

03.

Process Mining with ML

소 속 : 부산대학교 산업공학과
이 름 : 배혜림 교수
이메일 : hrbae@pusan.ac.kr

Contents

3.1 Predictive Proces Monitoring

3.2 Remaining Time Prediction

3.3 AutoPM

3.4 Data Pixelization

3.5 Process Similarity Measure

3.6 Process Analysis with Clustering

3.7 Abnormal Process Detection

3.1 Predictive Process Monitoring

PPM(Predictive Process Monitoring)

Definition

: Process Mining의 수많은 분야 중 하나로 현재 진행 중인(Pre-Mortem) 프로세스 내 객체의 미래를 예측함



어떤 분야에서, 어떻게 활용할 수 있을까?

3.1 Predictive Process Monitoring

PPM(Predictive Process Monitoring)

Application

시간 예측

프로세스 내의 어떤 객체에 대해 해당 객체의 남은 프로세스가 종료되기까지 얼마나 걸릴 것인가, 등을 예측 가능함

행동 예측

프로세스 내의 어떤 객체에 대해 해당 객체가 다음에 어떤 행동을 할 것인가, 등을 예측 가능함

결과물 예측

프로세스의 최종 결과물에 대한 예측을 수행함. 최근 동향으로 비정상적인 예측된 결과물에 대한 원인 분석을 XAI와 접목하여 수행 가능함

3.1 Predictive Process Monitoring

PPM(Predictive Process Monitoring)

Application : Example



Delay가 발생하기 전에 미리 Delay를 예측할 수 있다면?

→ 해당 문제에 대한 방지책을 미리 세울 수 있음

3.1 Predictive Process Monitoring

PPM(Predictive Process Monitoring)

Recall : What Data do we need?

Main Purpose of Process Mining

“Process Mining” is not only about creating a process model or evaluation and improving the performance of the model based on **accumulated historical data**, but it also enables us to provide direction and make predictions for current **unfinished data**.

<Data>

Post-Mortem (=Historic)

- 이미 완료된 케이스의 데이터
- 프로세스도출 및 기준 프로세스 향상에 활용 가능
- 과거데이터로 인해 해당 케이스에 대한 예측 등의 변화 제공 등 이제한

Pre-Mortem (=Current)

- 아직 완료되지 않은 케이스의 데이터
- 케이스에 대한 추천이나 다음 행동을 예측하는 등의 변화를 줄 수 있는 데이터



Model
for Prediction

3.1 Predictive Process Monitoring

PPM(Predictive Process Monitoring)

Recall : What Data do we need?

Post-Mortem (=Historic)

Patient	Activity	Timestamp	Doctor
5781	Make X-ray	23-1-2014@10.30	Dr. Jones
5833	Blood test	23-1-2014@10.27	Dr. Scott
5781	Blood test	23-1-2014@10.49	Dr. Scott
5781	CT scan	23-1-2014@11.10	Dr. Fox
5833	Surgery	23-1-2014@12.34	Dr. Scott
5781	Handle payment	23-1-2014@12.41	Carol Hope

Model
for Prediction

Model
: Train Completed

Pre-Mortem (=Current)

Patient	Activity	Timestamp	Doctor
5781	Make X-ray	23-1-2014@10.30	Dr. Jones
5833	Blood test	23-1-2014@10.27	Dr. Scott
5781	Blood test	23-1-2014@10.49	Dr. Scott
5781	CT scan	23-1-2014@11.10	Dr. Fox
5833	Surgery	23-1-2014@12.34	Dr. Scott
5781	Handle payment	23-1-2014@12.41	Carol Hope

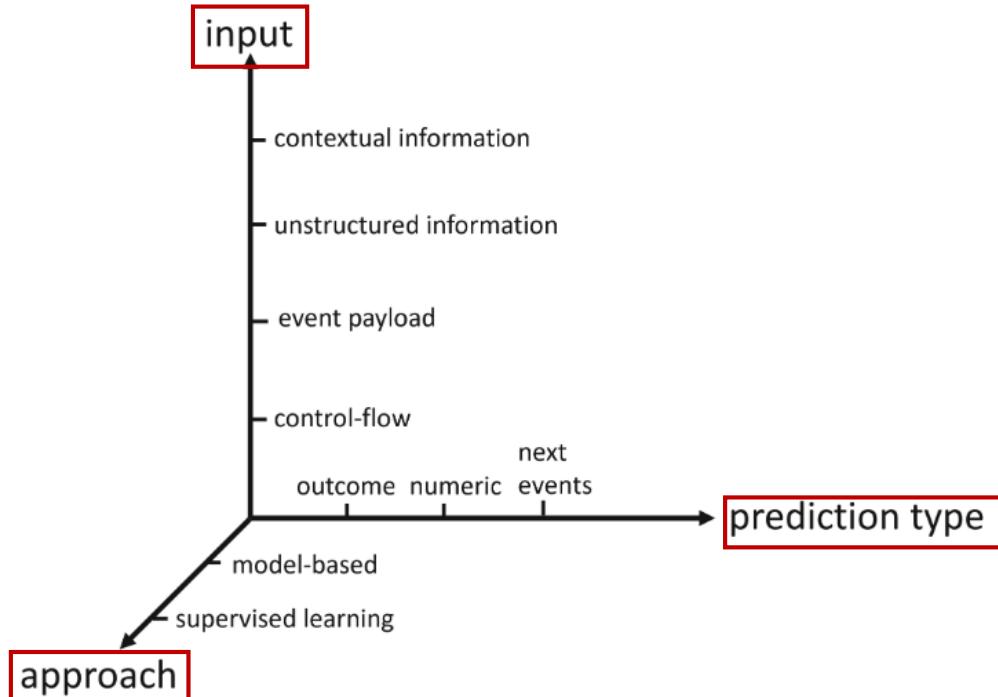
Model
: Train Completed

Prediction

3.1 Predictive Process Monitoring

Three Main Dimension

- **Prediction type** : Type of predictions provided as output
- **Approach** : Type of adopted approach and technique
- **Input** : Type of information exploited in order to get predictions



3.1 Predictive Process Monitoring

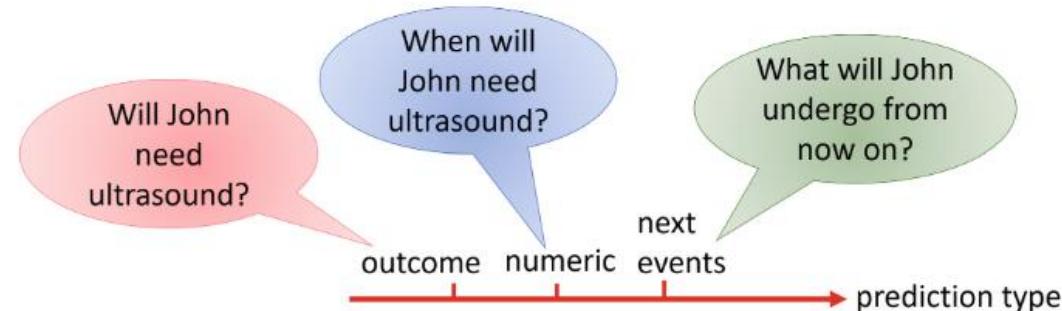
Prediction Type

- **Outcome** : Predictions related to predefined categorical or boolean outcome values
- **Numeric** : Predictions related to measures interest taking numeric or continuous value
- **Next events** : Predictions related to sequence of future activities and related data payloads

Pre-Mortem

Register patient	Go to radiology dept.	Visit patient	Perform X-ray	Perform ultrasound	Compute rate	Get payment
08:00 am	08:10 am	08:15 am	08:54 am	09:20 am	09:45 am	09:47 am

