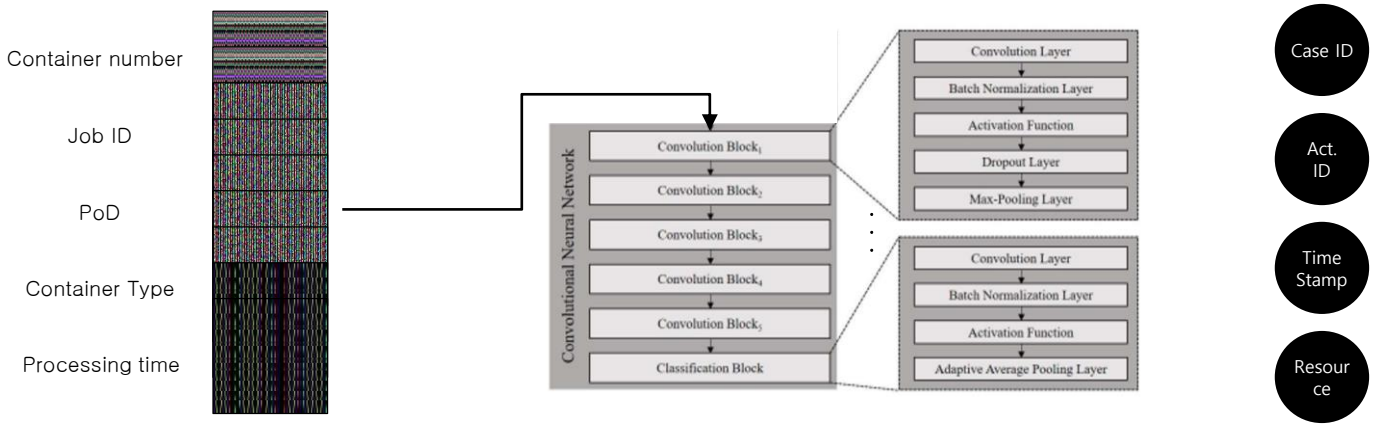


RGB image(3-dimensional array)

Research Results: When the training data is large enough

[Experiments 1-1] CNN training performance comparison experiment and mapping accuracy verification according to embedding method

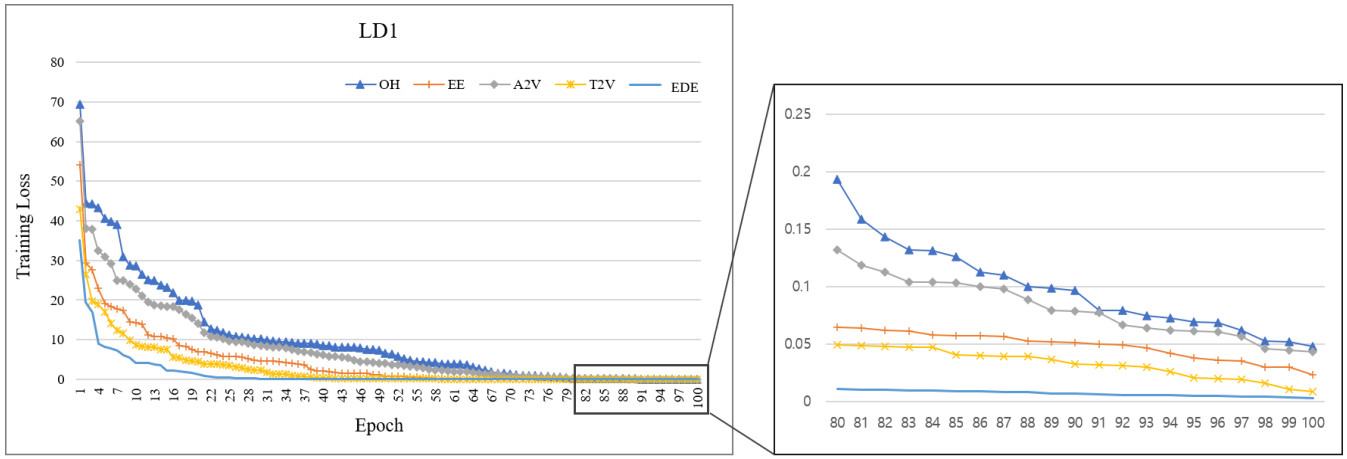
- ✓ Comparison of learning performance by performing training using CNN with the same structure using five different embedding methods (one-hot encoding, entity embedding, act2vec, trace2vec, **event density embedding***).
- ✓ Experiments were conducted using event log data (LD1 – LD 9)
- ✓ Data for learning and testing were divided into 70% and 30% of the total data, respectively, and performed 30 times.



Research Results: When the training data is large enough

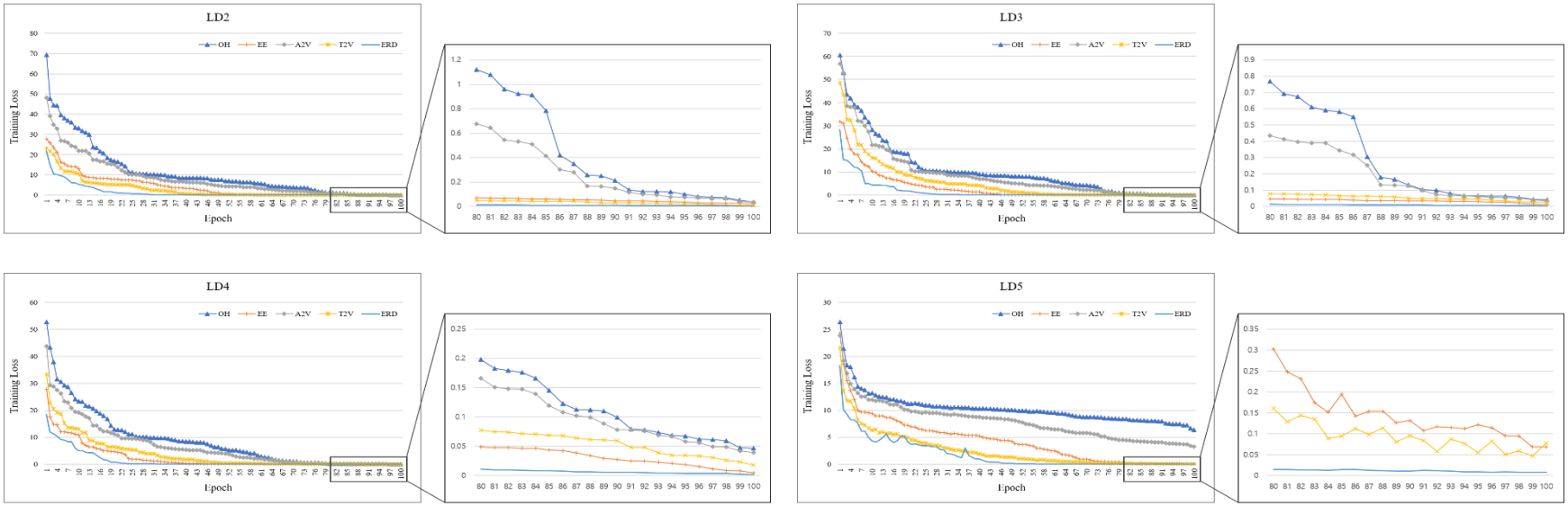
[Experiments 1-1] CNN training performance comparison experiment and mapping accuracy verification according to embedding method

✓ Learning performance comparison experiment result: The best learning performance was obtained when the proposed embedding method (EDE) was used compared to other methods.



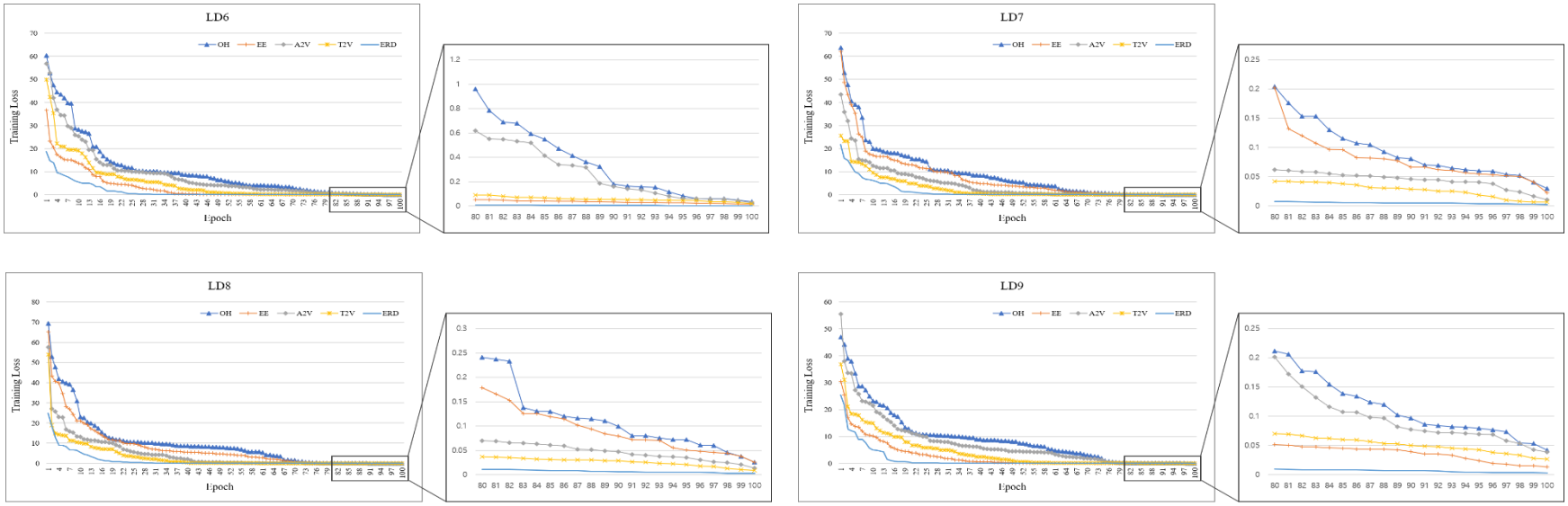
Research Results: When the training data is large enough

[Experiments 1-1] CNN training performance comparison experiment and mapping accuracy verification according to embedding method



Research Results: When the training data is large enough

[Experiments 1-1] CNN training performance comparison experiment and mapping accuracy verification according to embedding method



Research Results: When the training data is large enough

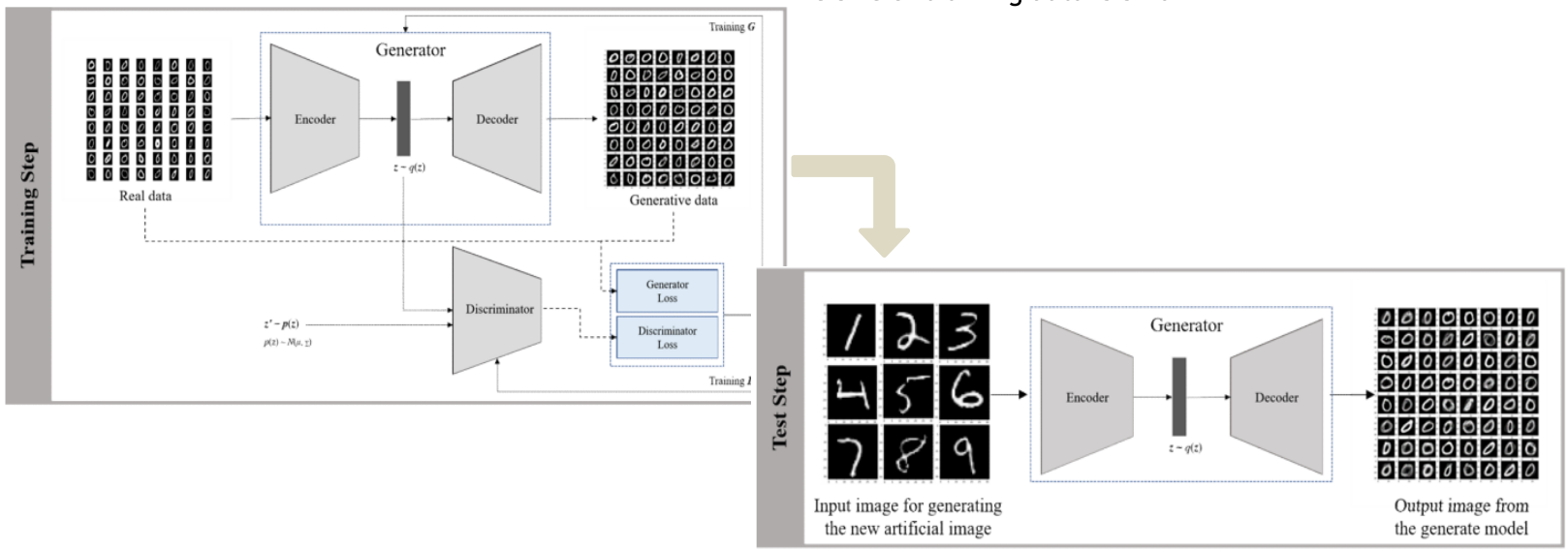
[Experiments 1-1] CNN training performance comparison experiment and mapping accuracy verification according to embedding method

✓ Mapping performance comparison experiment result: As a result of performing mapping using the learned model and verification data and comparing it with the actual label, the best mapping accuracy was shown when the Event Density Embedding method proposed in this study was used for all data.

	One-Hot Encoding	Entity Embedding	Act2Vec	Trace2Vec	EDE Embedding		One-Hot Encoding	Entity Embedding	Act2Vec	Trace2Vec	EDE Embedding
LD1	80.99 (3.54)	91.38 (1.31)	89.87 (1.45)	92.76 (0.98)	94.80 (0.32)	LD6	81.23 (2.12)	93.10 (0.94)	75.80 (2.62)	87.16 (0.99)	93.10 (0.58)
LD2	77.01 (3.69)	84.81 (2.31)	82.15 (2.37)	87.46 (1.54)	95.44 (1.40)	LD7	71.50 (3.55)	77.12 (1.28)	84.32 (0.74)	90.65 (0.51)	92.78 (0.47)
LD3	68.44 (4.76)	90.66 (1.10)	84.76 (2.55)	85.23 (1.20)	93.19 (0.37)	LD8	74.59 (5.82)	88.33 (2.44)	89.59 (1.67)	92.32 (0.66)	95.17 (0.13)
LD4	79.50 (5.94)	84.51 (1.94)	82.31 (3.33)	86.11 (0.82)	90.35 (0.12)	LD9	66.00 (5.49)	89.20 (1.98)	86.00 (2.29)	93.32 (1.84)	94.88 (1.67)
LD5	63.66 (4.23)	75.12 (3.33)	71.10 (3.37)	81.21 (2.91)	88.61 (1.34)						

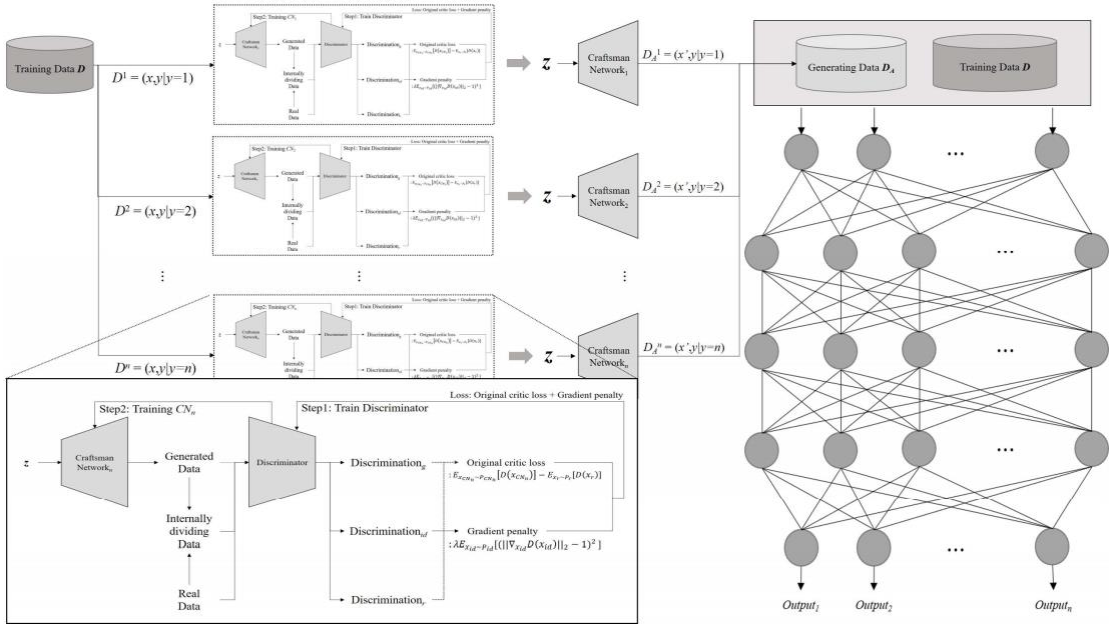
Research Results: When the training data is not large enough

"*Jangin training* (匠人, Master craftsman)" methodology based on generative models: Learning methodology for securing excellent learning performance in situations where the size of training data is small



Research Results: When the training data is not large enough

New supervised learning framework using Jangin training method



Research Results: When the training data is not large enough

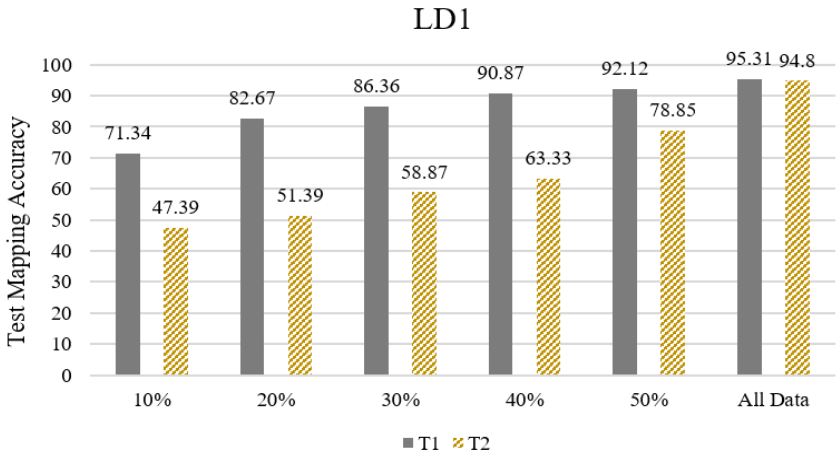
[Experiment 1-2] Accuracy comparison experiment of automatic mapping algorithm for event data using "*Jangin Training*" method when there is not enough learning data

- ✓ Experiments were conducted on 9 event data collected from the areas of the port logistics, steel manufacturing, IT, finance, and public administration.
- ✓ The network structure used the same structure as Experiment 1-1, and the EDE embedding method was used.
- ✓ After separating into 70% of the total training data, supervised learning is performed using only the ratio of 10% to 50% from the original training data.
- ✓ Data for the test is verified using 30% of the total data.

Research Results: When the training data is not large enough

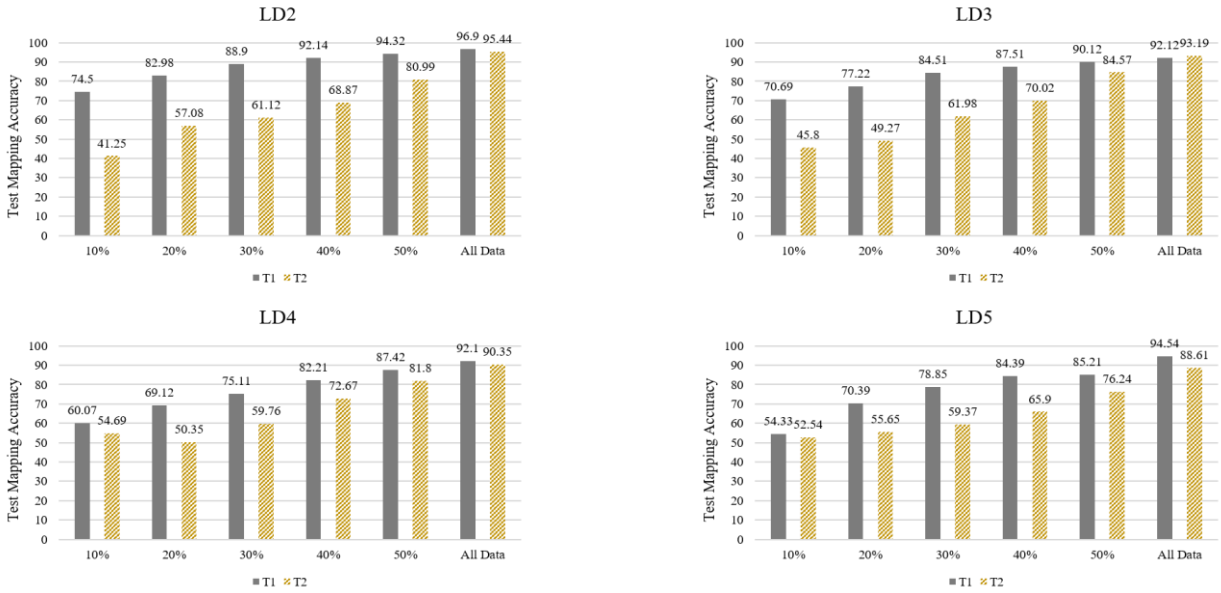
[Experiment 1-2] Accuracy comparison experiment of automatic mapping algorithm for event data using "*Jangin Training* (JT) " method when there is not enough learning data

- ✓ T1 is the result of supervised learning using the proposed learning method
- ✓ T2 is the result of supervised learning with the given data.
- ✓ Excellent mapping accuracy was shown when the proposed artisan learning method was used. JT contributes a lot to the classification accuracy.