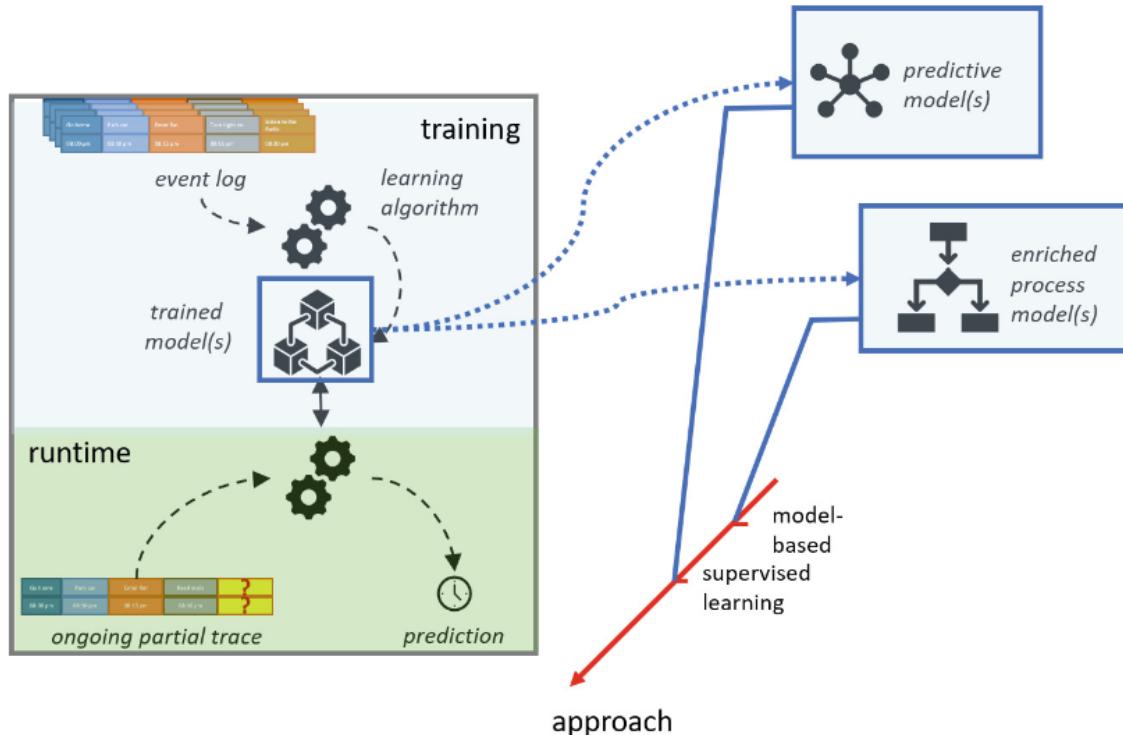
Past Now! Future



3.1 Predictive Process Monitoring

Approach

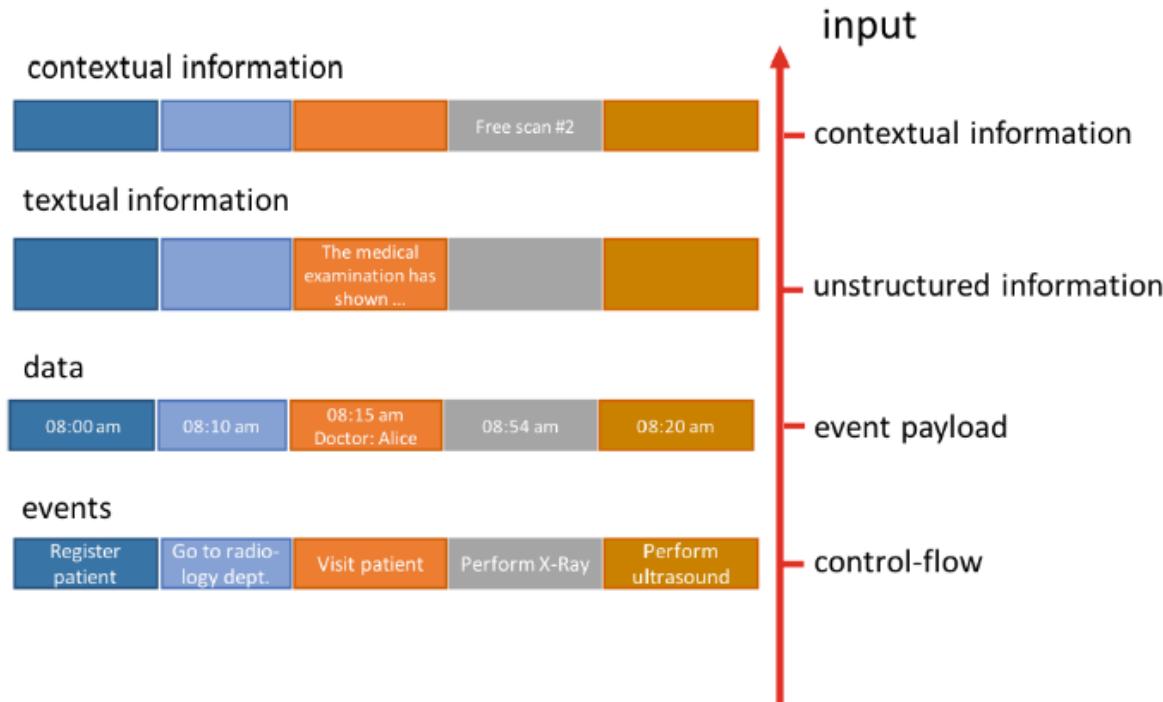
- Model-based : Approaches relying on an explicit discovered model
- **Supervised learning : Approaches leveraging machine learning and statistical techniques**



3.1 Predictive Process Monitoring

Input

- **Control-flow** : Information related to the sequence of events
- **Event payload** : Information related to the structured data payload associated to the events
- **Unstructured information** : Information related to unstructured (textual) content.
- **Contextual information** : Information related to process context



3.2 Remaining Time Prediction

Predictive Process Monitoring For Remaining Time Prediction With Transfer Learning

- Introduction

17th International Conference on Innovative Computing, Information and Control (ICICIC2023)
August 29-31, 2023, Kumamoto, Japan
Copyright©2023 ICIC International

**PREDICTIVE PROCESS MONITORING FOR REMAINING TIME PREDICTION
WITH TRANSFER LEARNING**

Alif Nur Iman¹, Imam Mustafa Kamal², Muhammad Hanif Ramadhan¹, and Hyerim Bae^{1,*}

¹Industrial Data Science and Engineering, Department of Industrial Engineering
²Institute of Intelligent Logistics Big Data
Pusan National University
2, Busandaehak-ro 63beon-gil, Geumjeong-gu, Busan 46241, South Korea
{ alifnuriman, imamkamal, mhanif }@pusan.ac.kr
Corresponding author: hrbae@pusan.ac.kr
Received XXX 2023; accepted XXX 2023

ABSTRACT. Recent advancements in predictive process monitoring have led to better performance by incorporating deep learning methodologies. However, some of these approaches employ a massive parameter architecture that requires a large dataset for training, which presents a challenge for many business process management scenarios. This paper presents a transformer-based model with transfer learning for estimating the remaining time prediction. By employing our approach, we aim to improve the efficiency of business process management by accurately predicting the remaining time of a process. Additionally, we assess the feasibility and transferability of the transfer learning method to enhance the performance of predictive process monitoring. We conduct experiments on four publicly available event logs. Besides obtaining better accuracy, we demonstrate that transfer learning could deliver positive or negative performance depending on the source and targeted process model characteristics.

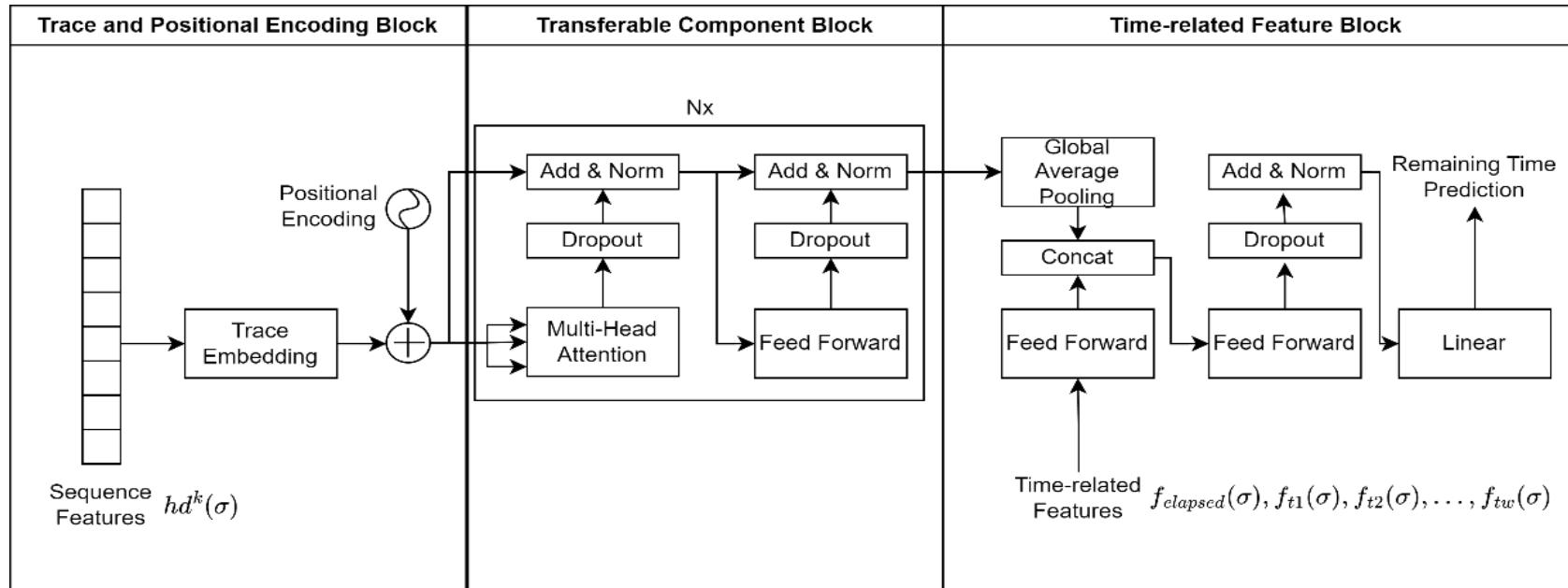
Keywords: Predictive Process Monitoring, Remaining Time Prediction, Deep Learning, Transformers, Transfer Learning.

A.N. Iman , I.M. Kamal, M.H.Ramadhan and H. Bae, "Predictive Process Monitoring for Remaining Time Prediction with Transfer Learning", 2023

3.2 Remaining Time Prediction

Predictive Process Monitoring For Remaining Time Prediction With Transfer Learning

- Framework



3.2 Remaining Time Prediction

Predictive Process Monitoring For Remaining Time Prediction With Transfer Learning

- Experiments
- Experimental Setup

Datasets	Cases	Events	Activities	Max. Case Length	Avg. Case Length	Transfer Learning
Helpdesk	4,580	21,348	14	15	5	Target Data
BPIC2020t	2,099	18,246	29	21	9	Target Data
BPIC2012	13,087	262,000	24	175	20	Source Data
Hospital	100,000	451,359	18	217	5	Source Data

MAE as metric evaluation 80% training 20% testing and 20% of training set as validation

100 Epochs, ADAM Optimizer, 10^{-3} learning rate, 1024 batch size 36 trace embedding sizes

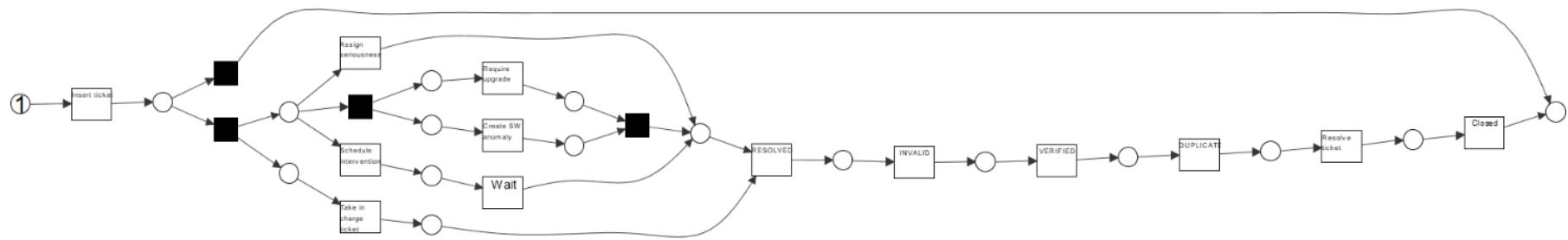
4 number of attention heads

3.2 Remaining Time Prediction

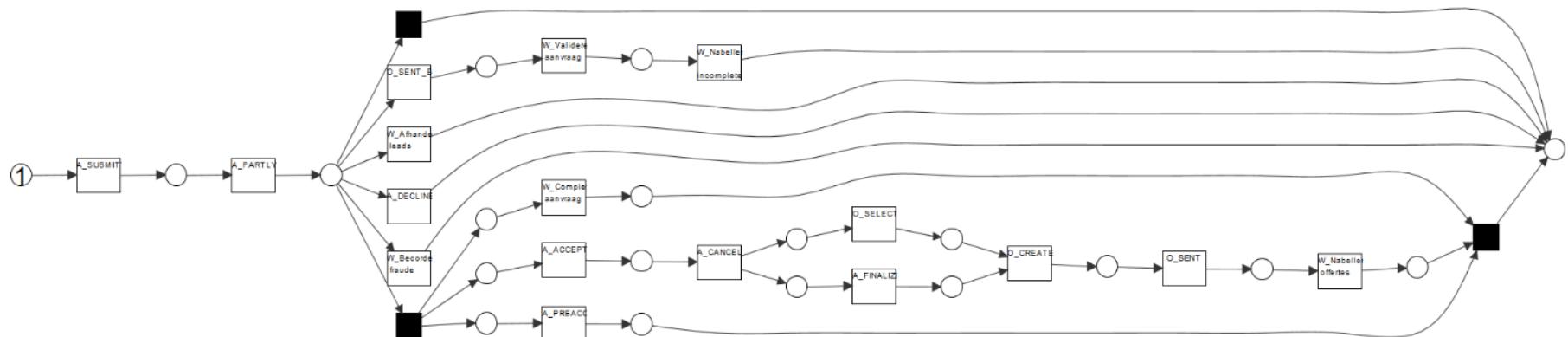
Predictive Process Monitoring For Remaining Time Prediction With Transfer Learning

- Experiments
- Process Model Analysis

Helpdesk



BPIC2012

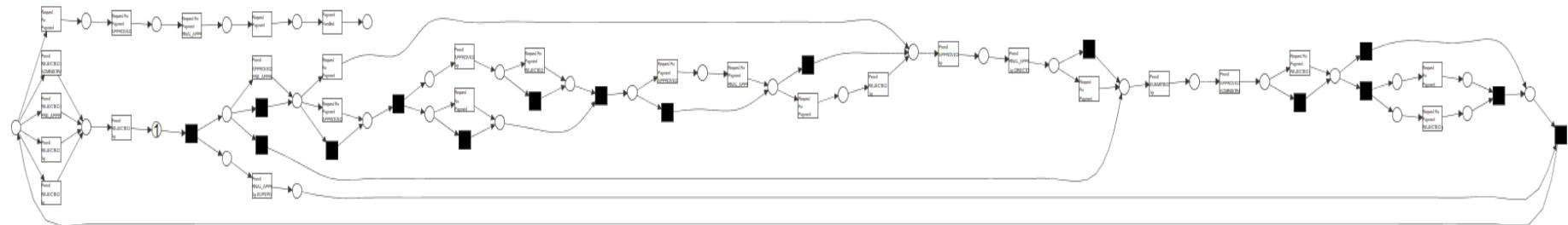


3.2 Remaining Time Prediction

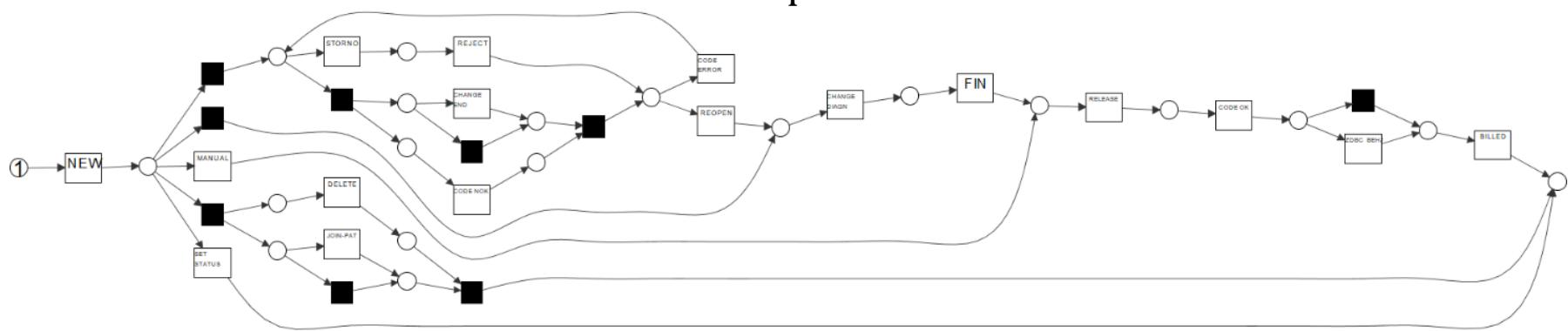
Predictive Process Monitoring For Remaining Time Prediction With Transfer Learning

- Experiments
- Process Model Analysis

BPIC2020t



Hospital



3.2 Remaining Time Prediction

Predictive Process Monitoring For Remaining Time Prediction With Transfer Learning

- Experiments
- Experimental Results

Transfer learning from	Helpdesk				BPIC2020t			
	MAE in days				MAE in days			
	Prefix 3	Prefix 7	Prefix 11	All	Prefix 5	Prefix 10	Prefix 15	All
w/o TL	2.863	2.882	1.390	3.823	64.680	39.939	35.920	57.045
BPIC2012	2.660	1.981	0.847	3.450	63.902	39.769	37.086	58.581
Hospital	2.748	2.923	1.510	3.832	63.088	39.270	34.379	54.785

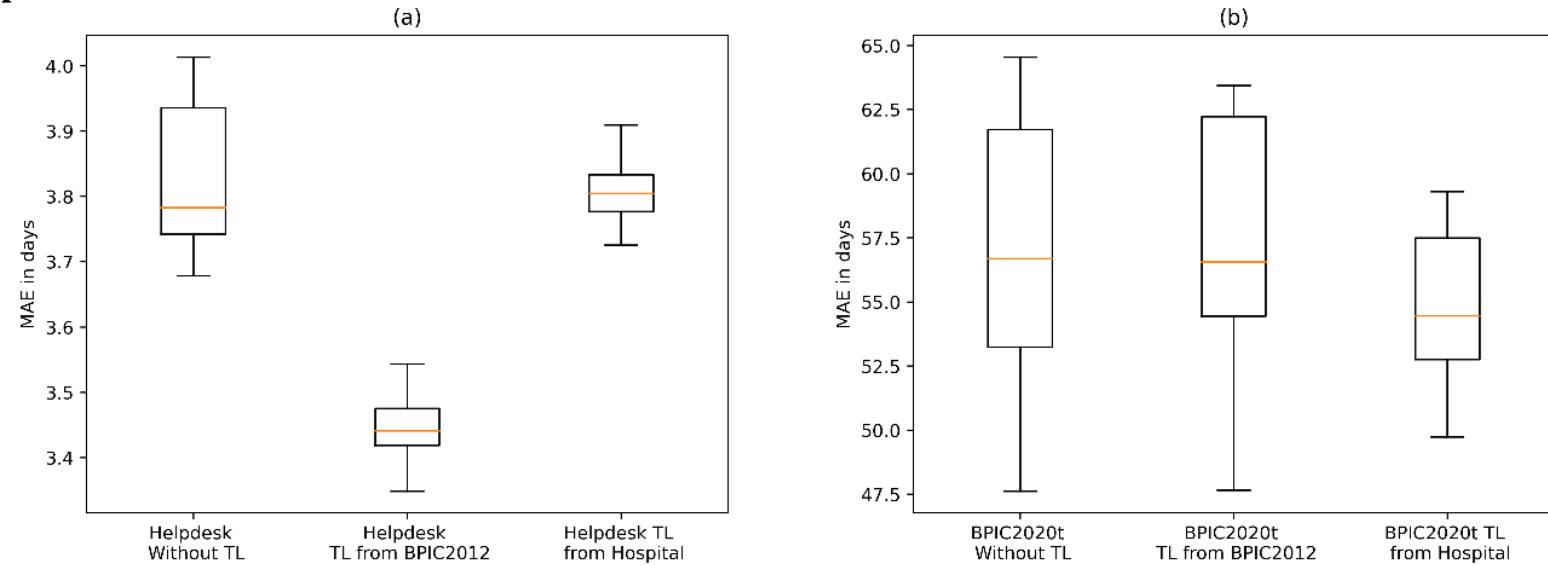
Improved performance using transfer learning

Negative performance possibly occur in
non-identical process model

3.2 Remaining Time Prediction

Predictive Process Monitoring For Remaining Time Prediction With Transfer Learning

- Experiments
- Experimental Results



Aside from improving performance
it also aids in reducing variance of results

3.2 Remaining Time Prediction

Predictive Process Monitoring For Remaining Time Prediction With Transfer Learning

- Experiments
- Experimental Results

Dataset	<i>MAE (In days)</i>					
	Tax et. al. [12]	Camargo et. al. [13]	Navarin et. al. [14]	w/o TL	TL from Hospital	TL from BPIC2012
Helpdesk	68.86	18.63	8.186	3.82	3.83	3.45

Proposed approach outperforms all baselines reported¹

TL from hospital still benefit in

- 1) Reduced variance results
- 2) Improved MAE in short prefix

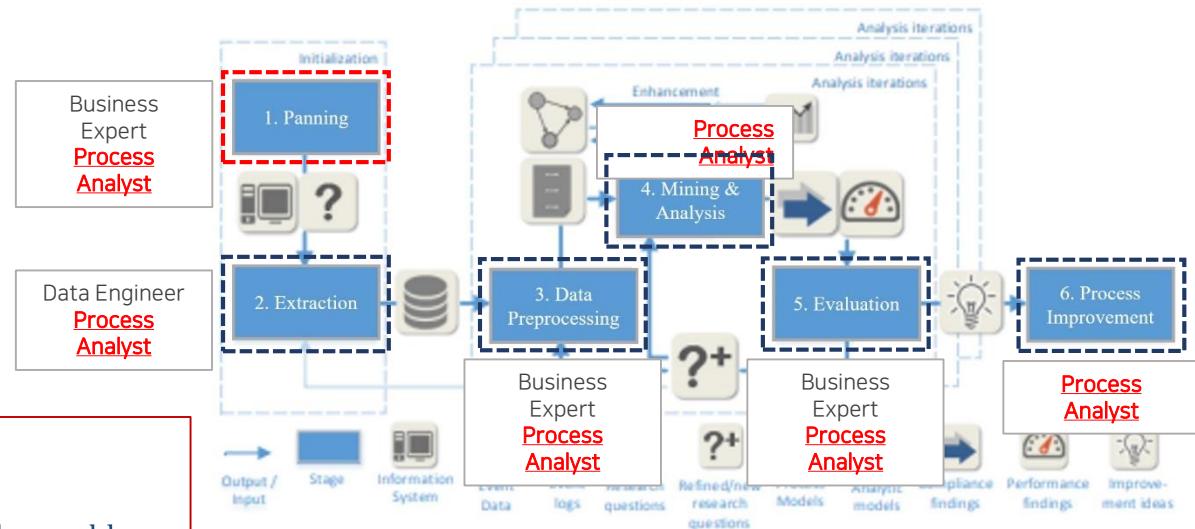
¹ E. Rama-Maneiro, J. Vidal, and M. Lama, "Deep Learning for Predictive Business Process Monitoring: Review and Benchmark," IEEE Trans. Serv. Comput., vol. 1374, no. c, pp. 1–18, 2021.