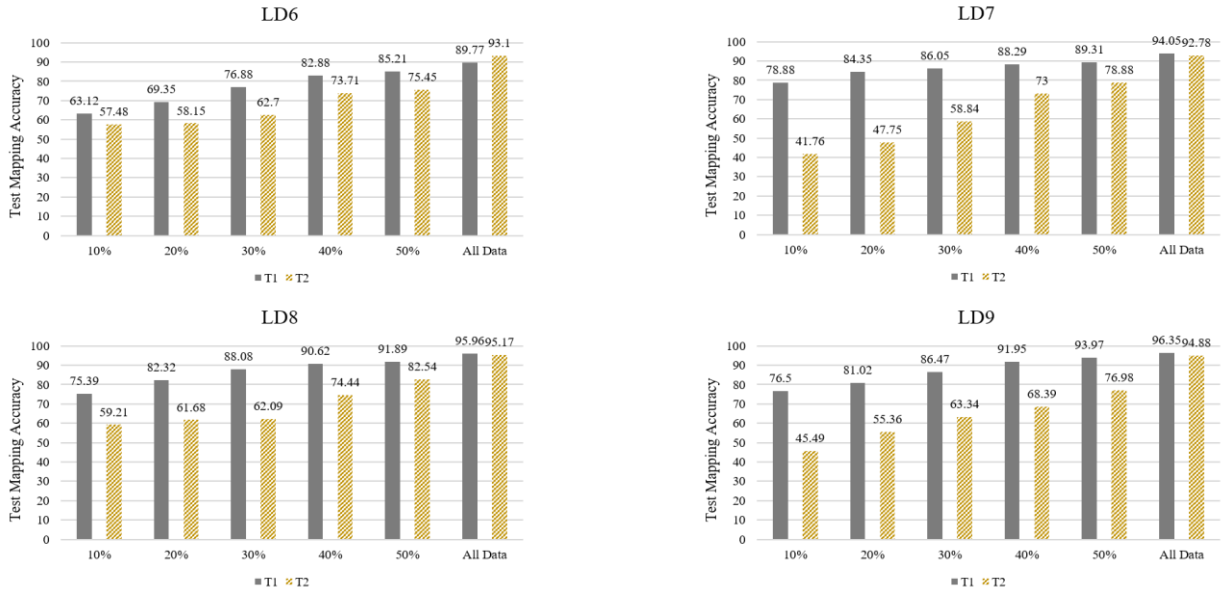
Research Results: When the training data is not large enough

[Experiment 1-2] Accuracy comparison experiment of automatic mapping algorithm for event data using "*Jangin Training*" method when there is not enough learning data



Research Results: When the training data is not large enough

[Experiment 1-2] Accuracy comparison experiment of automatic mapping algorithm for event data using "*Jangin Training*" method when there is not enough learning data



Contribution

- ✓ A new algorithm that can automatically map from event data to the event log is proposed.
- ✓ Proposed a new embedding method to learn the structure information of event data to perform mapping task
- ✓ By proposing a new learning method called Jangin training method, we propose a new learning method that can guarantee good mapping accuracy even in situations where the size of the learning data is small.

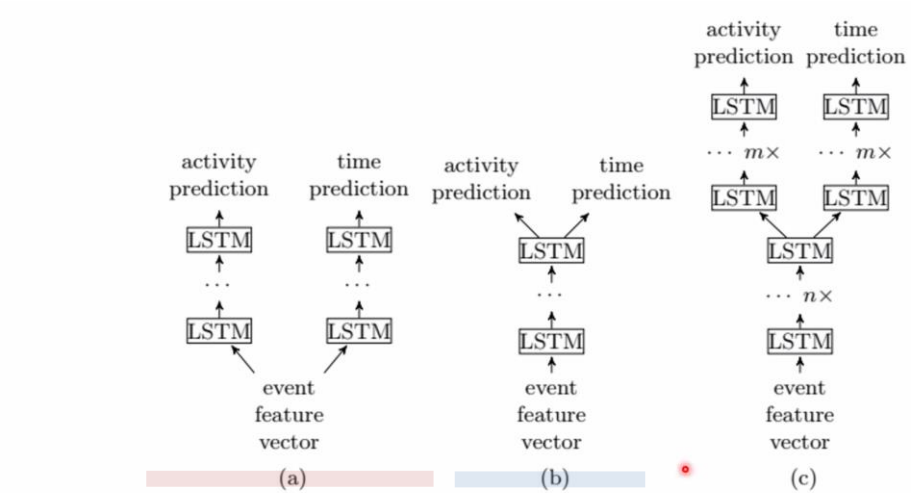
Supervised Learning in PM

Prediction of process remaining time

- Predictive Business Process Monitoring

Author	Method	Objective
van Dongen, Crooy and van der Aalst (2008)	Non-parametric Regression techniq ue	Predicting remaining time
van der Aalst, Schonenberg and Song (2011)	Annotated Transition Systems	Predicting remaining time
Folino, Guarascio and Pontieri (2012)	Predictive Clustering Tree	Predicting remaining time
Maggi et al. (2014)	Decision Trees	Predicting Linear Temporal Logic rules
Leontjeva et al. (2015)	HMM and Random Forest	Predicting outcome
Senderovich et al. (2015)	Queueing theory	Predicting delays
Francescomarino et al. (2016)	Clustering and Hyperparameters Op timization	Predicting Linear Temporal Logic Rules
Evermann, Rehse and Fettke (2017)	Deep learning, LSTM	Predicting next event
Tax et al. (2017)	Deep learning, LSTM	Predicting remaining time, outcome, and next event
Navarin et al. (2017)	Deep learning, Data Aware LSTM	Predicting remaining time
Senderovich et al. (2017)	Feature Engineering	Predicting remaining time
Francescomarino et al. (2018)	Genetic Algorithm	Optimizing hyperparameters

Predictive BP Monitoring by LSTM



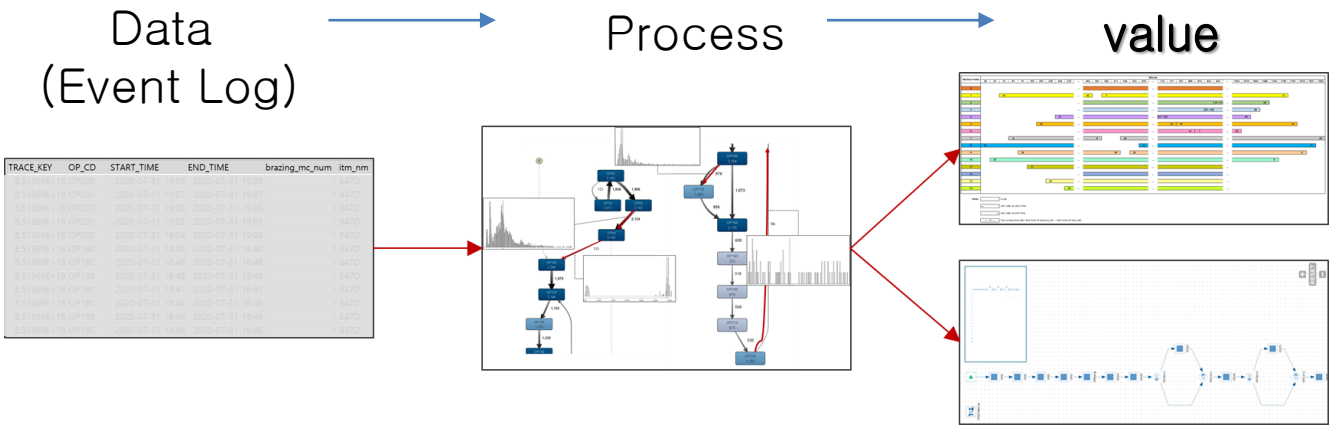
Source : Tax, Niek, I. Verenich, M. Rosa and M. Dumas. "Predictive Business Process Monitoring with LSTM Neural Networks." CAiSE (2017).

Layers	Shared	N/l	Helpdesk MAE in days Prefix 2	4	6	All	Accuracy	BPI'12 W MAE in days Prefix 2	10	20	All	Accuracy
						LSTM						
4	4	100	3.64	2.79	2.22	3.82	0.7076	1.75	1.49	1.02	1.61	0.7466
4	3	100	3.63	2.78	2.21	3.83	0.7075	1.74	1.47	1.01	1.59	0.7479
4	2	100	3.59	2.82	2.27	3.81	0.7114	1.72	**1.45**	1.00	1.57	0.7497
4	1	100	3.58	2.77	2.24	3.77	0.7074	1.70	1.46	1.01	1.59	0.7522
4	0	100	3.78	2.98	2.41	3.95	0.7072	1.74	1.47	1.05	1.61	0.7515
3	3	100	3.58	2.69	2.22	3.77	0.7116	**1.69**	1.47	1.02	1.58	0.7507
3	2	100	3.59	2.69	2.21	3.80	0.7118	**1.69**	1.47	1.01	1.57	0.7512
3	1	100	3.55	2.78	2.38	3.76	**0.7123**	1.72	1.47	1.04	1.59	0.7525
3	0	100	3.62	2.71	2.23	3.82	0.6924	1.81	1.51	1.07	1.66	0.7506
2	2	100	3.61	2.64	**2.11**	3.81	0.7117	1.72	1.46	1.02	1.58	0.7556
2	1	100	3.57	**2.61**	**2.11**	3.77	0.7119	**1.69**	**1.45**	1.01	**1.56**	**0.7600**
2	0	100	3.66	2.89	2.13	3.86	0.6985	1.74	1.46	0.99	1.60	0.7537
1	1	100	**3.54**	2.71	3.16	**3.75**	0.7072	1.71	1.47	**0.98**	1.57	0.7486
1	0	100	3.55	2.91	2.45	3.87	0.7110	1.72	1.46	1.05	1.59	0.7431
3	1	75	3.73	2.81	2.23	3.89	0.7118	1.73	1.49	1.07	1.62	0.7503
3	1	150	3.78	2.92	2.43	3.97	0.6918	1.81	1.52	1.14	1.71	0.7491
2	1	75	3.73	2.79	2.32	3.90	0.7045	1.72	1.47	1.03	1.59	0.7544
2	1	150	3.62	2.73	2.23	3.83	0.6982	1.74	1.49	1.08	1.65	0.7511
1	1	75	3.74	2.87	2.35	3.87	0.6925	1.75	1.50	1.07	1.64	0.7452
1	1	150	3.73	2.79	2.32	3.92	0.7103	1.72	1.48	1.02	1.60	0.7489
						RNN						
3	1	100	4.21	3.25	3.13	4.04	0.6581					
2	1	100	4.12	3.23	3.05	3.98	0.6624					
1	1	100	4.14	3.28	3.12	4.02	0.6597					
						Time prediction baselines						
Set abstraction [1]			6.15	4.25	4.07	5.83	-	2.71	1.64	1.02	1.97	-
Bag abstraction [1]			6.17	4.11	3.26	5.74	-	2.89	1.71	1.07	1.92	-
Sequence abstraction [1]			6.17	3.53	2.98	5.67	-	2.89	1.69	1.07	1.91	-
						Activity prediction baselines						
Evermann et al. [9]			-	-	-	-	-	-	-	-	-	0.623
Breuker et al. [3]			-	-	-	-	-	-	-	-	-	0.719

Table 1. Experimental results for the Helpdesk and BPI'12 W logs.

From event log to value

- Event log is special type of data



- Including categorical data
  - As input of PM: OK
  - As input of ML: NG

Event-log of container-handling process in port.

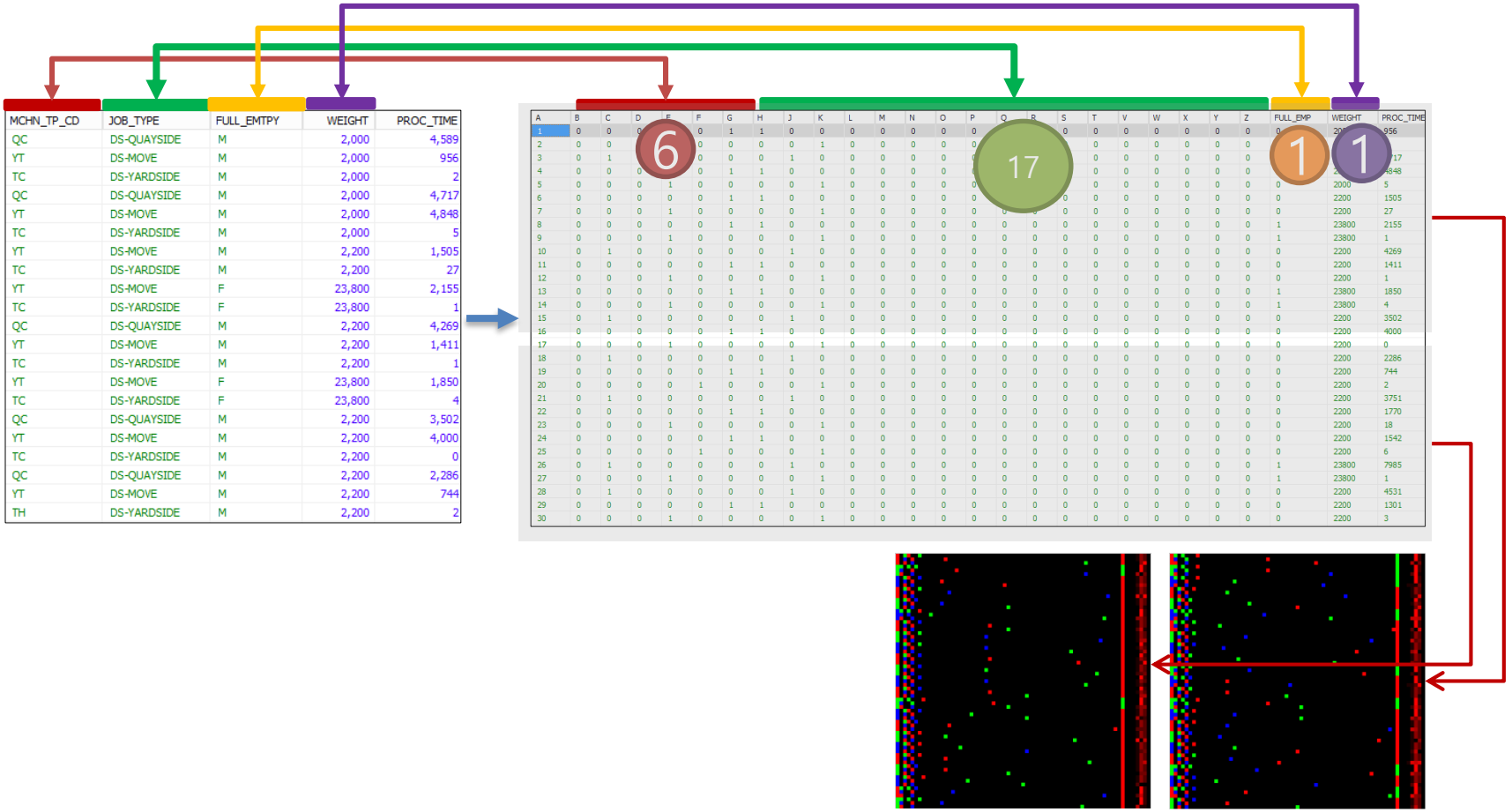
Container ID	Event Name*	Machine Type	Machine ID	Full/Empty*	Start Time	End Time
C0A12-2	DIS-VESSEL	MAQC	MCHNGC102	F	20180131211752	20180131211924
C0A12-2	DIS-YARD	MTTC	MCHNTC201	F	20180131213633	20180131213805
C0A12-2	DIS-MOVE	MTYT	MCHNYT501	F	20180131211924	20180131213633
C0B13-1	LOD-VESSEL	MTQC	MCHNGC101	F	20180131202120	20180131202353
C0B13-1	LOD-YARD	MATC	MCHNTC243	F	20180131200147	20180131200420
C0B13-1	LOD-MOVE	MTYT	MCHNYT541	F	20180131200420	20180131202120
C0C14-3	DIS-VESSEL	MTQC	MCHNGC139	M	20180131205748	20180131205920
C0C14-3	DIS-YARD	MTTC	MCHNTC246	M	20180131213633	20180131211304
C0C14-3	DIS-MOVE	MTYT	MCHNYT538	M	20180131205920	20180131211132
C0D15-1	DIS-VESSEL	MAQC	MCHNGC159	F	20180131205358	20180131205530
C0D15-1	DIS-YARD	MATC	MCHNTC256	F	20180131210446	20180131210618
C0D15-1	LOD-MOVE	MTYT	MCHNYT510	F	20180131205530	20180131210446
C0E16-1	DIS-VESSEL	MTQC	MCHNGC119	F	20180131205357	20180131205529
C0E16-1	DIS-YARD	MTTC	MCHNTC216	F	20180131210300	20180131210432
C0E16-1	DIS-MOVE	MTYT	MCHNYT550	F	20180131205529	20180131210300
C0F17-2	DIS-VESSEL	MAQC	MCHNGC129	M	20180131210505	20180131210637
C0F17-2	DIS-YARD	MATC	MCHNTC286	M	20180131211623	20180131211755
C0F17-2	LOD-MOVE	MTYT	MCHNYT586	M	20180131210637	20180131211623

* Abbrev. of LOD: Loading; DIS: Discharging; F: Full; M: Empty.