3.3 AutoPM

Hyperautomation

- The key components of hyperautomation

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



Problem

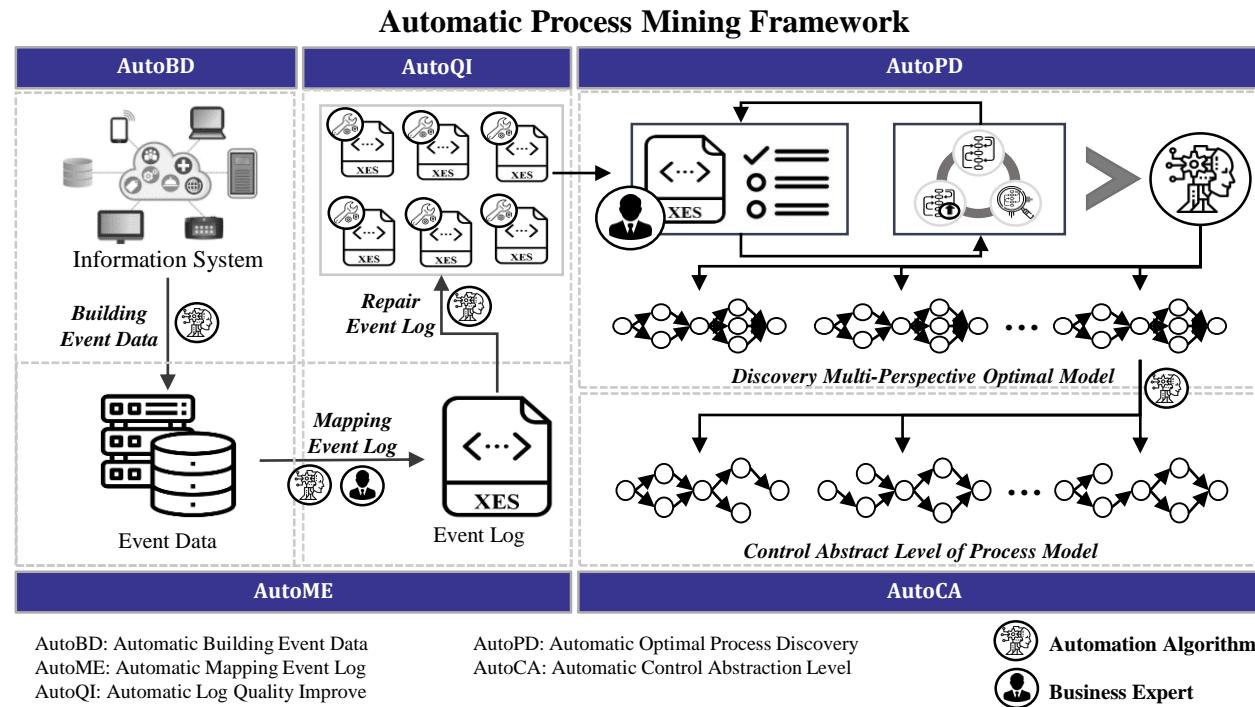
It is not easy for Domain Experts to solve problems in the field by performing the process mining process without the help of Process Analyst

3.3 AutoPM

AutoPM : Automatic Process Mining Framework

- The Goal

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

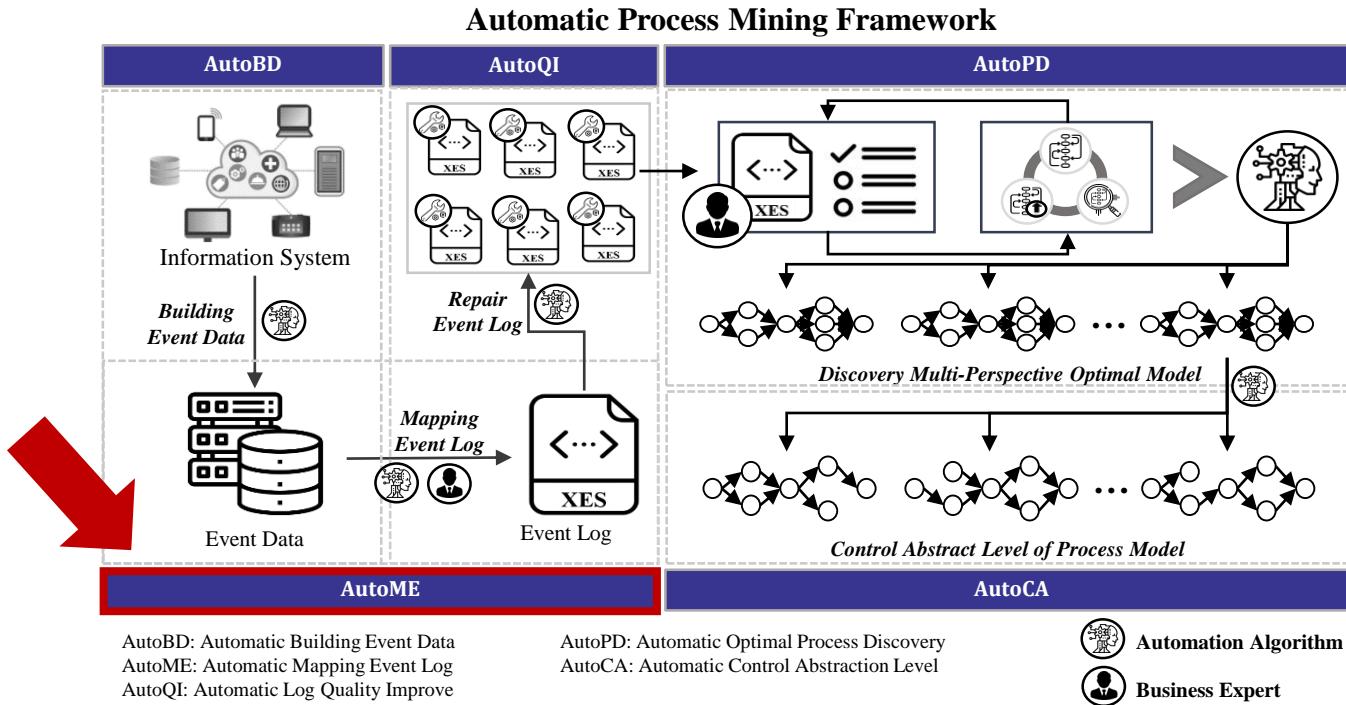


3.3 AutoPM

AutoPM : Automatic Process Mining Framework

- The Goal

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



3.3 AutoPM

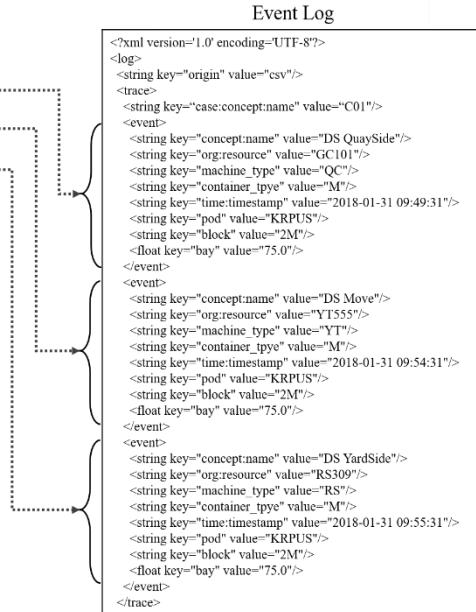
AutoME : Automatic Mapping Event Log

- Main Idea

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



3.3 AutoPM

AutoME : Automatic Mapping Event Log

- Contribution

1. A new algorithm that can automatically map from event data to the event log is proposed.
2. Proposed a new embedding method to learn the structure information of event data to perform mapping task
3. 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.