Handling categorical variable

- Imagification
  - Pixelization: I.M. Kamal, H. Bae and N.I. Utama et al. / Neurocomputing 374 (2020) 64–76
- Embedding

Transforming event log into image

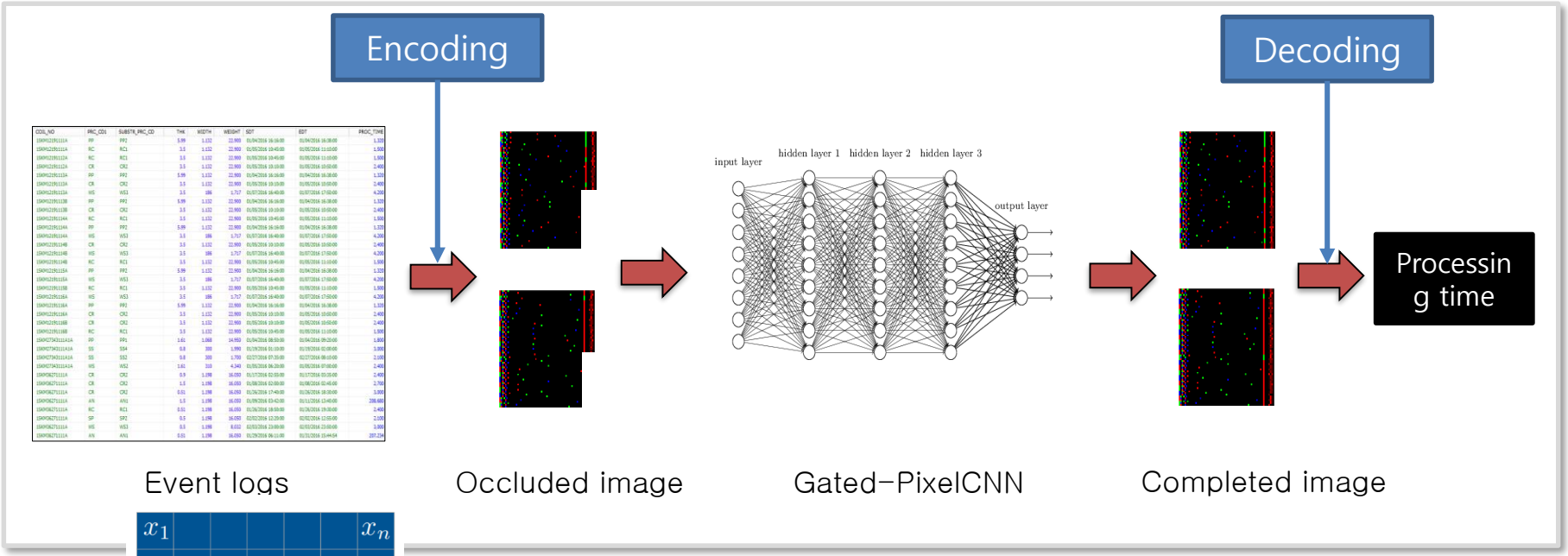


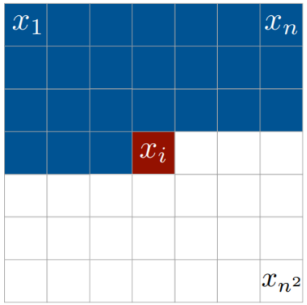
MCHN_TP_CD	JOB_TYPE	FULL_EMTPY	WEIGHT	PROC_TIME
QC	DS-QUAYSIDE	M	2,000	4,589
YT	DS-MOVE	M	2,000	956
TC	DS-YARDSIDE	M	2,000	2
QC	DS-QUAYSIDE	M	2,000	4,717
YT	DS-MOVE	M	2,000	4,848
TC	DS-YARDSIDE	M	2,000	5
YT	DS-MOVE	M	2,200	1,505
TC	DS-YARDSIDE	M	2,200	27
YT	DS-MOVE	F	23,800	2,155
TC	DS-YARDSIDE	F	23,800	1
QC	DS-QUAYSIDE	M	2,200	4,269
YT	DS-MOVE	M	2,200	1,411
TC	DS-YARDSIDE	M	2,200	1
YT	DS-MOVE	F	23,800	1,850
TC	DS-YARDSIDE	F	23,800	4
QC	DS-QUAYSIDE	M	2,200	3,502
YT	DS-MOVE	M	2,200	4,000
TC	DS-YARDSIDE	M	2,200	0
QC	DS-QUAYSIDE	M	2,200	2,286
YT	DS-MOVE	M	2,200	744
TH	DS-YARDSIDE	M	2,200	2

I.M. Kamal, H. Bae, N.I. Utama and Y. Choi., "Data pixelization for predicting completion time of events", Neurocomputing 374 (2020) 64–76

PixelCNN

- Overall framework



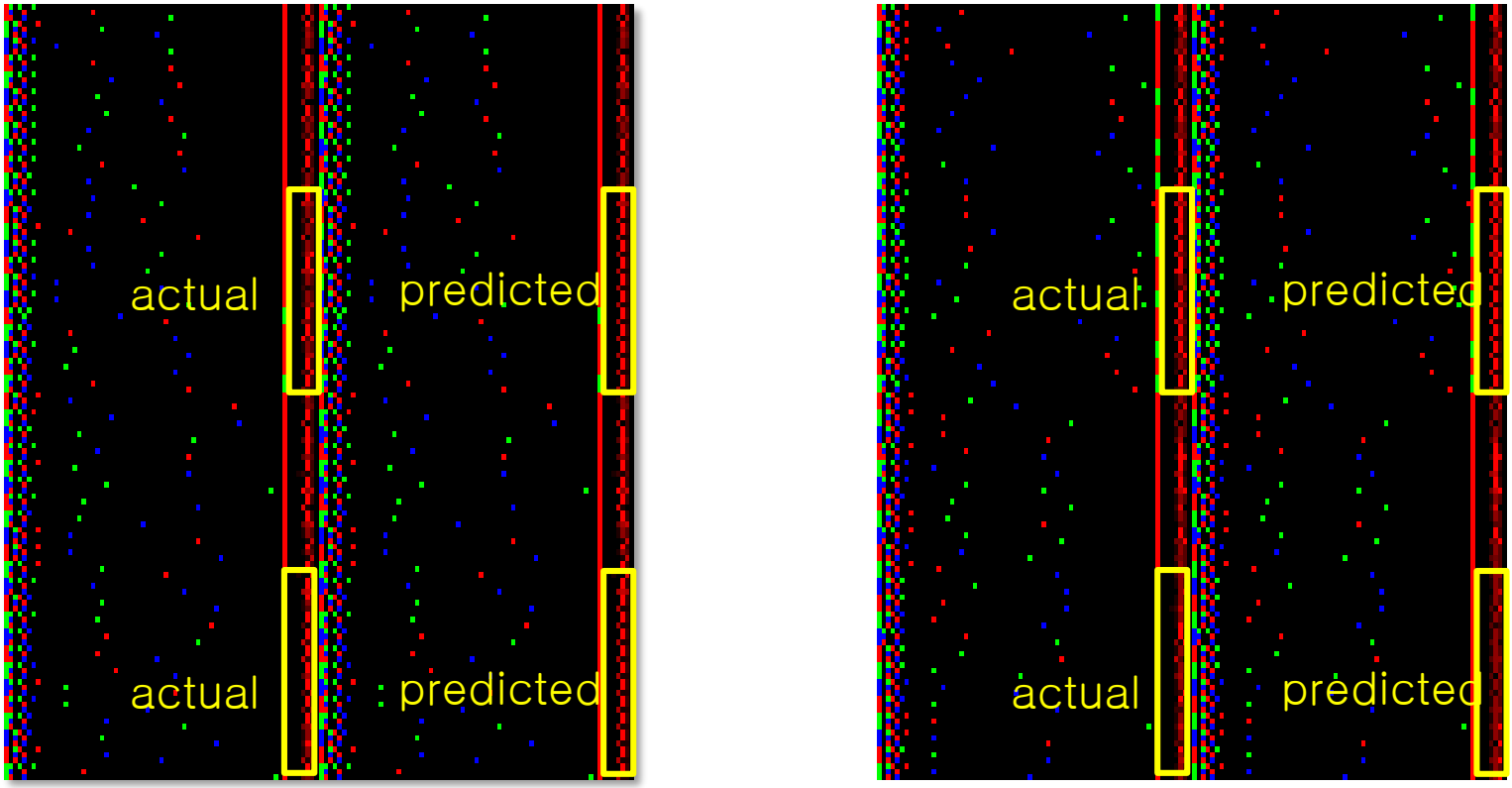


$$p(\mathbf{x}) = \prod_{i=1}^{n^2} p(x_i|x_1, ..., x_{i-1}) \qquad (1)$$

$$p(x_{i,R}|\mathbf{x}_{<i})p(x_{i,G}|\mathbf{x}_{<i}, x_{i,R})p(x_{i,B}|\mathbf{x}_{<i}, x_{i,R}, x_{i,G}) \quad (2)$$

I.M. Kamal, H. Bae, N.I. Utama and Y. Choi., "Data pixelization for predicting completion time of events", Neurocomputing 374 (2020) 64–76

Experimental Result (1)



I.M. Kamal, H. Bae, N.I. Utama and Y. Choi., "Data pixelization for predicting completion time of events", Neurocomputing 374 (2020) 64–76

Performance of pixelization

Accuracy measure comparison among models for event-log SK and MA.

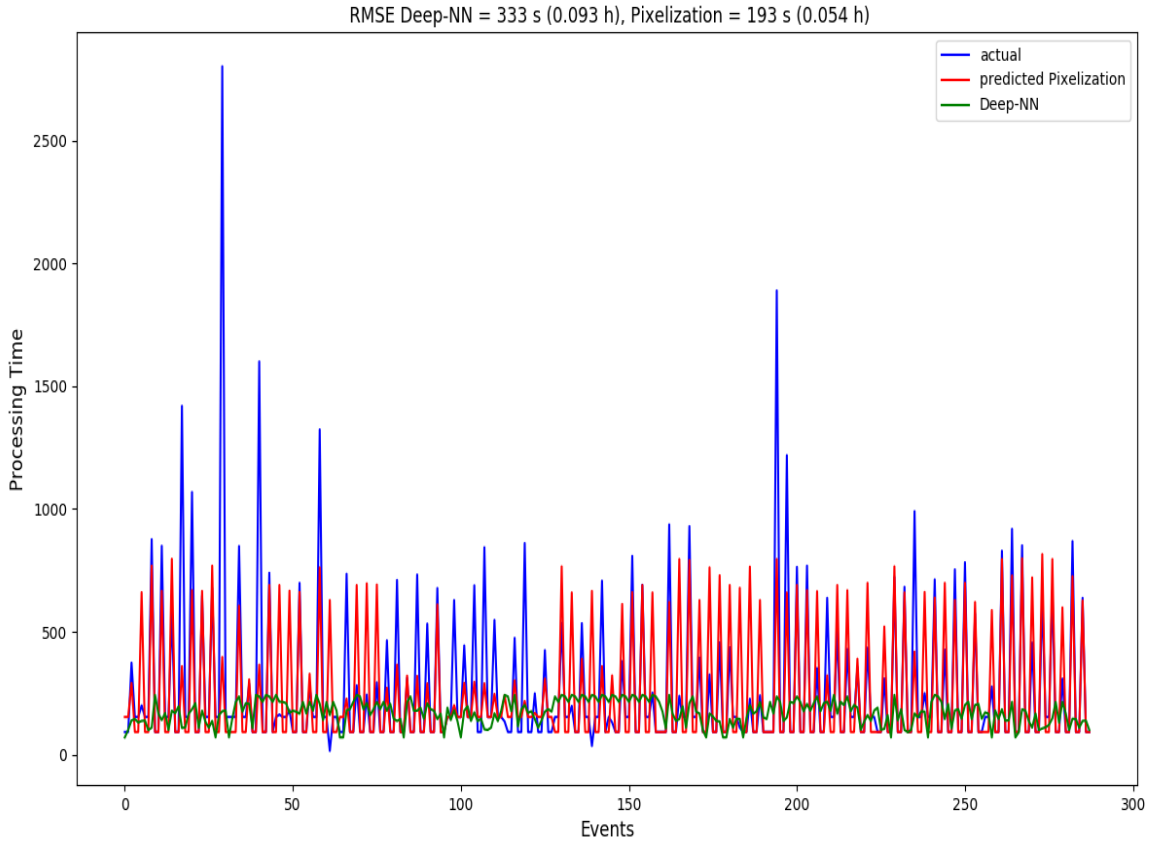
Model	Portion of testing data	Accuracy measure (minute)					
		Event-log SK			Event-log MA		
		RMSE	MAE	MAPE	RMSE	MAE	MAPE
One-hot Encoding w/o machine-id*	10%	4.365936	1.848417	1.763972	5.565939	2.518422	2.963112
	20%	4.377116	1.837501	1.755248	5.523901	2.437655	2.855255
	30%	4.390820	1.833780	1.751794	5.531108	2.533008	2.990233
One-hot Encoding w/ machine-id*	10%	3.372907	1.630705	1.687982	5.356707	2.330789	2.682001
	20%	4.198804	1.563236	1.278410	5.390002	2.262111	2.272288
	30%	3.943855	1.541569	1.363813	5.374356	2.343453	2.345556
Entity Embedding	10%	3.268861	1.575249	1.420940	5.394434	2.345990	2.720099
	20%	4.211195	1.610725	1.229747	5.381998	2.373432	2.529747
	30%	3.956048	1.621329	1.351303	5.376331	2.371112	2.621303
Pixelization (Gated-PixelCNN)	10%	3.109861	1.304340	1.220630	5.128977	2.100324	2.020332
	20%	3.589575	1.310709	1.141329	5.189672	2.011123	2.049256
	30%	3.490048	1.321009	1.163403	5.186998	2.021255	2.063765
Pixelization (C-PixelCNN)	10%	2.709213	1.020120	0.912955	4.891110	1.800982	1.773280
	20%	2.990319	1.010245	0.940078	4.873244	1.810045	1.870126
	30%	2.952744	1.019029	0.925053	4.883309	1.910822	1.794499

* Abbrev. of w/: with; w/o: without.

I.M. Kamal, H. Bae, N.I. Utama and Y. Choi., "Data pixelization for predicting completion time of events", Neurocomputing 374 (2020) 64–76

Experimental Result (2)

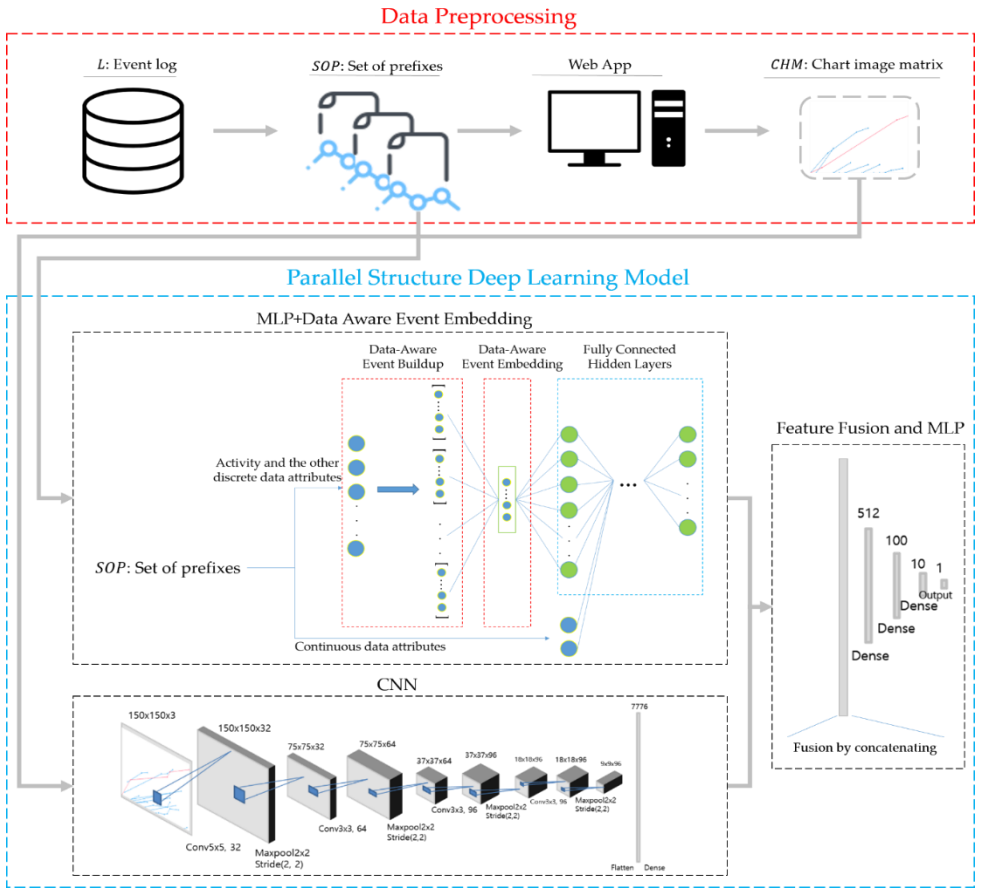
- Comparison between Deep-Neural Network and Our Approach



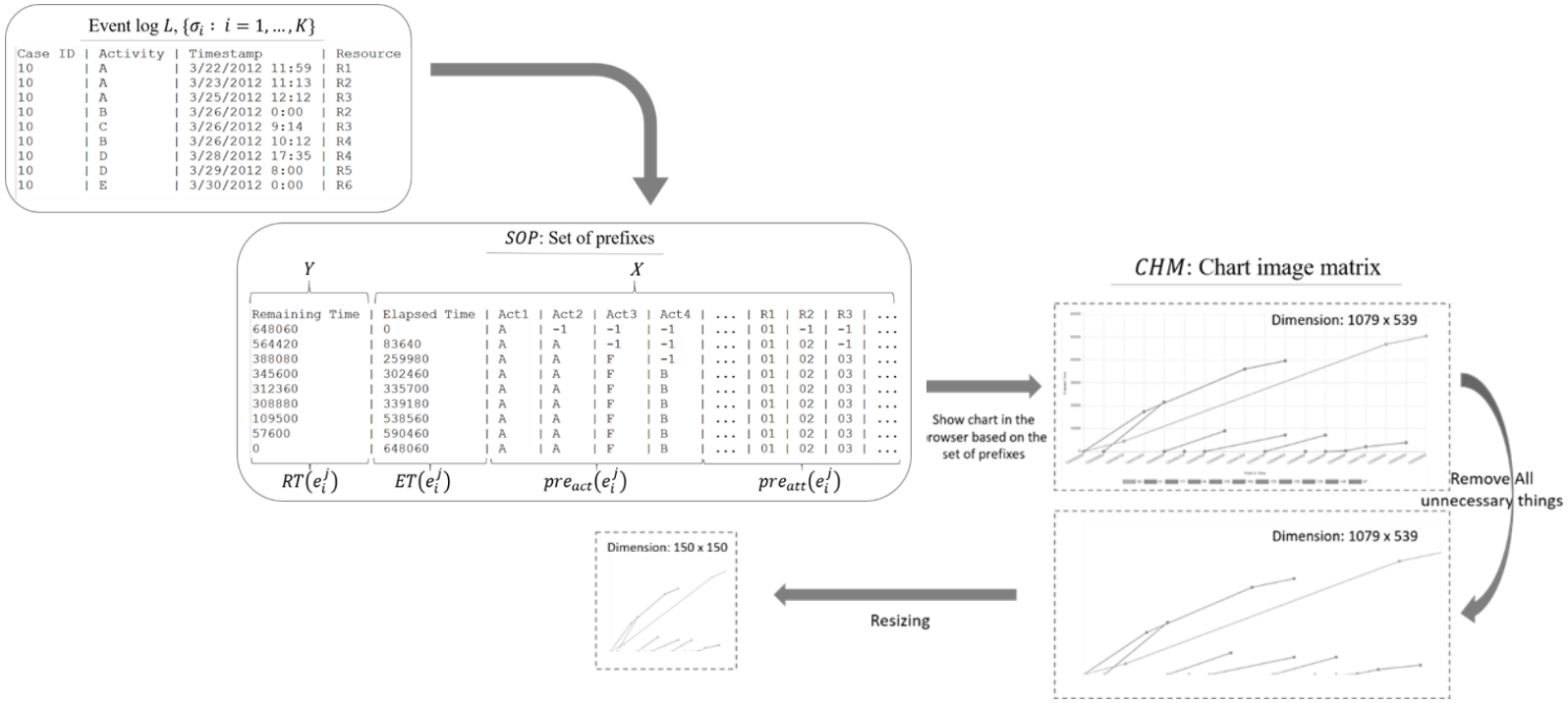
Deep-NN cannot perform well with categorical data !