3.3 AutoPM

AutoME : Automatic Mapping Event Log

- Research Result

When 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 09:49:51	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	IB	12
C02	DS Move	YT517	YT	F	2018-01-31 05:34:51	CN XMN	IB	12
C02	DS YardSide	YC213	YC	F	2018-01-31 05:40:00	CN XMN	IB	12
C02	Re Handling	YC216	YC	F	2018-01-31 11:23:33	CN XMN	SC	27
C02	GateOut YardSide	YC216	YC	F	2018-01-31 09:18:03	CN XMN	SC	27
C02	GateOut RT	RT	F	F	2018-01-24 00:45:00	CN XMN	SC	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 09:18:03	CN XMN	SC	27
C03	GateOut RT	RT	F	F	2018-01-24 00:45:00	CN XMN	SC	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 09:55:20	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

The reference column:

Job Name

The reference column:

Pod

The reference column:

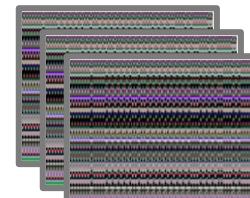
Container Type

Container No	Job Name	Machine No	Machine Type	Container Type	Pod	Block	Bay
C01	DS QuaySide	3	3	1	1	1	1
C02	DS Move	4	6	3	1	1	1
C03	DS YardSide	3	3	3	1	1	1
C04	DS QuaySide	3	3	3	1	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

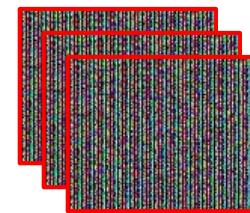
Job No	Container	Machine No	Machine Type	Container Type	Pod	Block	Bay
DS QuaySide	5	2	1	2	2	5	5
DS Move	5	4	1	2	2	5	5
DS YardSide	5	4	2	2	2	5	5
Re Handling	1	1	1	1	1	1	1
GateOut YardSide	1	2	1	1	1	2	2
GateOut	1	1	1	1	2	2	2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Pod No	Container	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
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

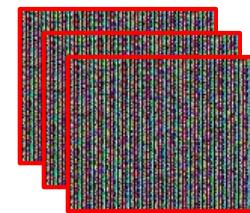
Container Type	Container No	Job Name	Machine No	Machine Type	Pod	Block	Bay
M	3	3	6	3	1	3	3
F	2	6	6	4	1	4	4
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮



Case Identifier



Activity Label



Originator Label

Attribute Label

RGB image(3-dimensional array)

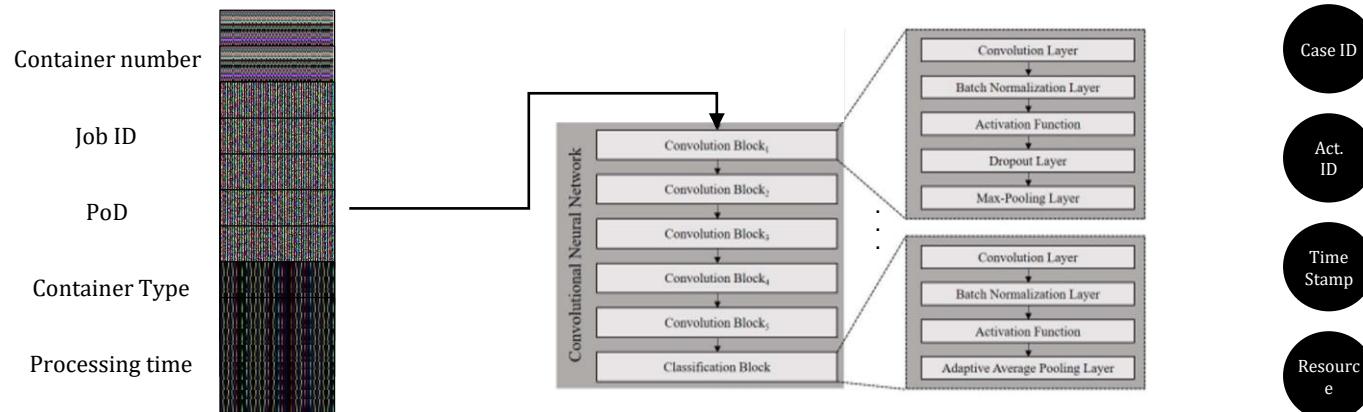
3.3 AutoPM

AutoME : Automatic Mapping Event Log

- **Research Result – Experiment 1-1**

CNN training performance comparison experiment and mapping accuracy verification according to embedding method

1. 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***).
2. Experiments were conducted using event log data (LD1 – LD 9)
3. Data for learning and testing were divided into 70% and 30% of the total data, respectively, and performed 30 times.



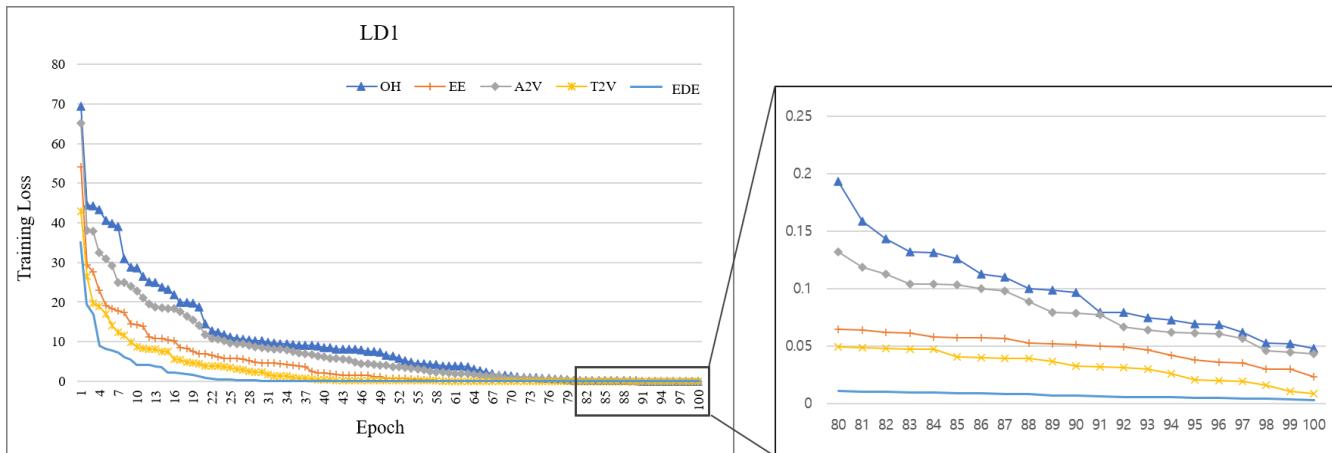
3.3 AutoPM

AutoME : Automatic Mapping Event Log

- **Research Result – Experiment 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.

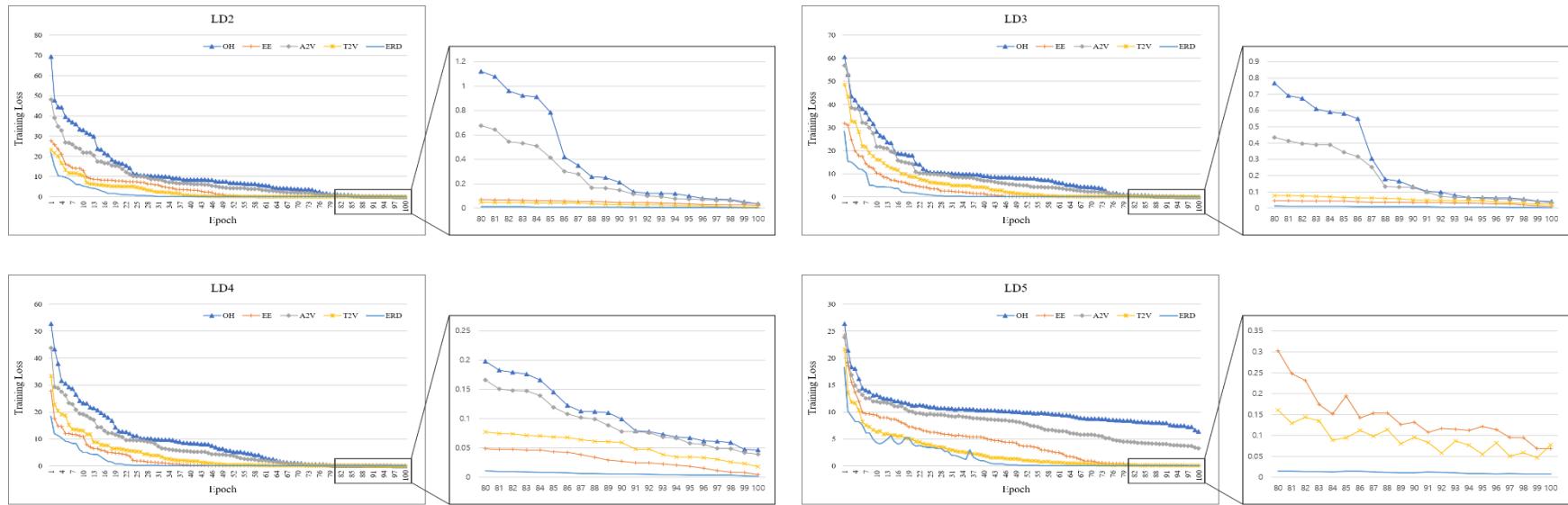


3.3 AutoPM

AutoME : Automatic Mapping Event Log

- **Research Result – Experiment 1-1**

CNN training performance comparison experiment and mapping accuracy verification according to embedding method

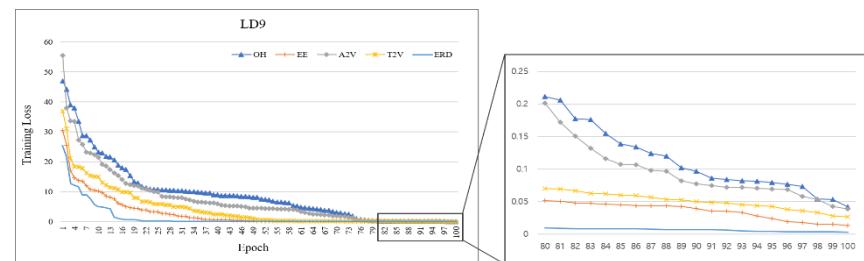
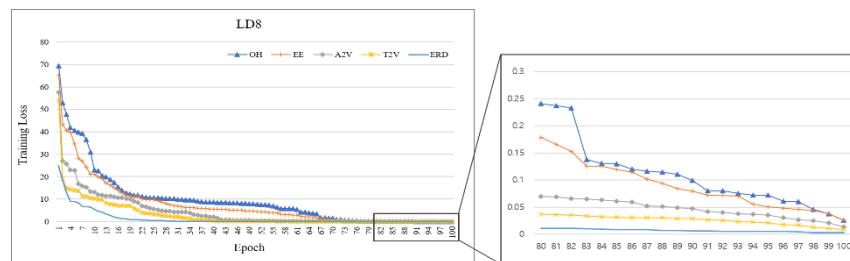
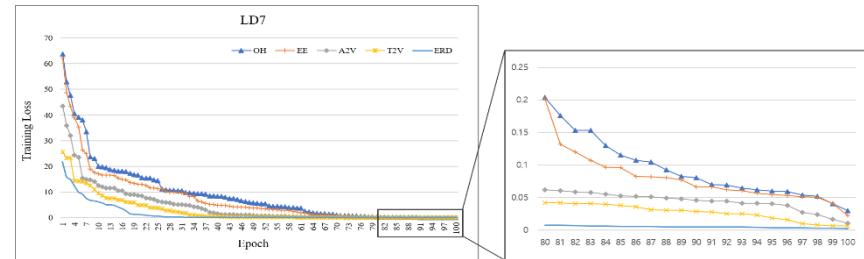
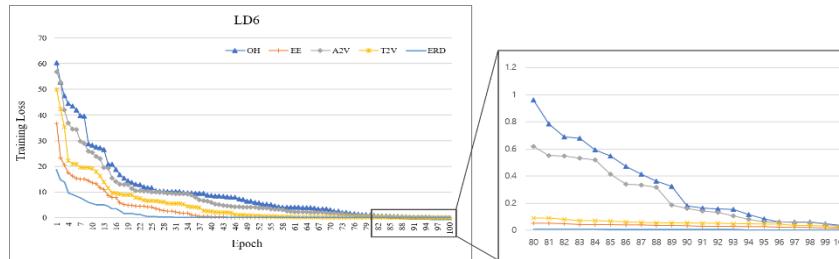