I.M. Kamal, H. Bae, N.I. Utama and Y. Choi., "Data pixelization for predicting completion time of events", Neurocomputing 374 (2020) 64–76

Combining imagification and embedding



Data preprocessing



DAEE

- Categorical variable in the event log

```
[Activity, Resource, Part Desc, Worker, Report Type, Rework]

[19 10 8 26 2 0]
```

```
[[ 0.858 -0.665  1.181 -0.296 -0.654 -0.861 -2.097 -2.731  0.135  2.526
  -0.456  1.137 -1.579 -3.724  2.048  1.083 -1.023 -1.737 -3.825  2.650
   1.087 -0.014  1.328  3.822 -0.593  0.621 -2.555  2.824  1.095 -1.266
   0.776  2.157  2.569 -0.850 -0.544 -1.083  1.088  0.684 -2.625  2.499
  -0.495 -1.257  0.018 -0.853 -2.817 -2.522 -0.776 -2.500 -0.677 -0.702 ]

 [  2.640 -1.846  0.359  4.170  0.465 -0.128 -0.670  0.844  1.706 -0.605
   0.410  0.125  1.111  0.949  0.515  0.003 -0.527  3.609  1.285 -1.829
   0.995  0.722 -4.111  1.215 -0.490  4.261  1.488  0.349 -0.995 -1.499
   1.192  0.953  0.630  0.793  0.600  0.018 -1.735  0.525 -0.529 -0.463
   1.384 -0.572 -0.753 -0.244 -1.124  2.562  0.401 -0.355  0.879  0.429 ]

 [ -0.149  5.135  1.492  3.464 -1.714 -0.567 -2.918  2.406  1.316 -1.516
  -4.051  1.865 -2.778  2.044 -2.975  3.132 -0.008  4.617  2.844 -1.668
   1.905  1.619 -2.842  1.475 -2.671  1.216  0.945  1.249  3.096 -1.097
   1.055 -2.223  1.642  2.951  2.870  2.066  2.615 -2.310 -0.617 -2.941
   2.288 -3.390 -3.172 -1.855 -1.235 -0.822  2.962  0.327  1.569 -0.804 ]

 [  0.457 -1.216 -3.054 -0.942  1.802  1.791 -2.059  1.643  2.295 -1.407
  -0.844 -0.718  1.046  0.677  1.421  0.988  0.635 -0.008  1.447  0.034
   1.907 -1.936  0.381 -0.286 -0.770 -0.617 -0.631 -3.067 -1.653 -0.820
  -1.955 -0.137 -1.942  0.751  0.688  0.742 -0.692  0.248  2.204 -2.250
   0.959 -0.358 -0.436  2.906  0.731  0.858  1.114 -2.422 -2.059  1.397 ]

 [ -1.741 -1.688  0.649  4.318  0.603 -1.362  2.042 -2.213  0.431  2.795
   0.125  2.831  0.587  1.550 -0.452  1.214 -0.552 -1.553 -0.405 -2.816
   2.422  3.589 -3.443  2.098  1.389  1.686 -1.838  2.373  1.125 -2.067
   2.149 -2.830  3.529  1.253 -1.899  0.636 -0.478 -3.070 -1.820 -0.672
   1.119 -1.825  1.013 -2.319 -3.157 -0.998  0.312  0.050  3.514 -2.713 ]

 [ -0.007  0.022  0.009 -0.001 -0.040 -0.012 -0.001 -0.074 -0.001  0.008
   0.004  0.009 -0.004 -0.006 -0.002  0.299 -0.002 -0.013 -0.009  0.030
   0.000 -0.000  0.008 -0.004  0.006 -0.005 -0.009  0.010  0.001  0.086
   0.030  0.060  0.001 -0.001  0.000 -0.004  0.004 -0.008 -0.007  0.003
   0.000 -0.000  0.007 -0.011 -0.008 -0.002 -0.006 -0.003  0.008 -0.002 ]]
```

Performance

Accuracy (MAE) in units of minutes

Iteration	MAE (minutes)			
	DA-LSTM	MLP+EE	CNN+EE	CNN+DAEE
1st	5221.646	5362.156	4416.111	3617.575
2nd	4945.532	4973.759	4749.251	3793.084
3rd	7191.880	5578.992	4853.482	3301.337
4th	7176.215	5654.634	4910.471	3641.002
5th	5988.119	5096.931	4935.983	3232.011

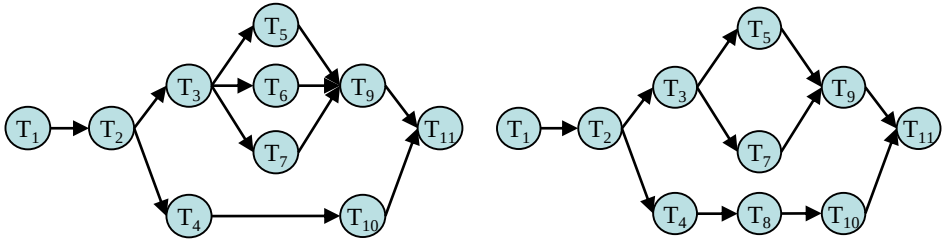
Mean training duration and MAE in units of minutes across five iterations of CV

	DA-LSTM	MLP+EE	CNN+EE	CNN+DAEE
MAE	6104.678	5333.294	4773.060	3517.002
Training duration	861.400	83.600	155.400	123.000

Unsupervised Learning

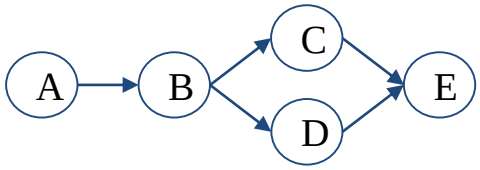
How to measure the distance between process models

- How similar each other?



Business Process Reference Model

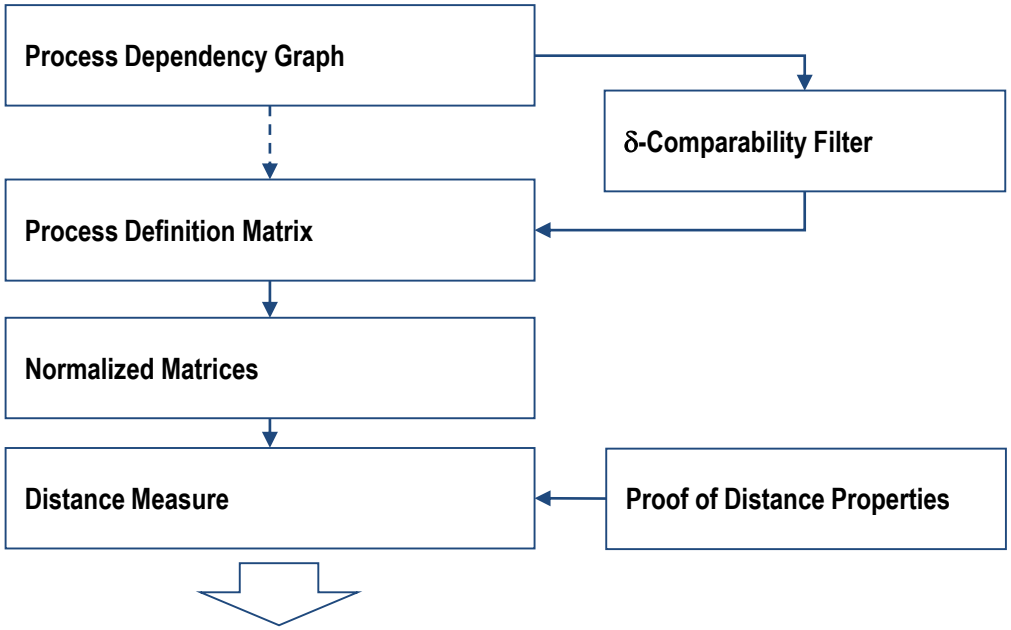
- Definition 1 (Dependency Graph, *DG*)
  - A dependency graph *DG* is defined by a binary tuple <*DN, DE*>, where
  - $DN = \{nd_1, nd_2, ..., nd_n\}$ is a finite set of activity nodes where $n \geq 1$.
  - $DE = \{e_1, e_2, ..., e_m\}$ is a set of edges, $m \geq 0$. Each edge is of the form $nd_i \rightarrow nd_j$. ■
  - Example



- *DN* = {*A, B, C, D, E*}
- *DE* = {*A* →*B, B* →*C, B* →*D, C* →*E, D* →*E*}

Overall Flow of Distance Measure Derivation

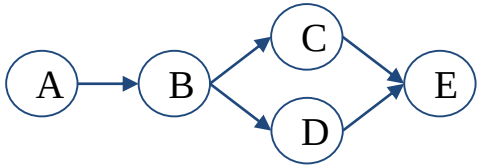
- Input: Set of Processes + Template



- Output: Ranked List of Processes

Process Definition Matrix

- Numerical Representation of Graph



  - Relation between activities → 2 dimensions
  - 2 dimensional matrix is needed

M	**TO**					
FROM		**A**	**B**	**C**	**D**	**E**
	A					
	B					
	C					
	D					
	E					

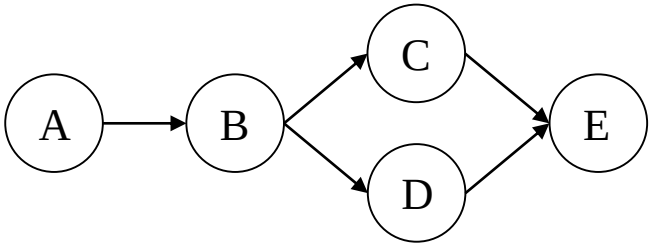
Process Definition Matrix

- Definition 4 (Process Definition Matrix, *M*)
  - Let $g = (DN, DE)$ be a dependency graph with $|DN| = n$ nodes. A process network matrix *M* of *g* is *n*-by-*n* matrix with *n* rows and *n* columns, and each row is named after the node name.
  - Let $M_g(i,j)$ denote the value of the ith row and the jth column in *M*, $1 \le i, j \le n$. We define $M_g(i,j)$ as follows:

$$M_g(i,j) = \begin{cases} 1 & if \ \exists nd_i, nd_j \in DN \ such \ that \ (nd_i, nd_j) \in DE \\ 0 & else \end{cases}$$

Process Definition Matrix

- An Example of process definition matrix



Transform

M	TO					
FROM		A	B	C	D	E
	A	0	1	0	0	0
	B	0	0	1	1	0
	C	0	0	0	0	1
	D	0	0	0	0	1
	E	0	0	0	0	0

Comparison Matrices

- Definition 2 (Identical dependency graphs)
  - Let $DG_1 = (DN_1, DE_1)$ and $DG_2 = (DN_2, DE_2)$ be two dependency graphs. We say that DG_1 and DG_2 are identical if the two graphs have the same set of nodes and the same set of edges.
  - i) $DN_1 =_{Set} DN_2$
  - ii) $DE_1 =_{Set} DE_2$ ■
  - Example

Comparison Matrices

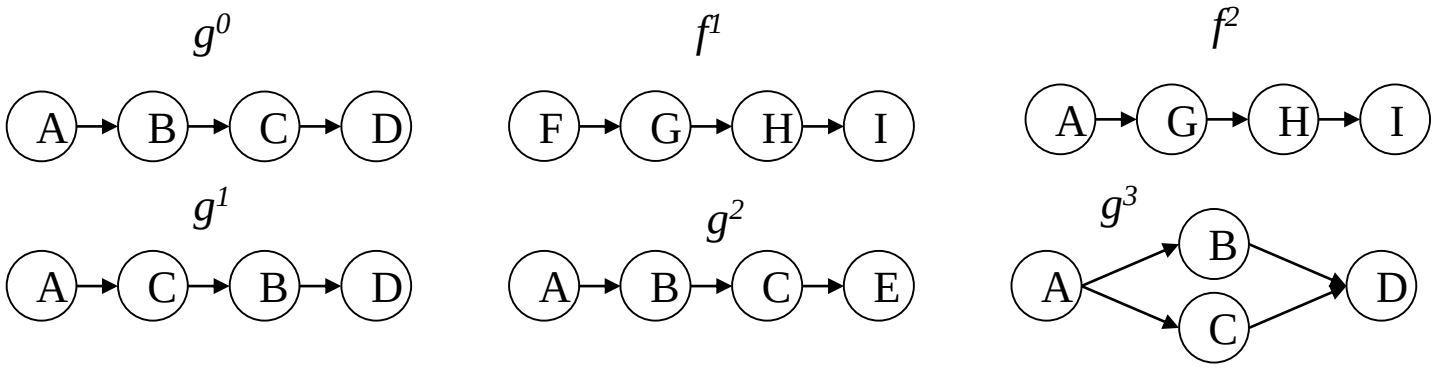
- Definition 3 (δ-Comparability of *DG*)
  - Let $DG_1 = (DN_1, DE_1)$ and $DG_2 = (DN_2, DE_2)$ be two dependency graphs, and δ be a user-defined control threshold. We say that DG_1 and DG_2 are *δ-comparable* if the condition

$$\frac{|DN_1 \cap DN_2|}{|DN_1 \cup DN_2|} \geq \delta \quad \text{holds, where} \quad 0 < \delta \leq 1 \quad \blacksquare$$

  - The degree of common activities
  - The value 0 means that there is no common node between two graphs, i.e., $DN_1 \cap DN_2 \neq \varnothing$.
  - The value of $\frac{|DN_1 \cap DN_2|}{|DN_1 \cup DN_2|}$ of identical dependency graphs is 1

Comparison Matrices

- Examples of δ-Comparability δ=0.5



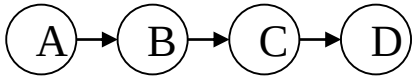
- g^0 and f^1 are not comparable because there is no common node in the two graphs
- g^0 and f^2 are not comparable because the number of common nodes is only one but the number of total nodes is 7, that is $\frac{|DN_1 \cap DN_2|}{|DN_1 \cup DN_2|} = \frac{1}{7} < 0.5$
- g^0 and g^1 are δ-comparable because all of the nodes in both graphs are common nodes.
- g^0 and g^2 are δ-comparable because there are 3 common nodes and the total number of nodes is 5, thus the two graphs satisfy the δ-comparability condition

$$\frac{|DN_1 \cap DN_2|}{|DN_1 \cup DN_2|} = \frac{3}{5} \geq 0.5$$

Comparison Matrices

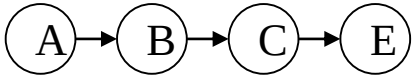
- We can represent graph by using matrix now.
- But, when the activities do not match completely, we cannot handle two matrices to compare.

g^0



NM_1	TO					
FROM		A	B	C	D	
	A	0	1	0	0	
	B	0	0	1	0	
	C	0	0	0	1	
	D	0	0	0	0	

g^2



NM_2	TO					
FROM		A	B	C		E
	A	0	1	0		0
	B	0	0	1		0
	C	0	0	0		1
	E	0	0	0		0

Comparison Matrices

- Definition 5 (Normalized Matrix, *NM*)
  - Let $DG_1 = (DN_1, DE_1)$ and $DG_2 = (DN_2, DE_2)$ be two dependency graphs. Let NM_1 and NM_2 denote the normalized matrices for DG_1 and DG_2 respectively. We generate NM_1 and NM_2 from DG_1 and DG_2 as follows.
  - i) The number of rows and columns are computed by

$$m = \left| DN_1 \cup DN_2 \right|$$

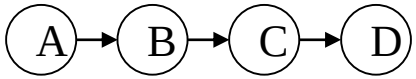
  - ii) Let $DN_1 \bigcup DN_2 = \{A_1, A_2, ..., A_m\}$
  - iii) Let $NM_1(i,j)$ *and* $NM_2(i,j)$ denote the value of the i^{th} row and the j^{th} column in NM_1, NM_2

$$NM_1(i,j) = \begin{cases} 1 & \text{if } (A_i, A_j) \in DE_1 \\ 0 & \text{otherwise} \end{cases} \qquad NM_2(i,j) = \begin{cases} 1 & \text{if } (A_i, A_j) \in DE_2 \\ 0 & \text{otherwise} \end{cases}$$

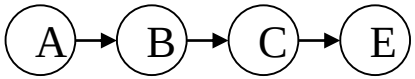
Comparison Matrices

- An example of normalized matrices

g^0



g^2



NM_1	TO					
FROM		A	B	C	D	E
	A	0	1	0	0	0
	B	0	0	1	0	0
	C	0	0	0	1	0
	D	0	0	0	0	0
	E	0	0	0	0	0

NM_2	TO					
FROM		A	B	C	D	E
	A	0	1	0	0	0
	B	0	0	1	0	0
	C	0	0	0	0	1
	D	0	0	0	0	0
	E	0	0	0	0	0

Distance-based Process Similarity Measures

- Difference between two normalized matrices

$$NM_1 - NM_2 = \begin{bmatrix} & A & B & C & D & E \\ A & 0 & 0 & 0 & 0 & 0 \\ B & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & 1 & -1 \\ D & 0 & 0 & 0 & 0 & 0 \\ E & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- Need a degree of discrepancy → Distance
  - Sum of square of all elements in NM_1-NM_2

Distance-based Process Similarity Measures

- Sum of Square of the Elements in NM_1-NM_2

$$(NM_1 - NM_2)(NM_1 - NM_2)^T = \begin{bmatrix} & A & B & C & D & E \\ A & 0 & 0 & 0 & 0 & 0 \\ B & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & 1 & -1 \\ D & 0 & 0 & 0 & 0 & 0 \\ E & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} & A & B & C & D & E \\ A & 0 & 0 & 0 & 0 & 0 \\ B & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & 0 & 0 \\ D & 0 & 0 & 1 & 0 & 0 \\ E & 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} & A & B & C & D & E \\ A & 0 & 0 & 0 & 0 & 0 \\ B & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 1^2 + (-1)^2 & 0 & 0 \\ D & 0 & 0 & 0 & 0 & 0 \\ E & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

➔ 2

Distance-based Process Similarity Measures

- In Linear Algebra

$$A \times A^T = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & & & & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix} \times \begin{bmatrix} a_{11} & a_{21} & a_{31} & \dots & a_{n1} \\ a_{12} & a_{22} & a_{32} & \dots & a_{n2} \\ a_{13} & a_{23} & a_{33} & \dots & a_{n3} \\ \dots & & & & \dots \\ a_{1n} & a_{2n} & a_{3n} & \dots & a_{nn} \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{j=1}^{n} a_{1j}^2 & \dots & \dots & \dots & \dots \\ \dots & \sum_{j=1}^{n} a_{2j}^2 & \dots & \dots & \dots \\ \dots & \dots & \sum_{j=1}^{n} a_{3j}^2 & \dots & \dots \\ \dots & & & & \dots \\ \dots & \dots & \dots & \dots & \sum_{j=1}^{n} a_{nj}^2 \end{bmatrix}$$

[Elementary Linear Algebra: Applications, Howard Anton and Chris Rorres, 1994]

Distance-based Process Similarity Measures

- Definition 6 (Trace of Matrix, *tr*)
  - The trace of an m-by-m square matrix *A* is defined as the sum of the elements on the main diagonal (the diagonal from the upper left to the lower right) of *A*, i.e.,

$$tr(A) = A_{1,1} + A_{2,2} + \ldots + A_{m,m}$$

  where A_{ij} represents the (i, j)th element of *A* ■
- Definition 7 (Inner Product of Matrix)
  - For an m-by-n matrix *A* with complex (or real) entries, we have

$$< A, A^T >= tr(A \times A^T) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^{\ 2} \geq 0$$

  with the trace of *A* equals to 0 only if *A*=0. ■

Distance-based Process Similarity Measures

- Definition 8 (Dependency Difference Metric, *d*)
  - Let $DG_1 = (DN_1, DE_1)$ and $DG_2 = (DN_2, DE_2)$ be two dependency graphs. Let NM_1 and NM_2 be the normalized matrix of DG_1 and DG_2 respectively.
  - We define the symmetric difference metric on graphs DG_1 and DG_2 by the trace of the difference matrix of NM_1 and NM_2 as follows:

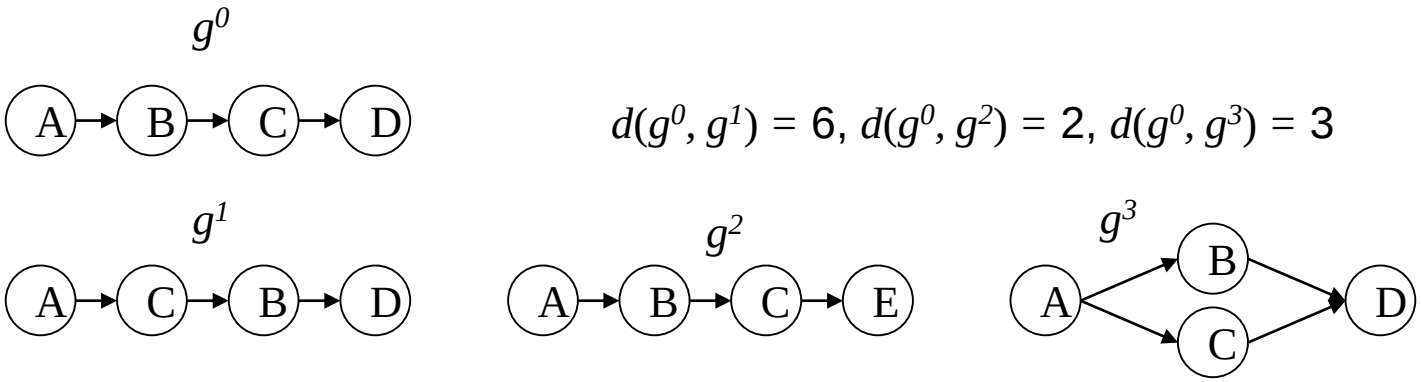
$$d(DG_1, DG_2) = tr[(NM_1 - NM_2) \times (NM_1 - NM_2)^T]$$

where *tr*[·] denotes the trace of a matrix, i.e., the sum of the diagonal elements. ■

  - That is, new difference metric *d* is inner product of NM_1-NM_2

Distance-based Process Similarity Measures

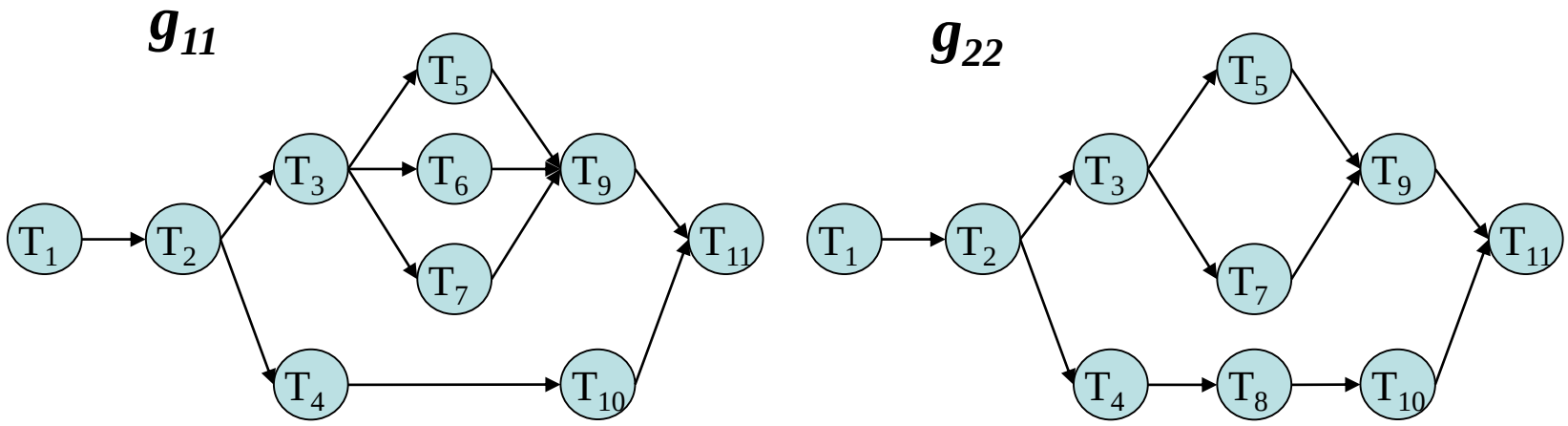
- Examples of dependency distance measurements



  - The distance of dependency between g^0 and g1 is 6, the distance of g^0 and g^2 is 2, and the distance of g^0 and g^3 is 3.
  - This means that g^0 and g^2 are the most similar, which is intuitively correct because the first three activities are in the same sequence but only the last activity is different.
  - g^0 and g^1 are mostly different because the sequence of the activities in g^1 is quite different from g^0.
  - In this dependency distance measure, the parallel execution in g^3 is not considered important and only the precedence relationships and common activities are considered important.

Extended Example

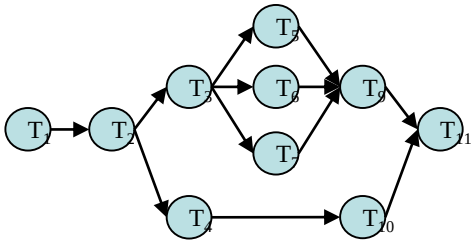
- Two graphs g_{11} and g_{22}



  - $d(g_{11}, g_{22})$ =?

Extended Example

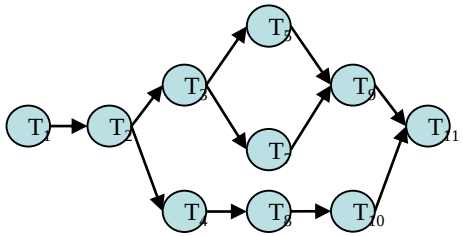
- Process Definition Matrix – M_{11}



		TO									
		T1	T2	T3	T4	T5	T6	T7	T9	T10	T11
FROM	T1	0	1	0	0	0	0	0	0	0	0
	T2	0	0	1	1	0	0	0	0	0	0
	T3	0	0	0	0	1	1	1	0	0	0
	T4	0	0	0	0	0	0	0	0	1	0
	T5	0	0	0	0	0	0	0	1	0	0
	T6	0	0	0	0	0	0	0	1	0	0
	T7	0	0	0	0	0	0	0	1	0	0
	T9	0	0	0	0	0	0	0	0	0	1
	T10	0	0	0	0	0	0	0	0	0	1
	T11	0	0	0	0	0	0	0	0	0	0

Extended Example

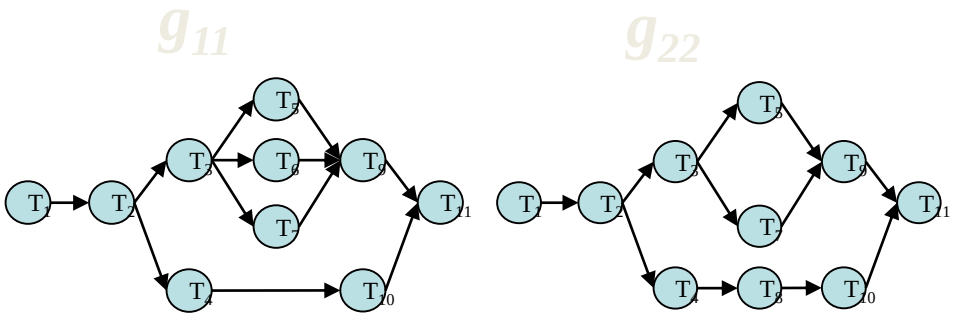
- Process Definition Matrix- M_{22}



FROM \ TO	T1	T2	T3	T4	T5	T7	T8	T9	T10	T11
T1	0	1	0	0	0	0	0	0	0	0
T2	0	0	1	1	0	0	0	0	0	0
T3	0	0	0	0	1	1	0	0	0	0
T4	0	0	0	0	0	0	1	0	0	0
T5	0	0	0	0	0	0	0	1	0	0
T7	0	0	0	0	0	0	0	1	0	0
T8	0	0	0	0	0	0	0	0	1	0
T9	0	0	0	0	0	0	0	0	0	1
T10	0	0	0	0	0	0	0	0	0	1
T11	0	0	0	0	0	0	0	0	0	0

Extended Example

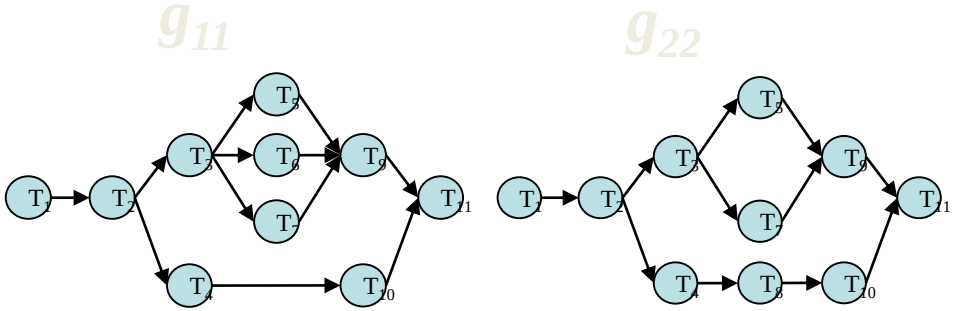
- Normalized Matrix – NM_{11}



	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
T1	0	1	0	0	0	0	0	0	0	0	0
T2	0	0	1	1	0	0	0	0	0	0	0
T3	0	0	0	0	1	1	1	0	0	0	0
T4	0	0	0	0	0	0	0	0	0	1	0
T5	0	0	0	0	0	0	0	0	1	0	0
T6	0	0	0	0	0	0	0	0	1	0	0
T7	0	0	0	0	0	0	0	0	1	0	0
T8	0	0	0	0	0	0	0	0	0	0	0
T9	0	0	0	0	0	0	0	0	0	0	1
T10	0	0	0	0	0	0	0	0	0	0	1
T11	0	0	0	0	0	0	0	0	0	0	0

Extended Example

- Normalized Matrix – NM_{22}



	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
T1	0	1	0	0	0	0	0	0	0	0	0
T2	0	0	1	1	0	0	0	0	0	0	0
T3	0	0	0	0	1	0	1	0	0	0	0
T4	0	0	0	0	0	0	0	1	0	0	0
T5	0	0	0	0	0	0	0	0	1	0	0
T6	0	0	0	0	0	0	0	0	0	0	0
T7	0	0	0	0	0	0	0	0	1	0	0
T8	0	0	0	0	0	0	0	0	0	1	0
T9	0	0	0	0	0	0	0	0	0	0	1
T10	0	0	0	0	0	0	0	0	0	0	1
T11	0	0	0	0	0	0	0	0	0	0	0

Extended Example

- Calculation of Distance – NM_{11} - NM_{22}

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
T1	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0
T3	0	0	0	0	0	1	0	0	0	0	0
T4	0	0	0	0	0	0	0	-1	0	1	0
T5	0	0	0	0	0	0	0	0	0	0	0
T6	0	0	0	0	0	0	0	0	1	0	0
T7	0	0	0	0	0	0	0	0	0	0	0
T8	0	0	0	0	0	0	0	0	0	-1	0
T9	0	0	0	0	0	0	0	0	0	0	0
T10	0	0	0	0	0	0	0	0	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0

Extended Example

- Calculation of Distance – $(NM_{11} - NM_{22})^T$

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
T1	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0
T3	0	0	0	0	0	0	0	0	0	0	0
T4	0	0	0	0	0	0	0	0	0	0	0
T5	0	0	0	0	0	0	0	0	0	0	0
T6	0	0	1	0	0	0	0	0	0	0	0
T7	0	0	0	0	0	0	0	0	0	0	0
T8	0	0	0	-1	0	0	0	0	0	0	0
T9	0	0	0	0	0	1	0	0	0	0	0
T10	0	0	0	1	0	0	0	-1	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0

Extended Example

- Calculation of Distance – $(NM_{11}\text{-}\ NM_{22}) \times (NM_{11}\ \text{-}\ NM_{22})^T$

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
T1	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0
T3	0	0	1	0	0	0	0	0	0	0	0
T4	0	0	0	2	0	0	0	-1	0	0	0
T5	0	0	0	0	0	0	0	0	0	0	0
T6	0	0	0	0	0	1	0	0	0	0	0
T7	0	0	0	0	0	0	0	0	0	0	0
T8	0	0	0	-1	0	0	0	1	0	0	0
T9	0	0	0	0	0	0	0	0	0	0	0
T10	0	0	0	0	0	0	0	0	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0

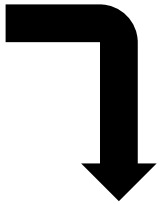
$$tr[\ (NM_{11} - NM_{22})\ x\ (NM_{11} - NM_{22})^T\ ] = 5$$

Applications of distance matrix

- K-means
- KNN
- Outlier process detection

Trace clustering

Block	Event log	Description
Block 1	{S, G}	S: Sub Assembly U: Unit Assembly G: Grand Assembly
Block 2	{S, U, G}	
Block 3	{G}	
Block 4	{S, G}	
Block 5	{S, U, G}	



$$sim(T_x, T_y) = \alpha \times sim_{act}(T_x, T_y) + \beta \times sim_{tran}(T_x, T_y) + \gamma \times sim_{actpat}(T_x, T_y)$$

$$\alpha + \beta + \gamma = 1,\ 0 \leq \alpha, \beta, \gamma \leq 1$$

Block	Activity profile			Transition profile			Activity pattern profile			
	S	U	G	S-G	S-U	U-G	S-G	S	G	S-U-G
Block 1	1	0	1	1	0	0	1/2	1/2	1/2	0
Block 2	1	1	1	0	1	1	0	1/3	1/3	1/3
Block 3	0	0	1	0	0	0	0	0	1/1	0
Block 4	1	0	1	1	0	0	1/2	1/2	1/2	0
Block 5	1	1	1	0	1	1	0	1/3	1/3	1/3

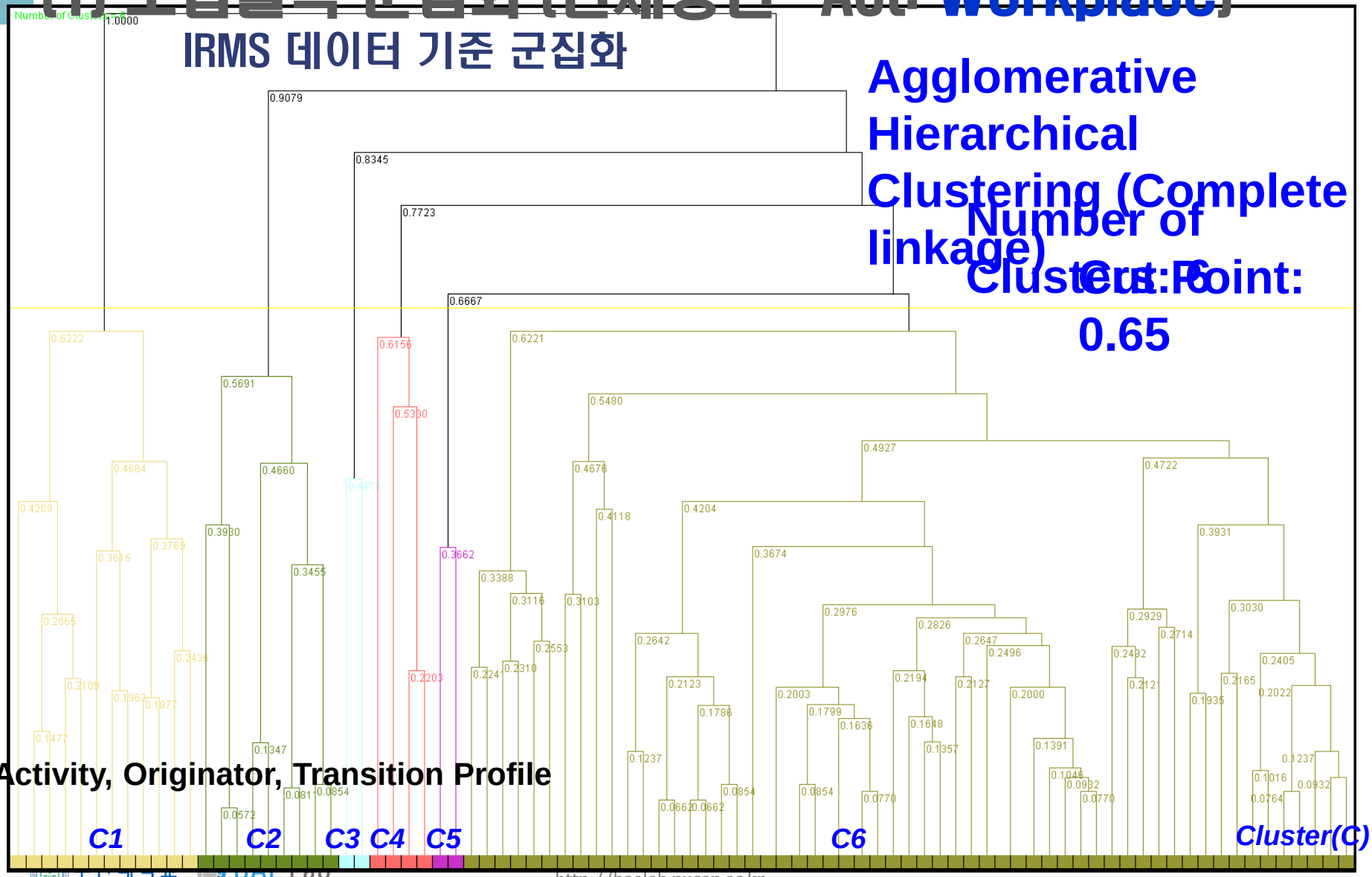
(1) 조립블록 군집화 (전체송선 –Act: Workplace)

IRMS 데이터 기준 군집화

Agglomerative Hierarchical Clustering (Complete linkage)

Number of Cluster: 6

Cut Point: 0.65

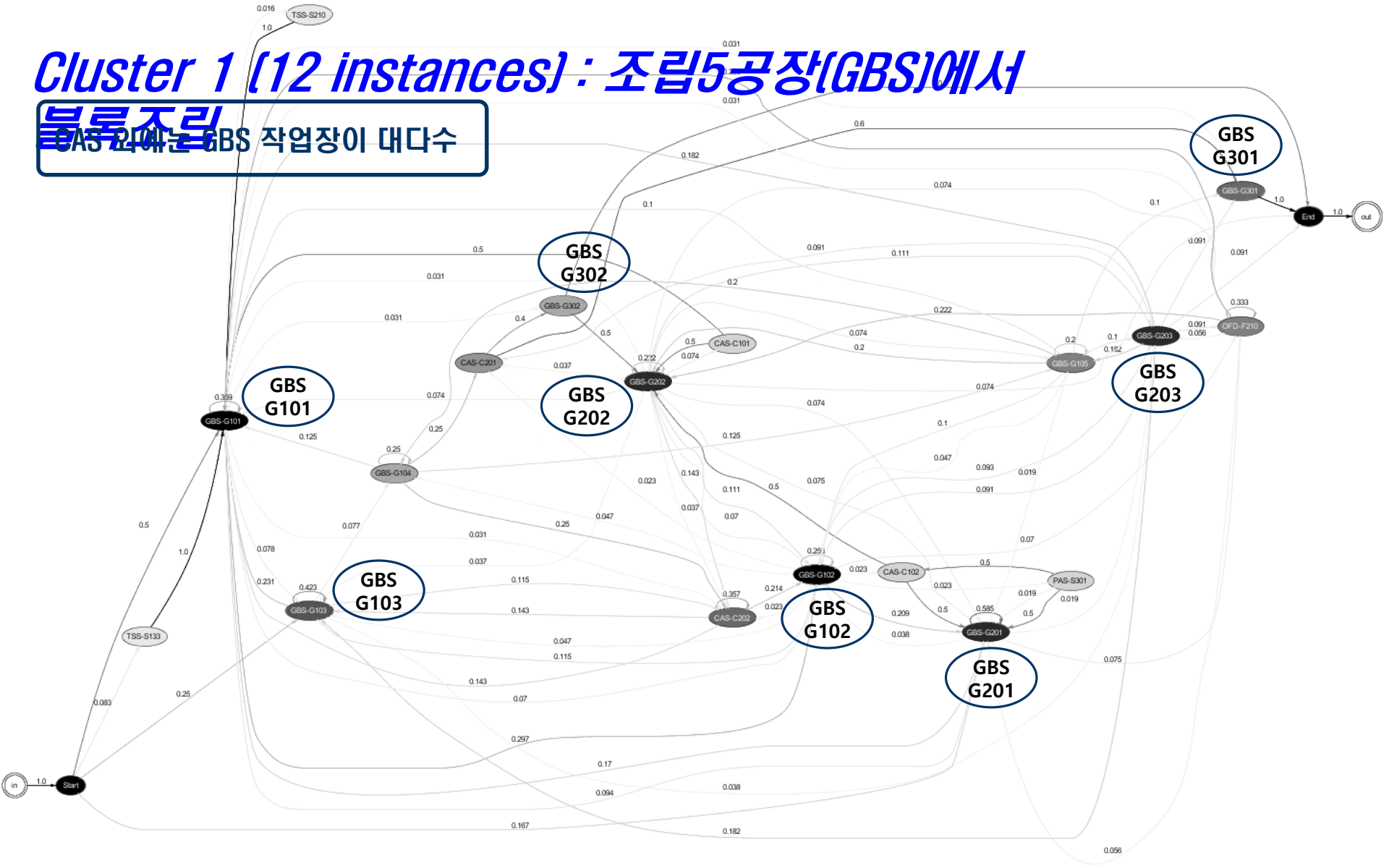


Activity, Originator, Transition Profile

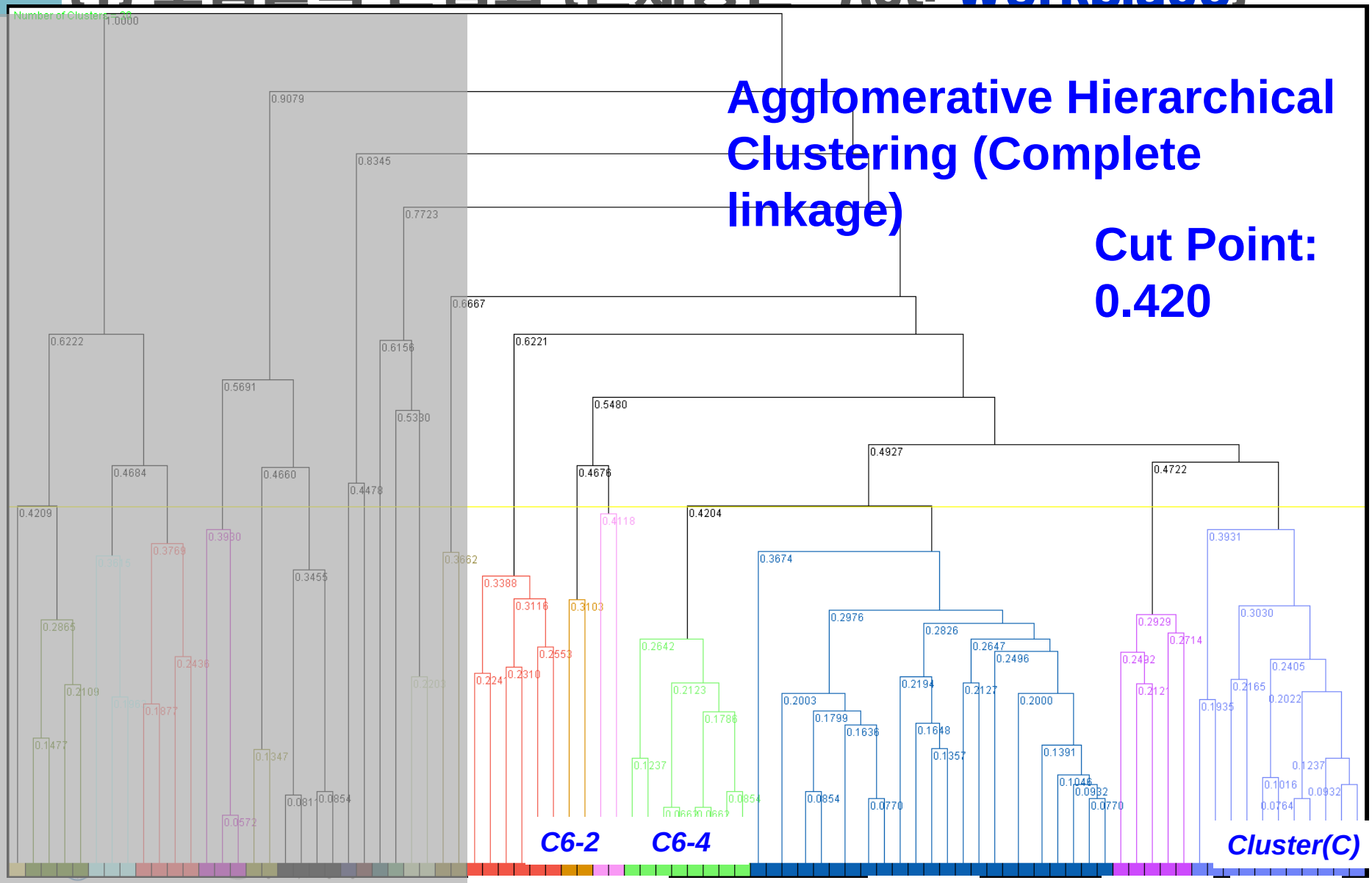
(1) 조립블록 군집화 (전체송선 –Act: Workplace)