3.3 AutoPM

AutoME : Automatic Mapping Event Log

- **Research Result – Experiment 1-1**

CNN training performance comparison experiment and mapping accuracy verification according to embedding method



3.3 AutoPM

AutoME : Automatic Mapping Event Log

- **Research Result – Experiment 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)						

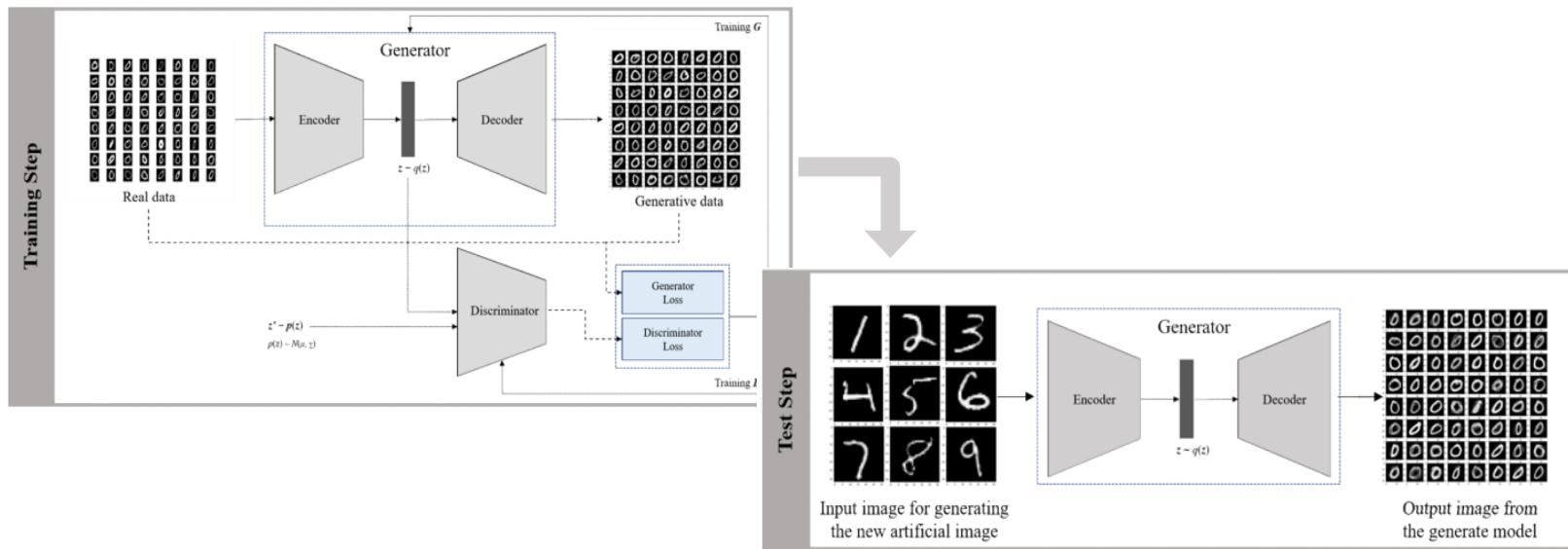
3.3 AutoPM

AutoME : Automatic Mapping Event Log

- Research Result

When 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

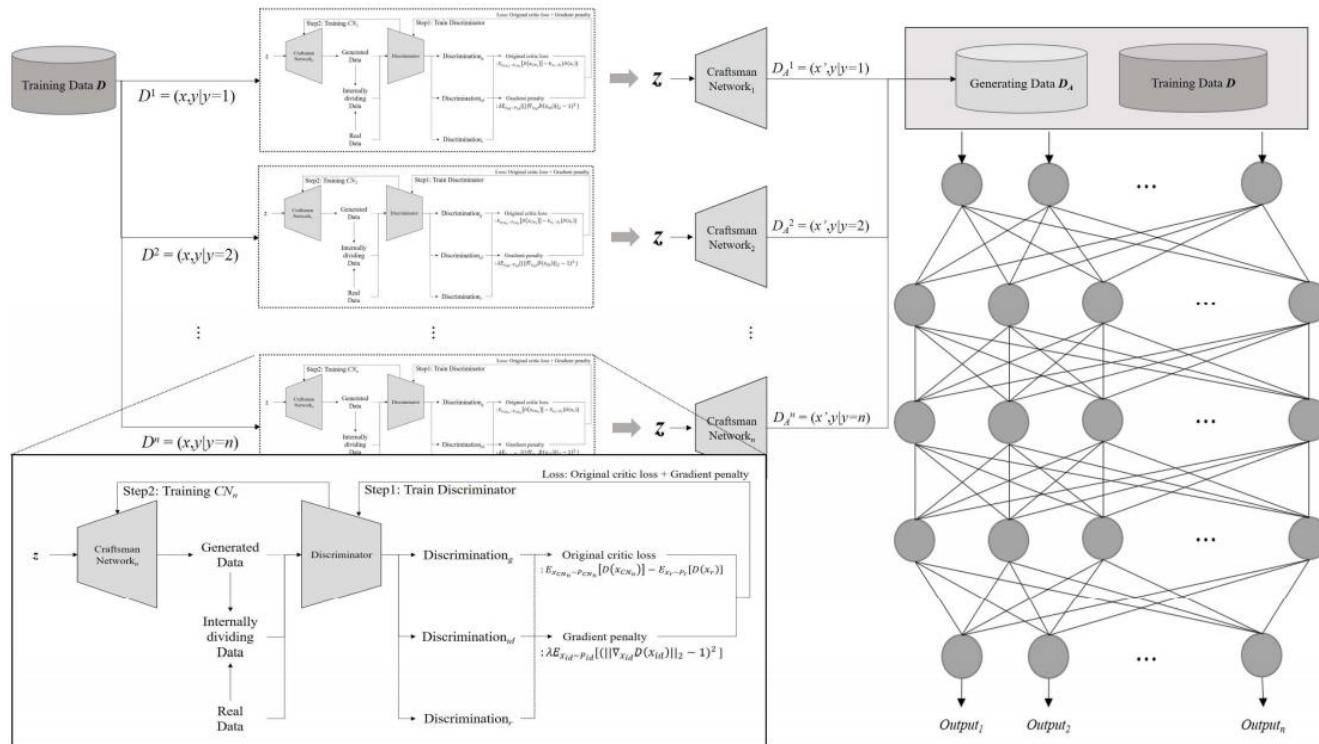


3.3 AutoPM

AutoME : Automatic Mapping Event Log

- Research Result

New supervised learning framework using Jangin training method



3.3 AutoPM

AutoME : Automatic Mapping Event Log

- **Research Result – Experiment 1-2**

Accuracy comparison experiment of automatic mapping algorithm for event data using "*Jangin Training*" method when there is not enough learning data

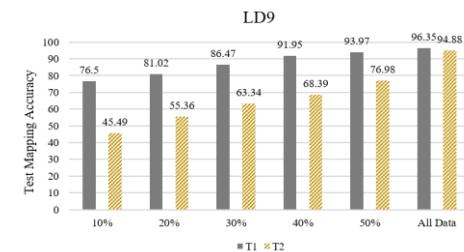
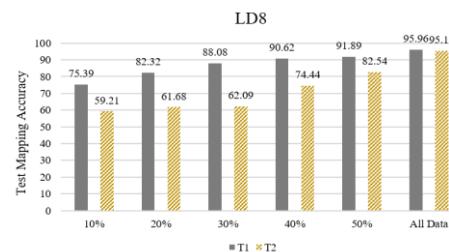
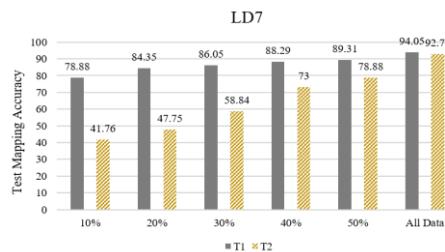
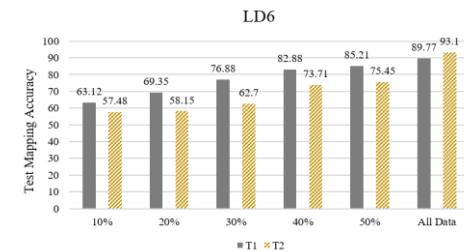
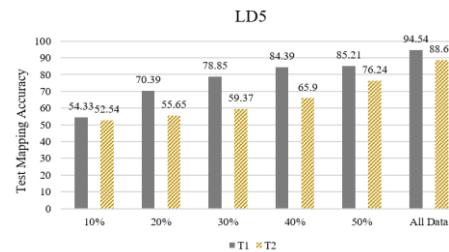
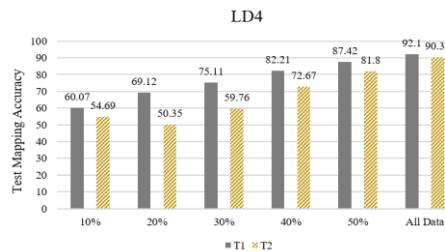
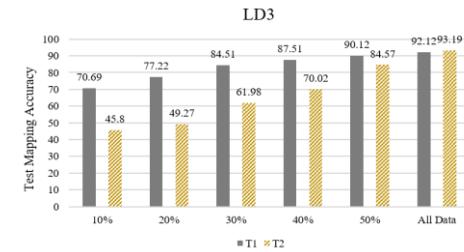
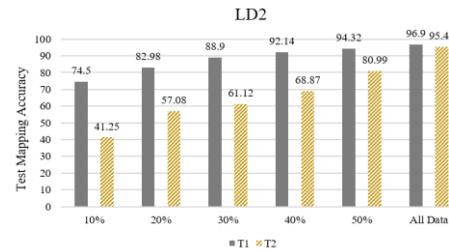
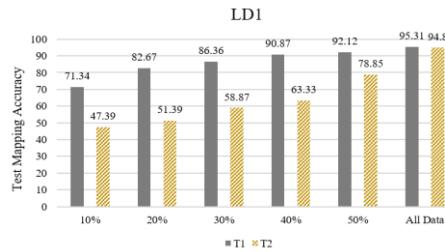
1. T1 is the result of supervised learning using the proposed learning method
2. T2 is the result of supervised learning with the given data.
3. Excellent mapping accuracy was shown when the proposed artisan learning method was used. **JT contributes a lot to the classification accuracy.**

3.3 AutoPM

AutoME : Automatic Mapping Event Log

- **Research Result – Experiment 1-2**

Accuracy comparison experiment of automatic mapping algorithm for event data using "*Jangin Training*" method when there is not enough learning data

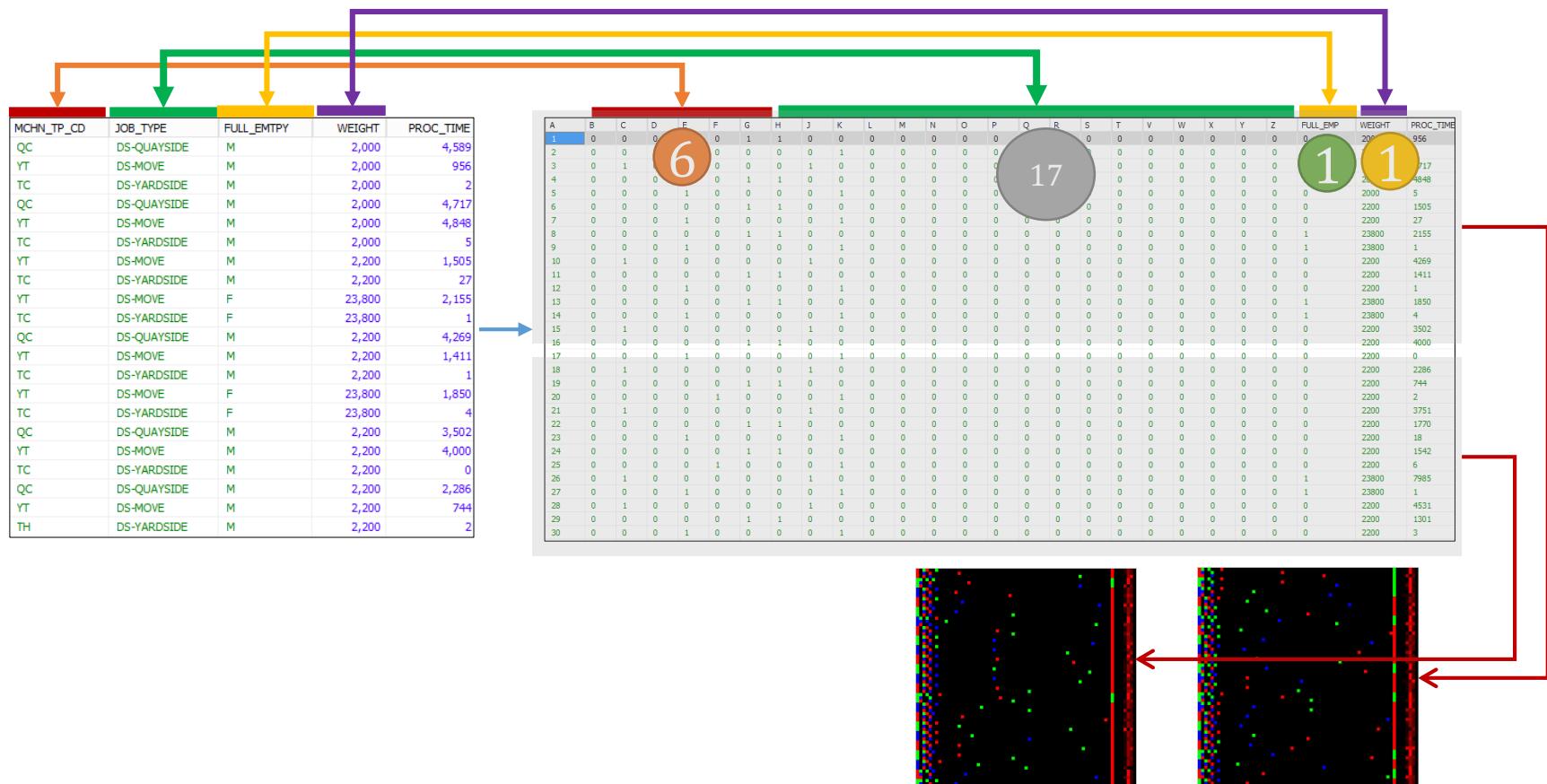


3.4 Data Pixelization

Data Pixelization for predicting completion time of events

- Research Framework

Transforming Event Log into Image

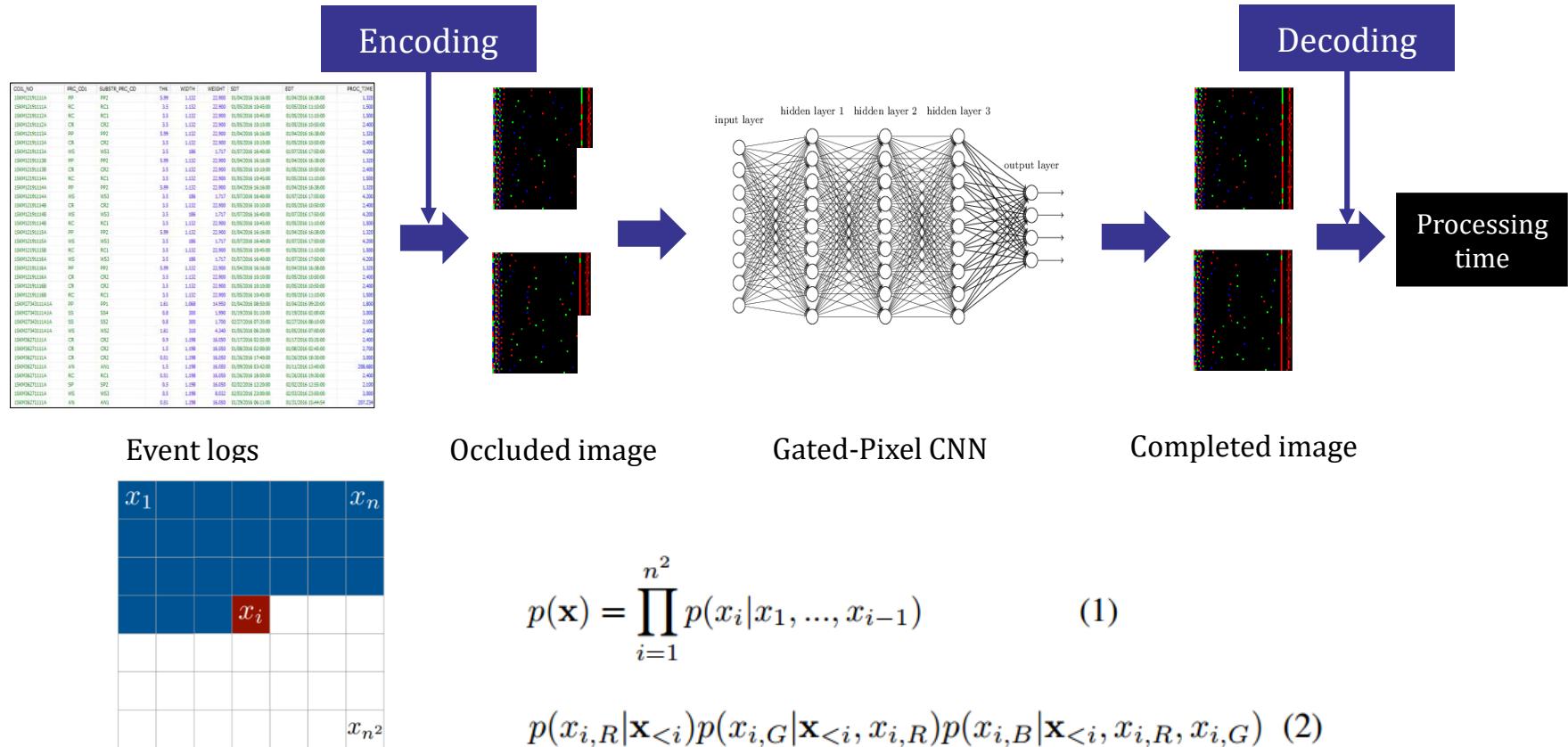


3.4 Data Pixelization

Data Pixelization for predicting completion time of events

- Research Framework

Transforming Event Log into Image

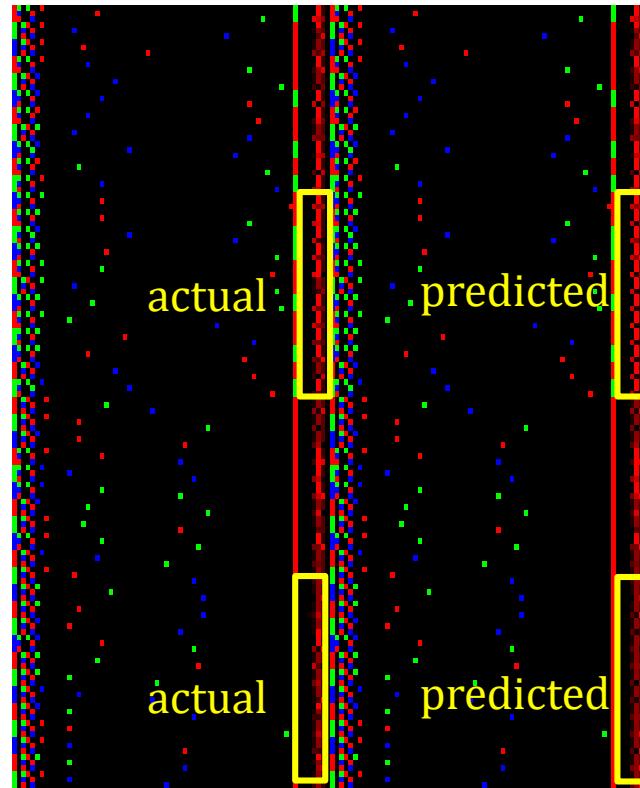
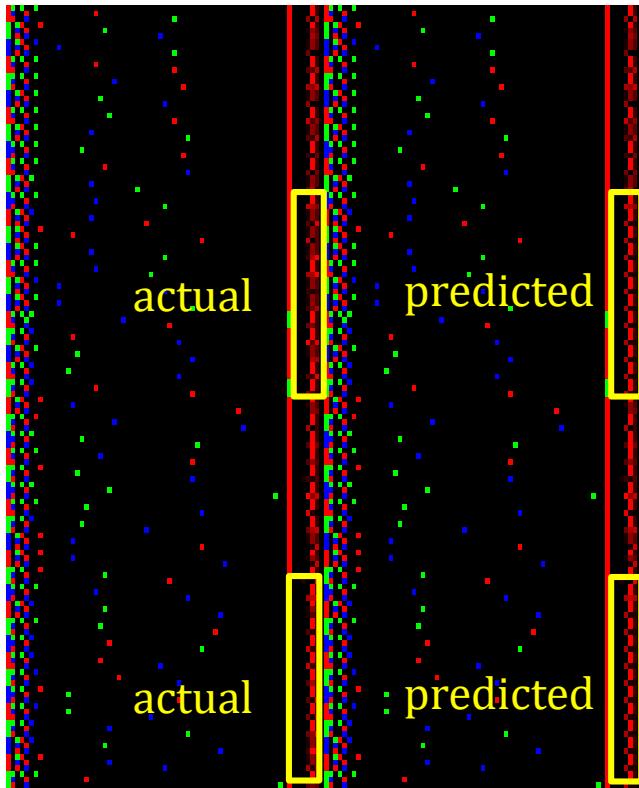


3.4 Data Pixelization

Data Pixelization for predicting completion time of events

- Experimental Result

Compare Predicted and Actual



3.4 Data Pixelization

Data Pixelization for predicting completion time of events

- Performance of Pixelization

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

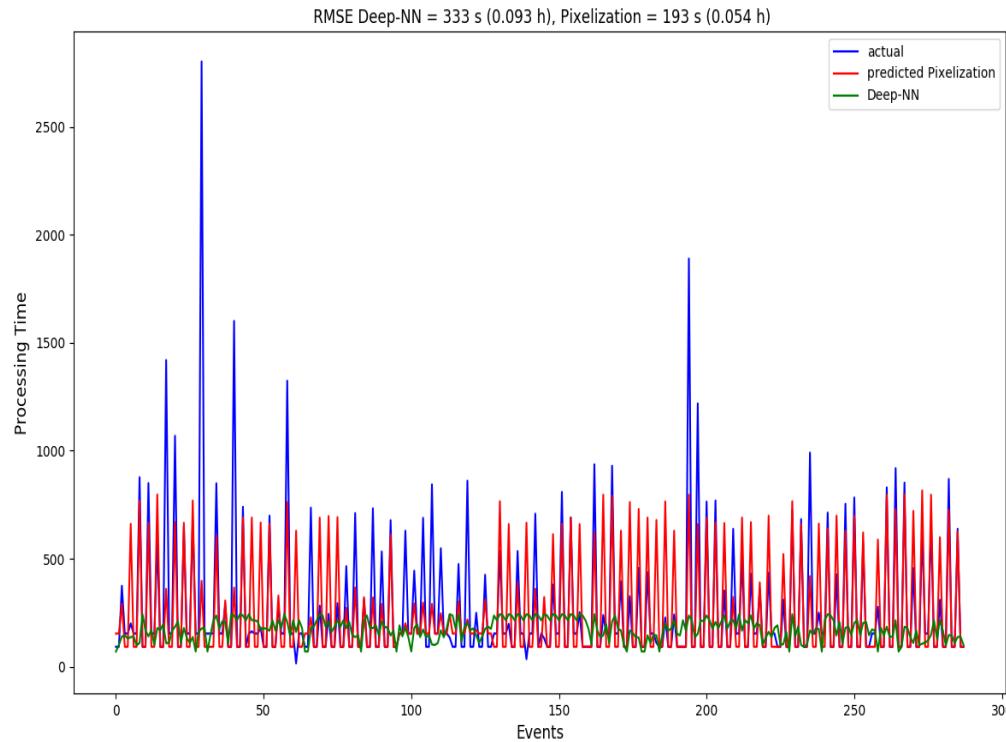
Model	Portion of testing data	Accuracy measure (minute)					
		Event-log SK			Event-log MA		
		RMSE	MAE	MAPE	RMSE	MAE	MAPE
One-hot Encoding w/o machine-id*	10%	4.365936	1.848417	1.763972	5.565939	2.518422	2.963112
	20%	4.377116	1.837501	1.755248	5.523901	2.437655	2.855255
	30%	4.390820	1.833780	1.751794	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.

3.4 Data Pixelization

Data Pixelization for predicting completion time of events