Cluster 1 (12 instances) : 조립5공장(GBS)에서 블록조립

CAS 외에는 GBS 작업장이 대다수



(1) 조립블록 군집화 (전체송선 –Act: Workplace)



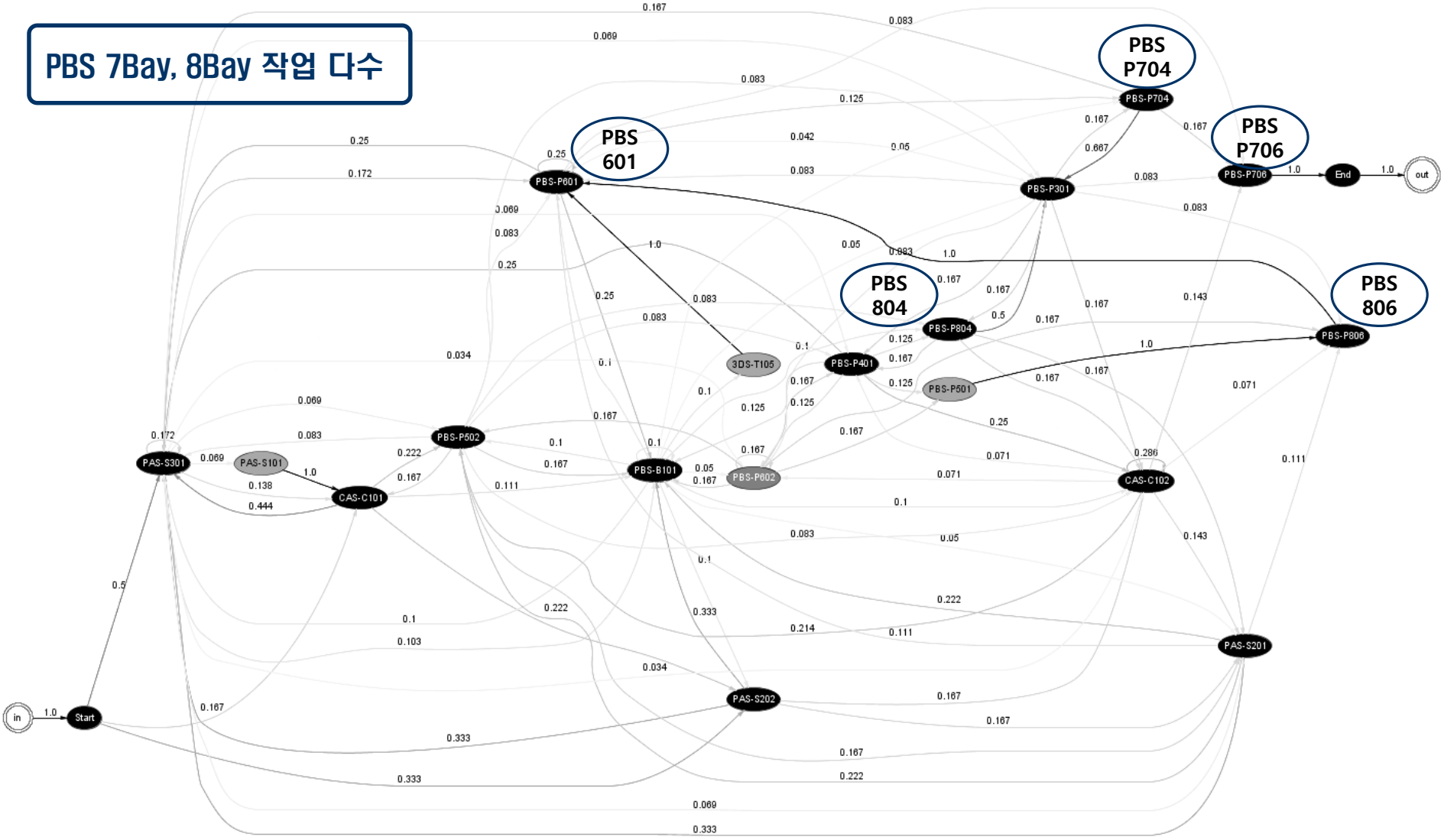
Agglomerative Hierarchical Clustering (Complete linkage)

Cut Point: 0.420

(1) 조립블록 군집화 (전체송선 –Act: Workplace)

Cluster 6-1 (6 instances) : PBS 대조 블록조립

PBS 7Bay, 8Bay 작업 다수



(1) 조립블록 군집화 (IRMS 기준) – 결과

번호	군집1	군집2	블록수	군집이름	블록
1	C1		12	조립5공장(GBS)에서 블록 조립	EM0-209, EM0-258, EM0-248, EM0-259, EM0-249, EM0-112, EM0-131, EM0-121, EM0-642,EM0-652, EM0-238, EM0-228
2	C2		9	조립4공장(OFD)에서 소조 작업 후 조립 4공장 또는 조립5공장(GBS)에서 대조 작업	DM9-115, EM0-171, EM0-161, EM0-261, EM0-271, EM0-638, EM0-628, DM9-156, DM9-146
3	C3		2	(정하지 못함)	EM0-803, EM0-213
4	C4		4	CAS, PAS에서 콤프, 판작업 후 조립3공장(NPS)에서 중,대조 작업	DM9-107, EM0-106, DM9-113, EM0-115
5	C5		2	조립2공장(3DS)에서 대조 작업 (CAS에서 콤프 제작)	DM9-105, EM0-105
6	C6	C6-1	6	조립1공장(PBS)에서 대조 작업	DM9-232, DM9-222, DM9-234, DM9-233, DM9-224, DM9-223
7		C6-2	2	조립2공장(3DS)에서 대조 작업 (CAS, TSS, PAS에서 콤프/소조/판 제작)	DM9-121, DM9-131
8		C6-3	2	조립2공장(3DS)에서 대조 작업 (CAS, TSS, PAS, PBS에서 콤프/소조/판/중조 제작)	EM0-212, EM0-211
9		C6-4	8	조립1공장(PBS)에서 대조 블록 조립	DM9-634,DM9-624, DM9-632, DM9-622, DM9-633, DM9-623, DM9-631, DM9-621
10		C6-5	23	(정하지 못함)	DM9-221, EM0-175, EM0-165, EM0-156, EM0-146, DM9-126, DM9-136, EM0-126, EM0-136, DM9-104, EM0-110, EM0-104, EM0-102, DM9-231, DM9-117, DM9-114, EM0-176, EM0-166, EM0-101, DM9-102, DM9-101, DM9-103, EM0-103
11		C6-6	5	3DS 또는 TSS 대조 블록 조립 작업	EM0-802, DM9-801, EM0-801, DM9-803, EM0-804
12		C6-7	11	3DS/TSS/CAS/PAS에서 소중조 작업 후 특수선 또는 OFD에서 대조 블록 조립	EM0-164, EM0-174, EM0-193, EM0-183, EM0-154 EM0-144, EM0-195, EM0-185, EM0-116, EM0-194 EM0-184

(1) 조립블록 군집화 (배량 케이스 기준) – 결과

Mining

번호	배량 케이스	블록수	군집특징	블록
1	3D6	19	대조 L/T이 7일이 적용되는 3DS 일반 BLK, 모든 본 송선을 3DS 자체 제작	DM9-121, DM9-803, DM9-801, DM9-105, EM0-195, EM0-193, EM0-164, EM0-185, EM0-183, EM0-804, EM0-803, EM0-802, EM0-801, EM0-174, EM0-213, EM0-212, EM0-211, EM0-105, DM9-131
2	PB4	18	PBS LINE BLK(T/BOX BLK)	DM9-126, EM0-156, DM9-634, DM9-633, DM9-632, DM9-631, EM0-126, DM9-624, DM9-623, DM9-622, DM9-621, EM0-176, EM0-175, EM0-136, DM9-136, EM0-166, EM0-165, EM0-146
3	GB6	17	대조 L/T이 7일이 적용되는 GBS일반 BLK, 일반형상의 Block으로 모든 본 송선 GBS 자체 제작	EM0-261, EM0-259, EM0-258, DM9-114, EM0-112, EM0-249, EM0-248, DM9-104, EM0-131, EM0-238, EM0-121, EM0-228, EM0-642, EM0-652, EM0-104, EM0-271, EM0-209
4	PB1	6	PBS LINE BLK(중조 PBS LINE)	DM9-234, DM9-233, DM9-232, DM9-224, DM9-223 DM9-222
5	3D1	5	대조 L/T이 9일이 적용되는 3DS BLK(E/ROOM D/BOTTOM), 대조 용접 물량이 많은 BLK	DM9-103, DM9-102, DM9-101, EM0-103, EM0-101
6	TS2	5	TSS에서 제작하는 일반적인 BLK	EM0-154, EM0-194, EM0-184, EM0-116, EM0-144
7	OF5	5	일반적인 OFD BLK	EM0-638, EM0-628, DM9-115, EM0-171, EM0-161
8	NP1	4	NPS 고정 Block	DM9-117, DM9-107, EM0-115, EM0-106
9	PA2	2	PBS H2(LINE) 제작 후 OFD 완료 BLK	DM9-156, DM9-146
10	NA2	2	NPS에서 H2(LINE) 제작 후 OFD에서 대조 완료 BLK	DM9-231, DM9-221
11	GB1	1	대조 L/T이 9일이 적용되는 GBS BLK(E/ROOM D/BOTTOM)	EM0-102
12	GB2	1	STERN BOSS BLK(110BLK) 대조 L/T 18일	EM0-110
13	3N1	1	3DS에서 곡중조를 공급해서 NPS에서 대조를 완료하는 BLK	DM9-113

(5) 계획과 실적 비교 (배량 케이스, IRMS 기준)

배량 케이스 기준 군집

군집	블록	차이 값		계획		실적	
		Σ Event	Σ Task	Fit	Cross Fit	Fit	Cross Fit
3D6	19	-17	12	0.375	0.118	0.357	0.167
PB4	18	-13	3	0.78	0.141	0.424	0.28
GB6	17	-17	9	0.375	0.16	0.422	0.307
PB1	6	18	5	0.715	0.111	0.353	0.11
TS2	5	5	3	0.883	0.5	0.86	0.6
3D1	5	-1	1	0.823	0.148	0.825	0.28
OF5	5	7	2	0.37	0.15	0.36	0.21
NP1	4	4	2	0.711	0.128	0.6	0.134
PA2	2	-2	0	0.816	0.208	0.833	0.244
NA2	2	4	1	0.875	0.183	0.717	0.196
GB1	1	0	0	1	1	1	1
GB2	1	-1	0	1	0.8	1	0.667
3N1	1	0	0	0.85	0.125	0.875	0.1
정리(전체)	**86**	**89 (+38, -51)**	**38 (+38,-0)**	**0.736**	**0.290**	**0.664**	**0.330**
정리(상위 30%)	**65**	**70**	**32**	**0.626**	**0.206**	**0.483**	**0.293**

IRMS 기준 군집

군집	블록	차이 값		계획		실적	
		Σ Event	Σ Task	Fitness	Cross Fit	Fitness	Cross Fit
C6-5	23	-27	10	0.756	0.307	0.706	0.387
C1	12	-17	3	0.305	0.131	0.386	0.237
C6-7	11	4	7	0.634	0.219	0.608	0.131
C2	9	5	-2	0.504	0.125	0.355	0.186
C6-4	8	16	1	0.873	0.128	0.458	0.247
C6-1	6	18	5	0.715	0.111	0.353	0.11
C6-6	5	-3	6	0.6	0.144	0.711	0.215
C4	4	4	-1	0.654	0.131	0.493	0.084
C6-3	2	-5	5	0.726	0.156	0.529	0.21
C3	2	-8	4	0.694	0.117	0.754	0.167
C5	2	0	3	0.78	0.115	0.707	0.101
C6-2	2	0	4	0.696	0.096	0.621	0.068
정리(전체)	**86**	**107 (+47,-60)**	**51 (+48,-3)**	**0.661**	**0.148**	**0.557**	**0.179**
합계(상위 30%)	**63**	**69**	**23**	**0.614**	**0.182**	**0.503**	**0.238**

- 배량 케이스에 의한 군집: 계획에 없는 단위업무들이 실적에 많이 나타남
- IRMS에 의한 군집의: 계획에 정의된 단위업무들이 실적에서 다수 발견되지 않음

- [정리 전체] 배량 케이스에 의한 군집이 IRMS에 의한 군집보다 단위업무 차이도 적게 나고, Cross Fit 값도 높게 나옴

- [상위 30% 군집 – 블록 개수 기준] 배량 케이스에 의한 군집이 IRMS에 의한 군집보다 단위업무 차이는 많이 나나, Cross Fit 값은 큰 차이가 없음

Abnormal process detection

- Definition of Anomaly
  - The intuitive definition of an outlier would be '**an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism**'.

Distance measure

Methodology

(Hawkins, 1980)

- Related works of anomaly detection

Intrusion Detection Systems	In many host-based or networked computer systems, network traffic data may show unusual behavior because of malicious activity. The detection of such activity is referred to as intrusion detection(Agarwal & Mittal, 2012).
Credit Card Fraud	In many cases, unauthorized use may show different patterns, such as a buying spree from geographically obscure locations. Such patterns can be used to detect outliers in credit card transaction data(Ngai, Hu, Wong, Chen, & Sun, 2011).
Interesting Sensor Events	The sudden changes in the underlying patterns may represent events of interest. Event detection is one of the primary motivating applications in the field of sensor networks(Du, Fang, & Peng, 2006).
Medical Diagnosis	In many medical applications the data is collected from a variety of devices such as MRI scans, PET scans or ECG time-series. Unusual patterns in such data typically reflect disease conditions(Purarjomandlangrudi, Ghapanchi, & Esmalifalak, 2014).
Law Enforcement	Determining fraud in financial transactions, trading activity, or insurance claims typically requires the determination of unusual patterns in the data generated by the actions of the criminal entity(Lin & Brown, 2006).
Earth Science	A significant data about weather patterns, climate changes, or land cover patterns is collected through a variety of mechanisms such as satellites or remote sensing. Anomalies in such data provide significant insights about hidden human or environmental trends, which may have caused such anomalies(Potter et al., 2003).

(Aggarwal, 2013)

Distance based API Detection (DAPID)

- Activity relation matrix

Caseid	Event trace
c^1	⟨a, b, c, d, d⟩
c^2	⟨a, b, c, b, c, d⟩
c^3	⟨a, c, b, c, d, d, d⟩
c^4	⟨a, b, c, b, c, b, c, d, d⟩
c^5	⟨a, b, c, d, d⟩
c^6	⟨a, b, c, b, c, d⟩
c^7	⟨c, b, c, d, d, d⟩
c^8	⟨a, b, c, b, b, c, d, d⟩
c^9	⟨a, b, c, d, d⟩
c^{10}	⟨a, b, c, b, c, d⟩
c^{11}	⟨a, c, b, c, d, d, d⟩
c^{12}	⟨a, b, c, b, c, b, c, d, d⟩
c^{13}	⟨a, b, c, d, d⟩
c^{14}	⟨a, b, c, b, c, d⟩
c^{15}	⟨c, b, c, d, d, d⟩
c^{16}	⟨a, b, c, b, c, b, c, d, d⟩
c^{17}	⟨a, b, c, d, d⟩
c^{18}	⟨a, b, b, c, d⟩
c^{19}	⟨a, c, b, c, d, d, d⟩
c^{20}	⟨a, b, c, b, c, b, c, d⟩

Definition :

e: Event

a_i: Activity

c^l: Caseid

M^1		Target Activity			
		a	b	c	d
Base Activity	a		1		
	b			1	
	c				1
	d				1

h_{ij}^l: Number of c^l Frequency from a_i to a_j

M^l: Activity relation matrix of c_l

I^l: Set of process instances for c^l

$$I^l = \{e | \exists e \in \mathcal{E}\}, I \subseteq \mathcal{E}$$

Distance: $$d^{lm} = \|M^l - M^m\| = \left\{\left(h_{11}^l - h_{11}^m\right)^2 + \left(h_{12}^l - h_{12}^m\right)^2 + \cdots + \left(h_{ij}^l - h_{ij}^m\right)^2\right\}^{1/2}$$

DAPID

- Algorithm

Definition:

$\mathcal{C}$: Set of cases

$\mathcal{C} = \{c^1, \cdots, c^T\}$

T : Total number of cases

r : Distance threshold $(r > 0)$

π : Fraction threshold $(0 < \pi \leq 1)$

$$\frac{\left|\left\{c^i \middle| d^{ij} \leq r\right\}\right|}{T} < \pi$$

Algorithm: Detection of API using arc matrix

- Input data:

  A set of Cases $\mathcal{C} = \{c^1, \cdots, c^T\}$

  Threshold r $(r > 0)$ and π $(0 < \pi \leq 1)$

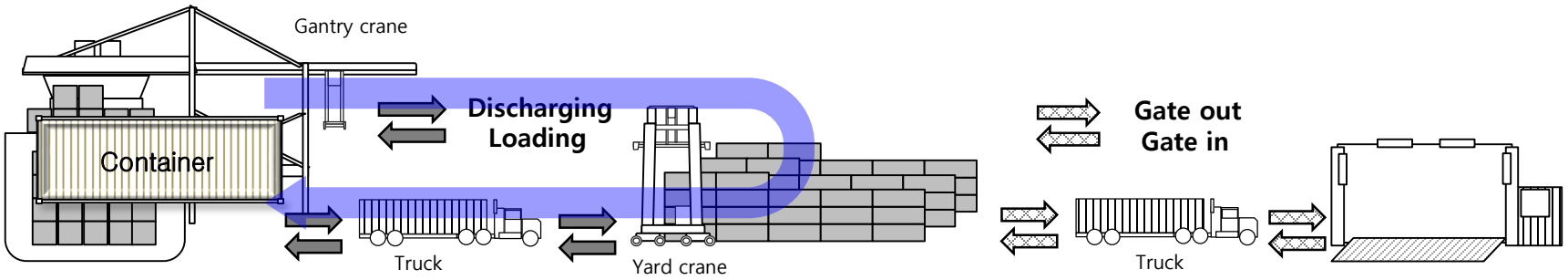
- Output: Distance-based (r, π) anomaly in $\mathcal{C}$

- Method:

1: **for** int i=0; i< *T*; i++ (Base instance)
2: **for** int j=0; j< *T*; j++ (Target instance)
3: **if** $d_{ij} < r$
4: count++;
5: **if** count $< (\pi \times T)$
6: the case is anomaly;
7: **else if** count $\geq (\pi \times T)$
8: the case is not anomaly;

Case study of DAPID

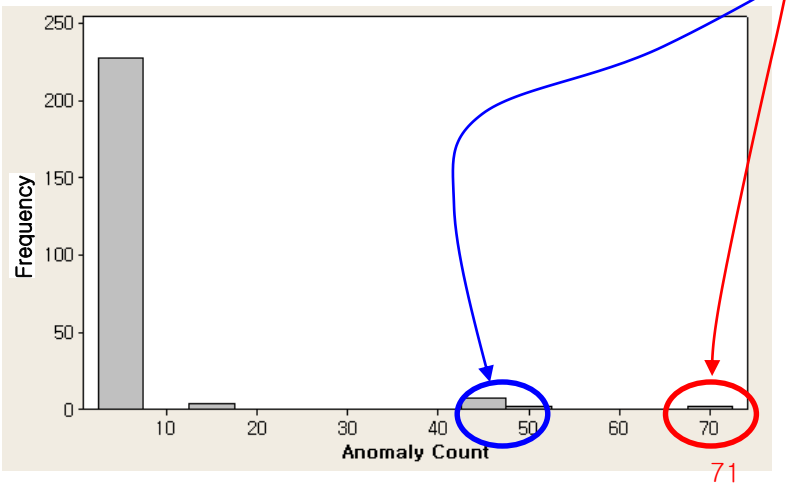
- Events of Container Terminal Operation System



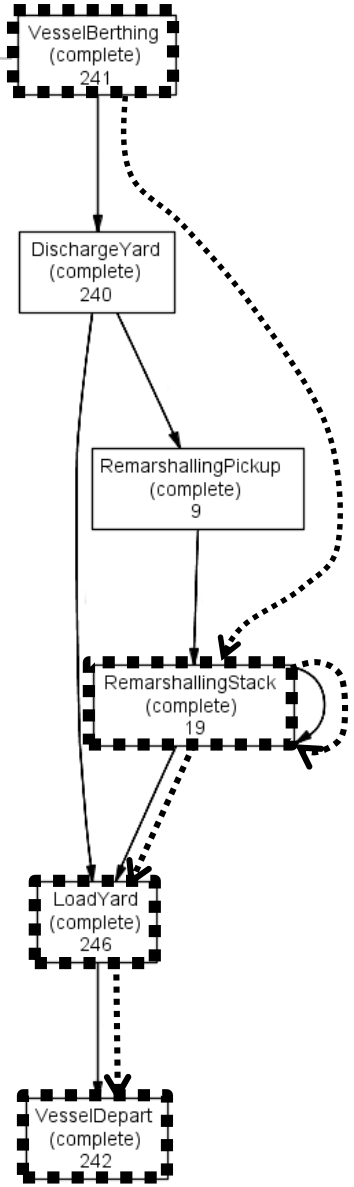
Trans shipment

Case study of DAPID

- Max Anomaly Sum of Instances

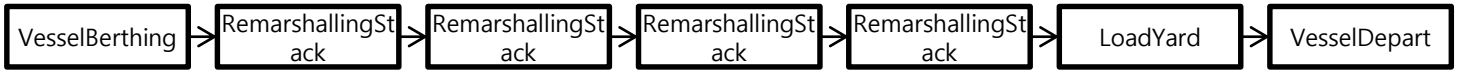


Caseid	Anomaly Sum
Container 197	71
Container 198	71
Container 060	51
Container 167	51
Container 015	44
Container 013	43
Container 014	43
Container 017	43
Container 018	43
Container 019	43
Container 020	43
⋮	
Average	9.095
Standard Deviation	9.213
Maximum	71
Minimum	7



Container 197

Container 198

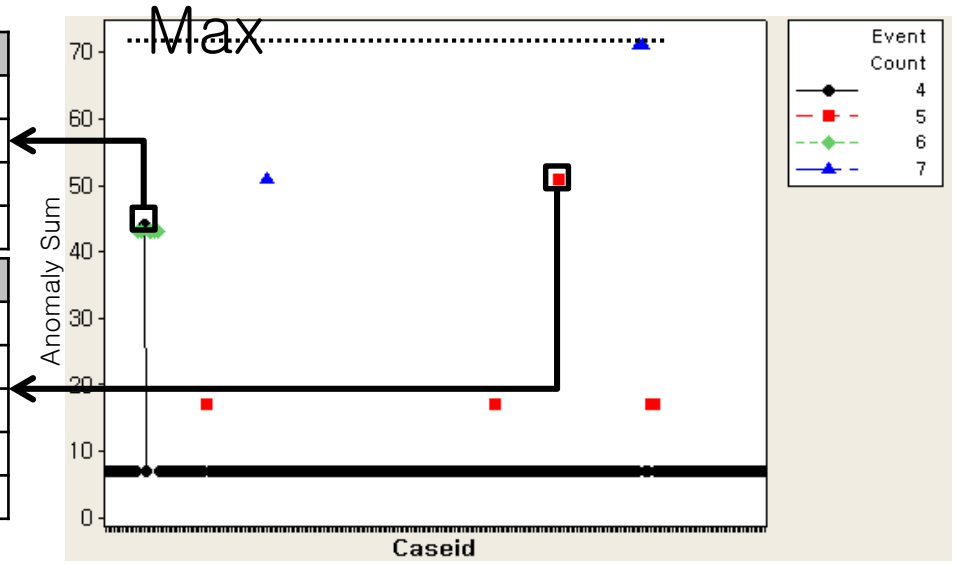


Case study of DAPID

- Needs for discovering ‘Container015’ and ‘Container167

Caseid	Activity	Timestamp
Container 015	VesselBerthing	2013-08-06 15:55:00
Container 015	DischargeYard	2013-08-06 20:58:00
Container 015	RemarshallingPickup	2013-08-08 23:24:00
Container 015	RemarshallingStack	2013-08-08 23:31:00

Caseid	Activity	Timestamp
Container 167	RemarshallingPickup	2013-08-04 3:49:00
Container 167	RemarshallingStack	2013-08-04 4:07:00
Container 167	RemarshallingStack	2013-08-04 4:07:00
Container 167	LoadYard	2013-08-05 3:48:00
Container 167	VesselDepart	2013-08-05 10:00:00



Local Anomaly Process Instance

LAPID

Algorithm: Local API Detection (LAPID)
Input data:
$\mathrm{M} = \{M^1, \cdots, M^l\}$: a set of arc matrix
k: the number of nearest neighbors
Output: Local API
Method:
l: length of M
$bC = \{\ \}$: a set of k-th nearest arc matrix of base arc matrix
$tC = \{\ \}$: a set of k-th nearest arc matrix of target arc matrix
$n, m = 0$: number of elements
d_k^i: distance of k-th nearest arc matrix
$rd = 0$: reachdistance
$blrd = 0$: local reachability density of base arc matrix
$tlrd = 0$: local reachability density of target arc matrix
$sum\_lrd = 0$: temporary variable for calculating a sum
$lof = 0$: local outlier factor
$array\_lof = [10]$: top 10 of local outlier factor

```
1: for int i=0; i< l; i++
2:   for int j=0; j< l; j++
3:     if d^ij < d_k^i (i ≠ j)
4:       C ← M^j;
5:   n = |C|;
6:   for int j=0; j< n; j++
7:     if d_k^j > d^ij (i ≠ j)
8:       rd += d_k^j;
9:     else if d_k^j < d^ij (i ≠ j)
10:      rd += d^ij;
11:  blrd = n/rd;
12:  C = { };
13:  for int j=0; j< n; j++
14:    rd = 0;
15:    for int p=0; p< l; p++
16:      if d^jp < d_k^j (j ≠ p)
17:        C ← M^j;
18:    m = |C|;
19:    for int p=0; p<m; p++
20:      if d_k^p > d^jp (j ≠ p)
21:        rd += d_k^p;
22:      else if d_k^p < d^jp (j ≠ p)
23:        rd += d^jp;
24:    tlrd = m/rd;
25:    sum_lrd += tlrd/blrd;
26:  lof = sum_lrd/n;
27:  for int j=0; j<10; j++
28:    if lof > array_lof[j]
29:      array_lof[j] ← lof;
```

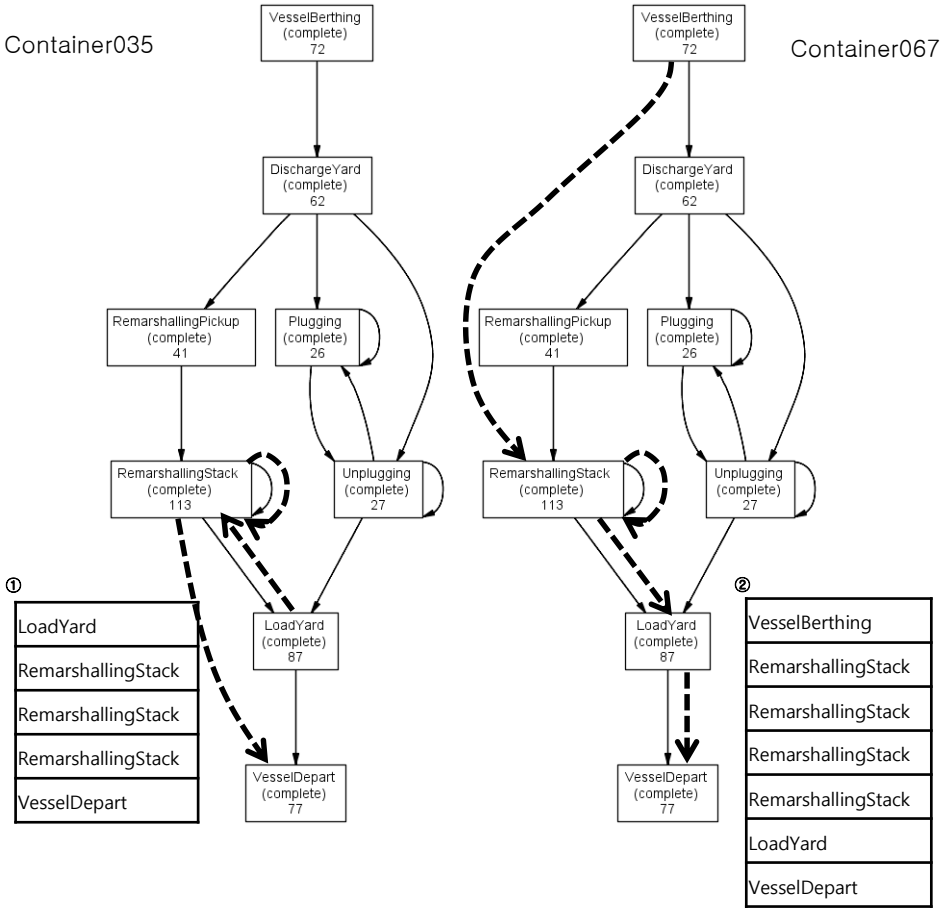
Case study (1)

- LAPID (Top 20)

①②New entry
③ Slope down
④ Slope up

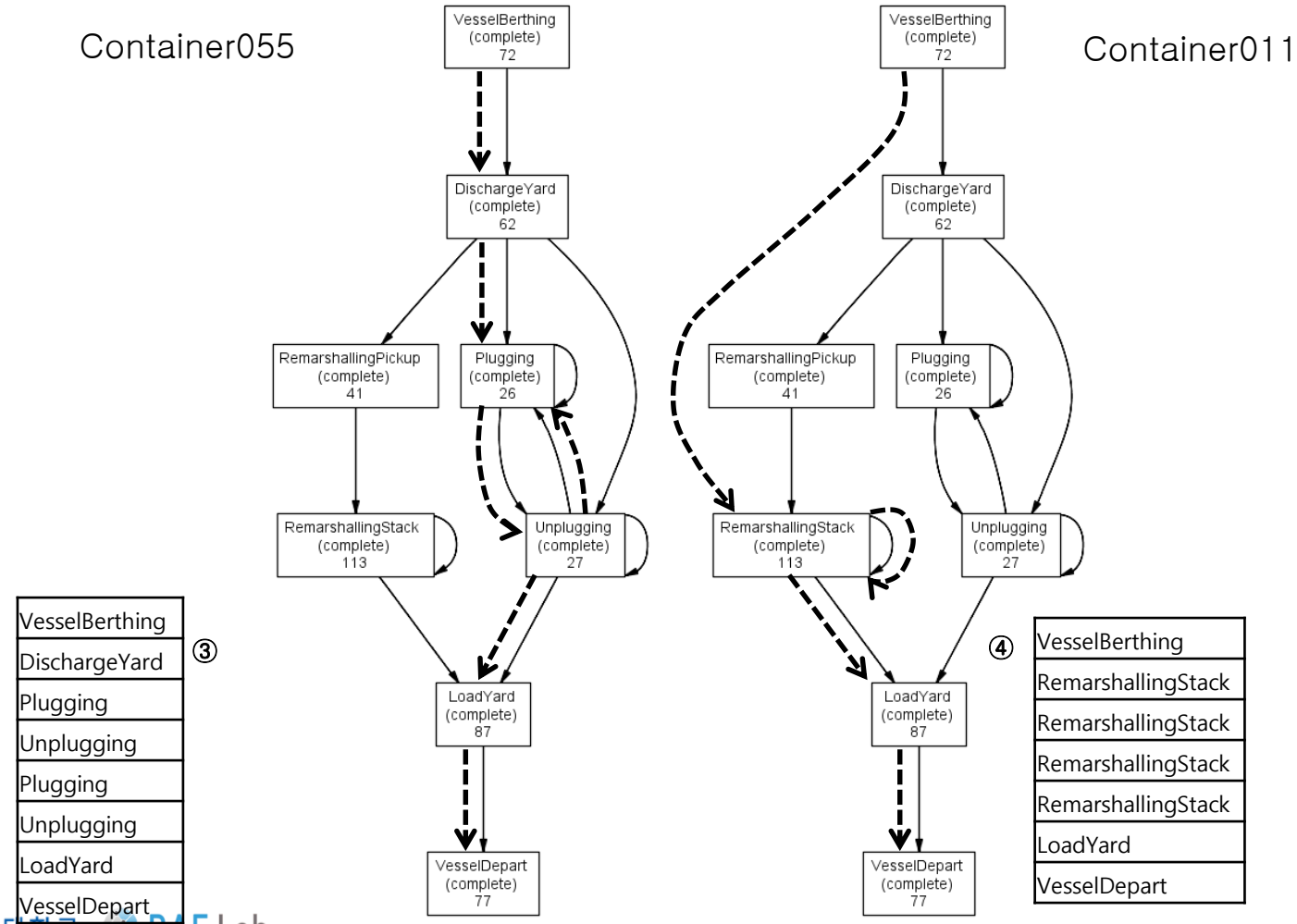
	DAPID	Results
⑤	Container058	80
⑤	Container041	80
	Container062	77
①	Container035	76
	Container037	76
	Container054	76
③	Container055	72
	Container056	72
	Container059	72
	Container063	72
	Container064	72
	Container022	71
	Container033	70
	Container077	70
	Container072	69
④	Container011	66
	Container012	66
	Container015	66
	Container023	66
	Container065	66

	LAPID	*k*=30
	Container058	4.1484
	Container041	3.6188
	Container059	3.0281
	Container062	3.0109
	Container054	2.9301
④	Container011	2.5824
	Container012	2.5824
	Container015	2.5824
	Container023	2.5824
	Container065	2.5824
②	Container067	2.5824
	Container070	2.5824
	Container071	2.5824
	Container078	2.5824
	Container040	2.4889
	Container033	2.3192
	Container077	2.3192
③	Container055	2.2922
	Container056	2.2922
	Container063	2.2922



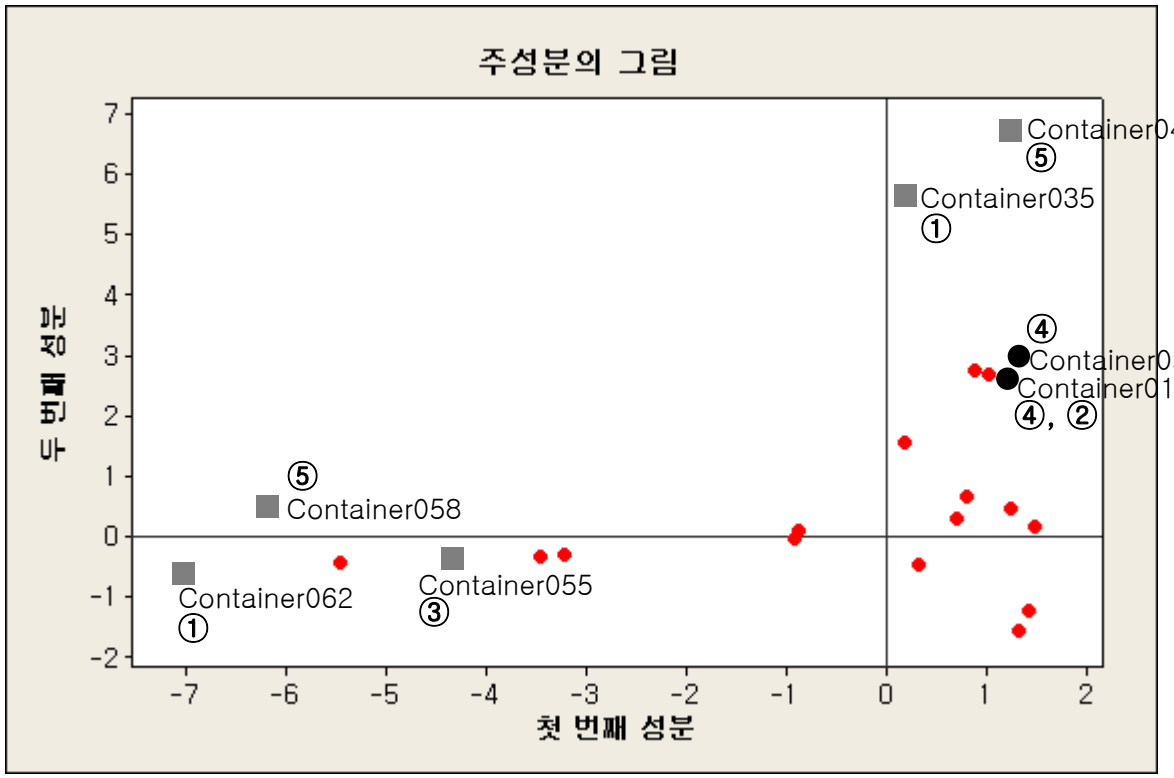
Case study (1)

- LAPID (Top 20)



Case study (1)

- Comparing DAPID & LAPID
  - Principal Component Analysis



① New entry

② New entry of LAPID

③ Slope down

④ Slope up to LAPID

⑤ Top 2

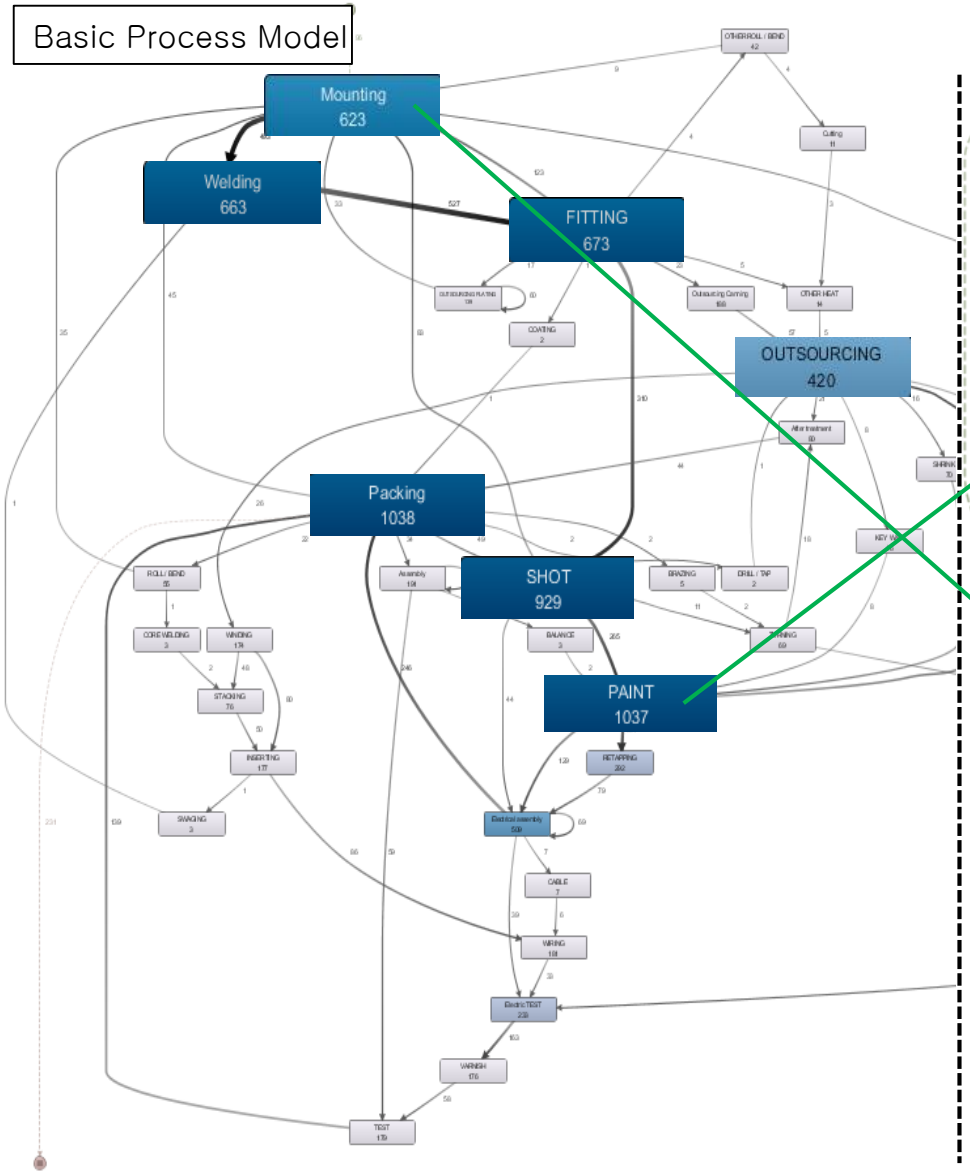
Case study (2)


- LAPID using Manufacturing data
  - 8,285 events
  - 236 caseid

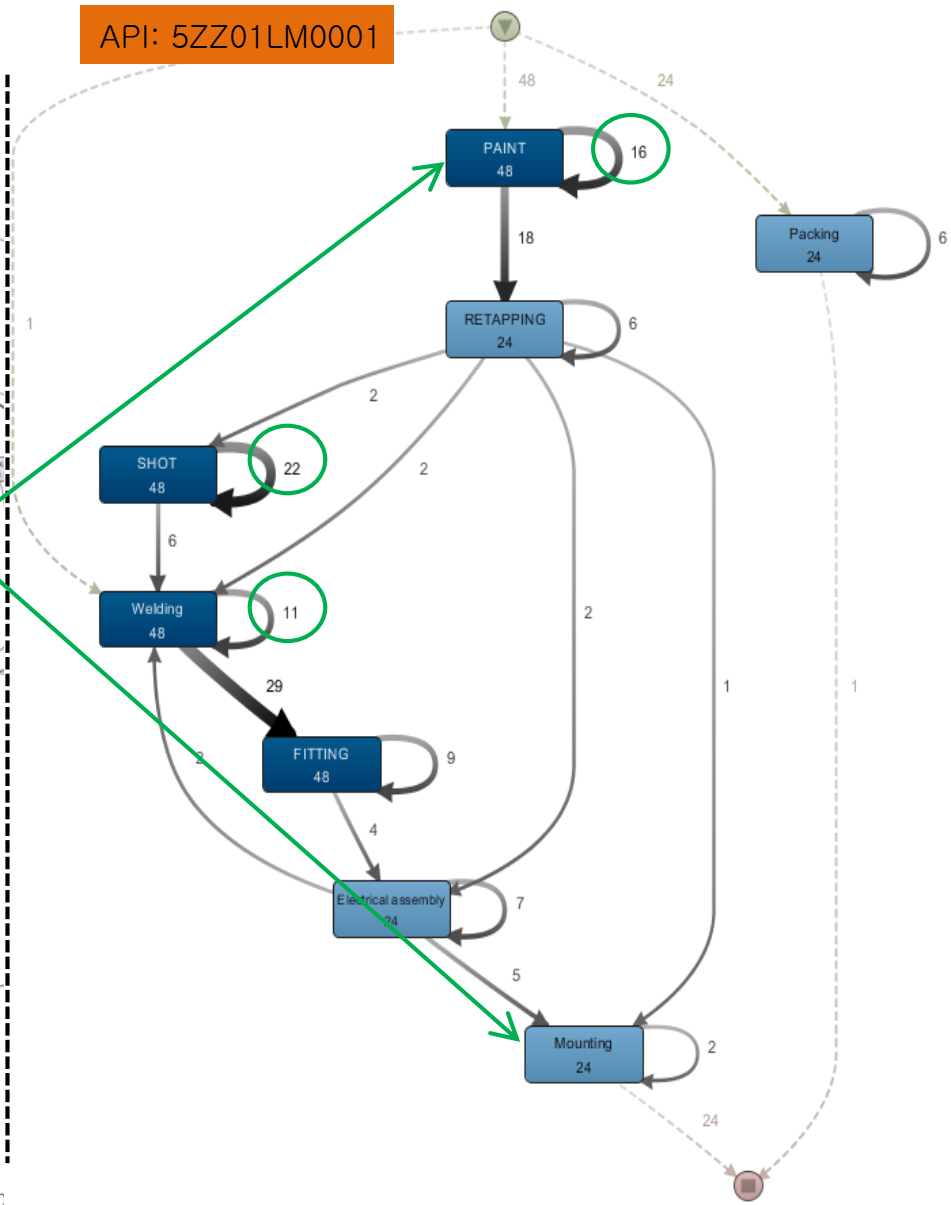
Caseid	(k=10)	rank	(k=15)	rank	(k=20)	rank
5ZZ01LM0001	5.150704838	1	5.559636419	1	5.585752894	1
43246RPM054	3.015462114	2	2.437775872	8	2.106783259	19
44257RAL633	2.938925109	3	2.464184825	7	2.207233025	11
40952RAS001	2.934231297	4	2.572686477	4	2.37872769	5
32301RAS012	2.900885409	5	2.152198964	19	1.946117464	30
40103RAL014	2.852259181	6	2.412442398	9	2.201341959	12
42383RAS010	2.843656218	7	2.219742455	17	1.924353618	33
43636RAL539	2.728799395	8	2.247893361	13	2.02089993	22
42075RAH108	2.702621059	9	2.63666672	3	2.575773608	2
43636RAL538	2.458285388	10	2.697906429	2	2.503150629	3
20365RBS001	2.419901384	11	2.313754958	11	2.320183363	8
42415RAL380	2.390590994	12	1.85309592	41	1.627223351	61
40603RAL120	2.382284617	13	1.984873436	29	1.816328357	38
40503RAL100	2.376927115	14	2.112693792	22	2.008891817	23
43731RAL547	2.362396308	15	2.238379339	14	2.327655596	7
43272RAL496	2.353544103	16	1.901627565	36	1.641563821	56
43635RBH026	2.306224119	17	1.976720856	30	1.972865702	26
42076RAH109	2.288941123	18	2.22053939	16	2.318081283	9
50120RBH001	2.287154127	19	2.064860123	25	2.080032486	21
43201RPH230	2.275566667	20	1.825459826	42	1.560376815	69
31872RAH034	2.274453768	21	1.710278124	57	1.494732338	88
32138RAL387	2.26717883	22	2.232726657	15	2.17423677	13
82570RAL964	2.250098024	23	1.612009973	73	1.40099068	122
42383RAS011	2.234786625	24	2.527964407	6	2.315005511	10
50408RAH015	2.230355728	25	2.541745345	5	2.332075963	6

Case study (2)

Basic Process Model

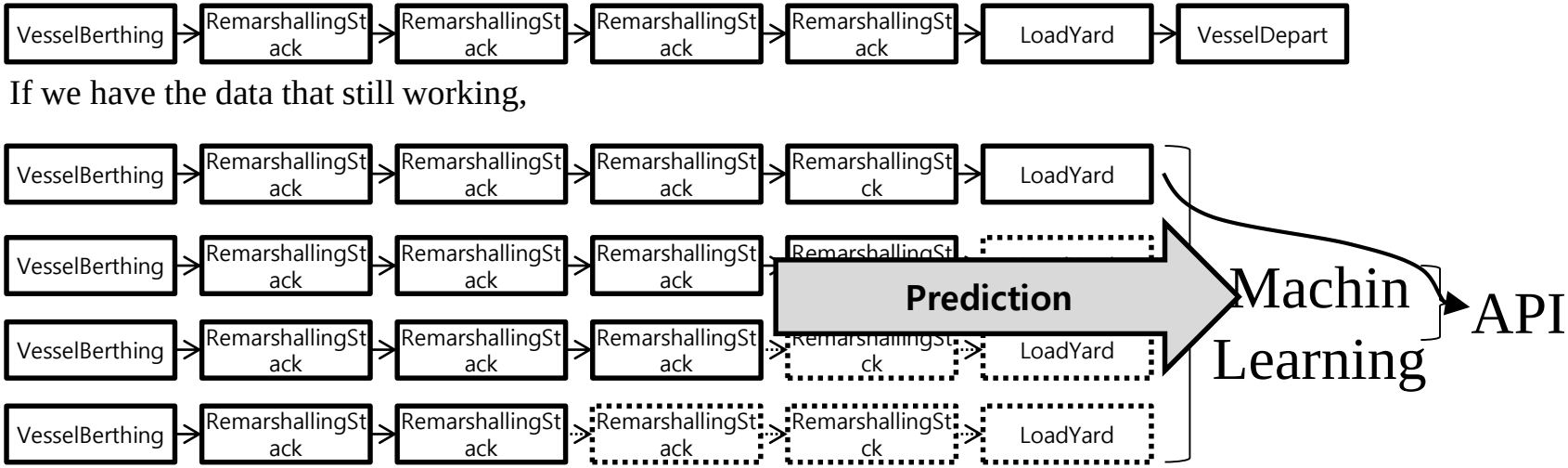


API: 5ZZ01LM0001



Incomplete Anomaly Process Instance Prediction (IAPIP)

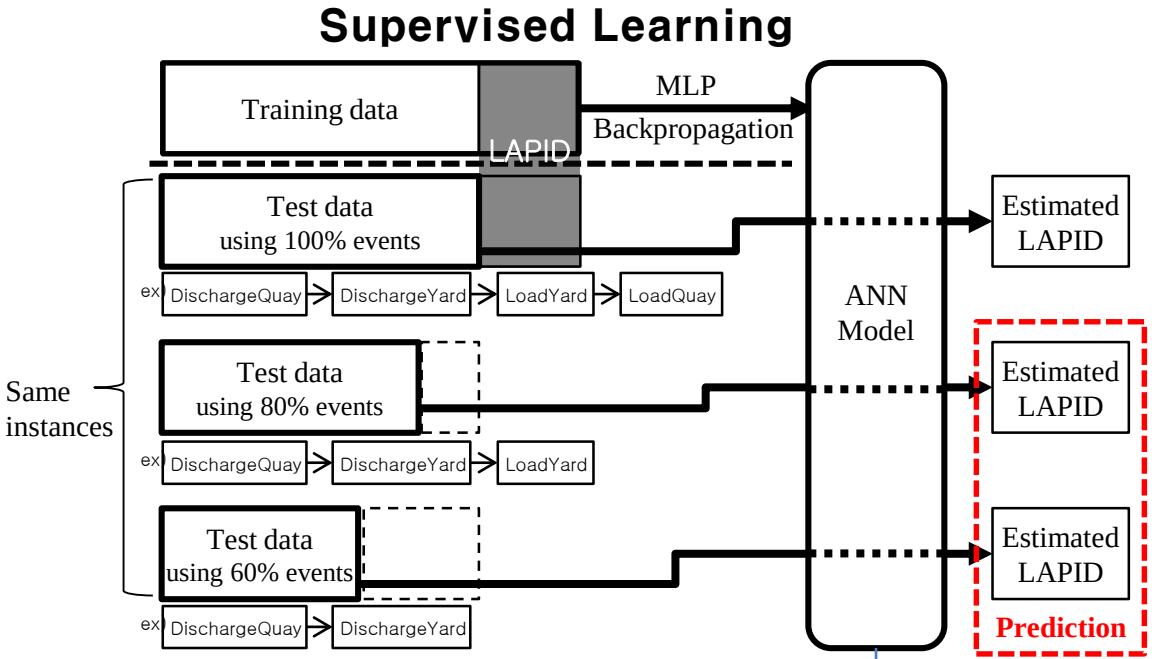
- Prediction for unfinished process instance
  - **API Container**

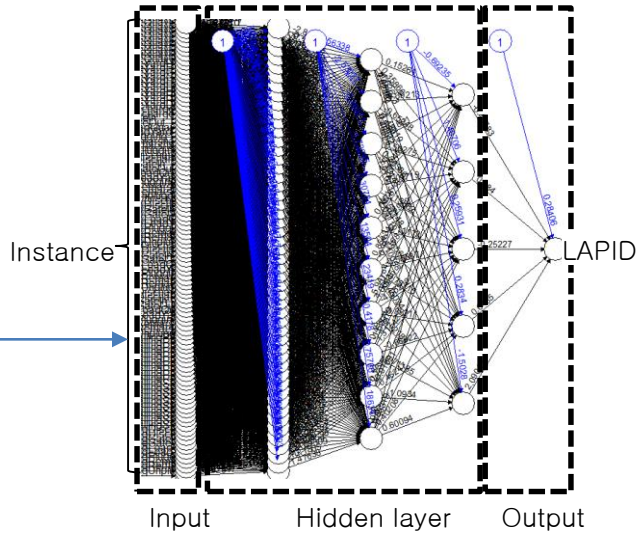


- Multilayer Perceptron (Feedforward neural network)
  - Algorithm
    - **Backpropagation**
    - Resilient backpropagation
    - Globally convergent
  - Error calculation
    - **Sum of squared errors**
    - Cross-entropy
  - Activating function
    - Logistic function
    - Tangent hyperbolics
  - Hidden layer (n_l)
    - **Usually, $n_l \leq 3$**
  - Hidden node (n_n)
    - **$n_n <$ a number of input**
    - **$n_n >$ a number of output**
  - Learning rate (r)
  - Decision of n_l, n_n & r
    - Trade off a decision against performance
    - No rule

IAPIP

- Overview of API prediction using MLP





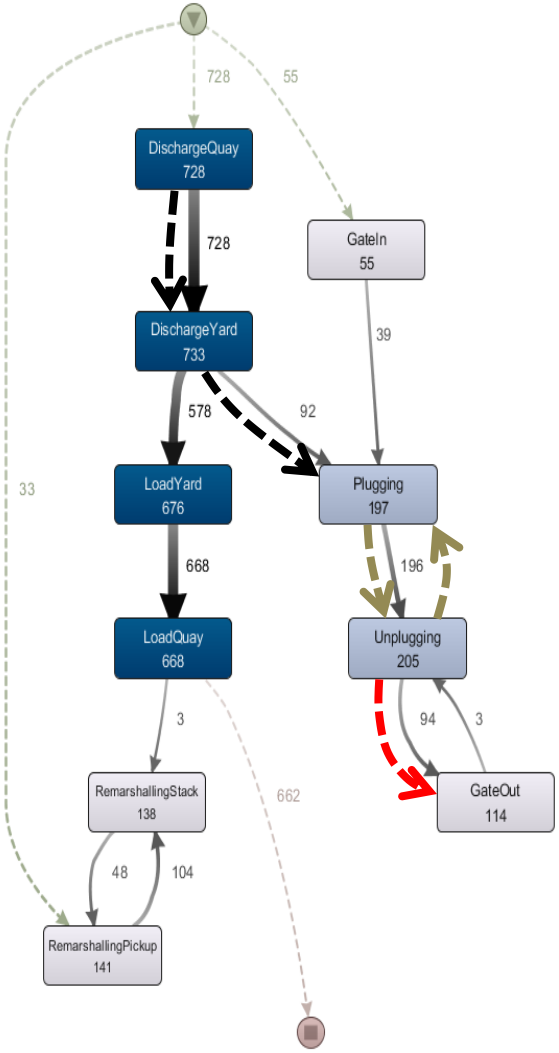
Top 20 LAPID & estimated LAPID

- Comparison between the LAPID and the estimated LAPID

LAPID	Caseid	Estimated LAPID (100%)	Estimated LAPID (80%)	Estimated LAPID (60%)
2.95051	Container1433	1.81982	1.91056	1.78222
2.49805	**Container0905**	2.50340	2.05424	1.56416
2.49805	Container1505	2.50340	2.05424	1.56416
1.91769	Container0916	1.91757	1.93757	1.41061
1.91769	Container1349	1.91757	1.93757	1.41061
1.91769	Container1527	1.91757	1.93757	1.41061
1.78479	Container1413	1.84440	1.76639	1.76388
1.71851	Container1317	1.71864	1.03382	0.95695
1.71851	Container1320	1.71864	1.03382	0.95695
1.70353	Container1344	1.90951	1.63720	1.22585
1.63422	Container1520	1.90222	1.79048	1.59162
1.61138	Container1422	2.13190	2.07654	1.89740
1.49652	Container1437	1.49661	1.50664	1.00671
1.49652	Container1442	1.49661	1.50664	1.00671
1.49639	Container1426	1.49634	1.56416	1.00671
1.45757	Container1191	1.64828	1.44372	1.03382
1.43935	Container1516	1.09934	1.10697	1.00082
1.43644	Container0862	1.43653	1.22733	1.18135
1.43644	Container0972	1.43653	1.22733	1.18135
1.43644	Container1108	1.43653	1.22733	1.18135

Container0905

DischargeQuay	DischargeQuay	DischargeQuay
DischargeYard	DischargeYard	DischargeYard
Plugging	Plugging	Plugging
Unplugging	Unplugging	Unplugging
Plugging	Plugging	Plugging
Unplugging	Unplugging	Unplugging
Plugging	Plugging	
Unplugging	Unplugging	
GateOut		
100%	80%	60%



Case study of IAPIP

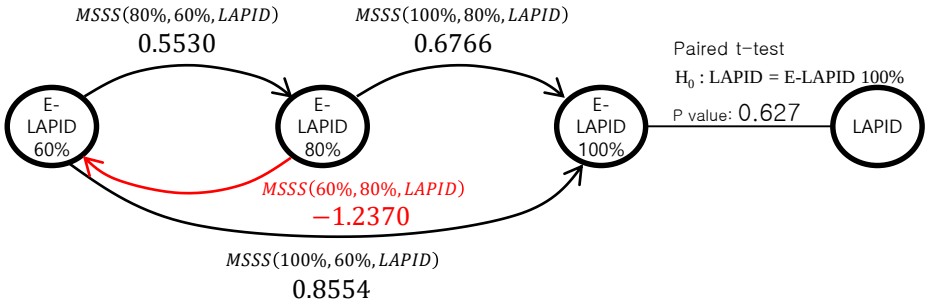
- Top 10 of actual LAPID with estimated LAPID & Descriptive statistics of test data

Caseid	Actual LAPID	Estimated LAPID (100% events)	Estimated LAPID (80% events)	Estimated LAPID (60% events)
Container1433	2.95051	1.81982	1.90618	1.77801
Container0905	2.49805	2.50340	2.05195	1.54864
Container1505	2.49805	2.50340	2.05195	1.54864
Container0916	1.91769	1.91757	1.92975	1.40244
Container1349	1.91769	1.91757	1.92975	1.40244
Container1527	1.91769	1.91757	1.92975	1.40244
Container1413	1.78479	1.84440	1.85511	1.69763
Container1317	1.71851	1.71864	1.03382	0.95621
Container1320	1.71851	1.71864	1.03382	0.95621
Container1344	1.70353	1.90951	1.64696	1.25287
≀	≀	≀	≀	≀
Descriptive statistics of test data				
Mean	1.05344	1.05249	1.01356	1.00806
Min	0.99852	0.90543	0.83277	0.84181
Max	2.95051	2.50340	2.09037	1.92907
Standard deviation	0.16525	0.15906	0.13858	0.08807

- Table 6. Correlation & Mean square error between actual LAPID and estimated LAPID

		Estimated LAPID (100% events)	Estimated LAPID (80% events)	Estimated LAPID (60% events)
Actual LAPID	Correlation coefficient	0.949	0.868	0.633
	Mean Square Error	0.00270	0.00836	0.01869

- MSSS (Mean Square Skill Score)



Overview of detection and prediction of API