- Comparison between DNN and Our Approach

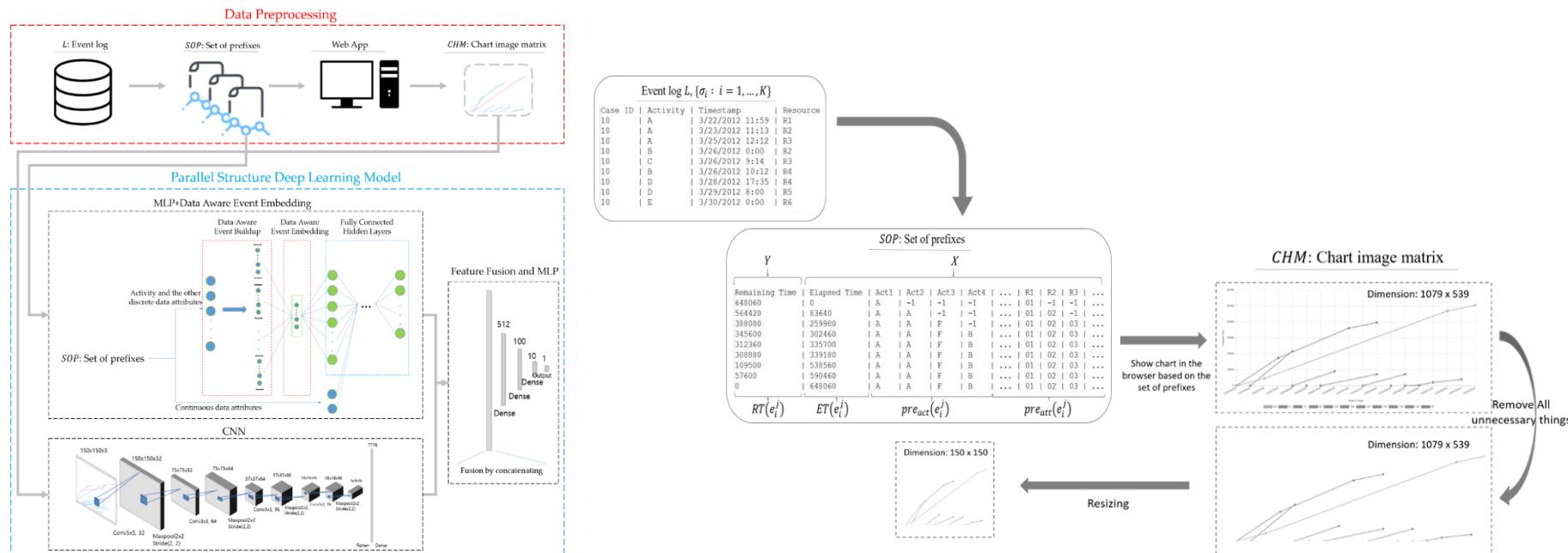


Deep-NN cannot perform well
with categorical data !

3.4 Data Pixelization

Data Pixelization for predicting completion time of events

- Combining Imagification and Embedding



3.4 Data Pixelization

Data Pixelization for predicting completion time of events

- DAEE

```
[ 0.858 -0.665  1.181 -0.296 -0.654 -0.861 -0.207 -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  0.639  0.793  0.600  0.018 -1.735  0.528 -0.529 -0.463
1.384 -0.573 -0.753 -0.244 -1.124  2.562  0.401 -0.358  0.879  0.429 ]
[ -0.149  5.135  1.492  3.464 -1.714 -0.567 -2.918  2.405  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 -0.009  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 ] ]
```

[Activity, Resource, Part Desc, Worker, Report Type, Rework]
[19 10 8 26 2 0]

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

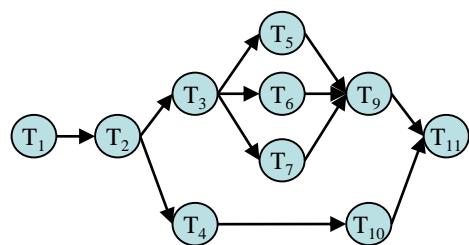
Mean training duration and MAE in units of minutes across five iterations of CV

	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

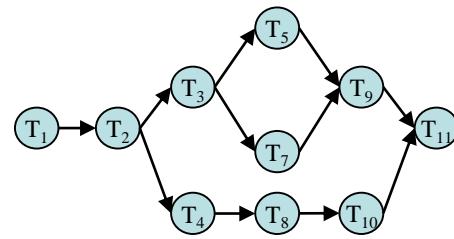
3.5 Process Similarity Measure

Distance-based Process Similarity Measures

- Introduction



How Similar each other?

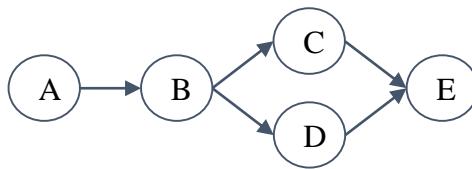


3.5 Process Similarity Measure

Distance-based Process Similarity Measures

- **(Recall) What is DFG**

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

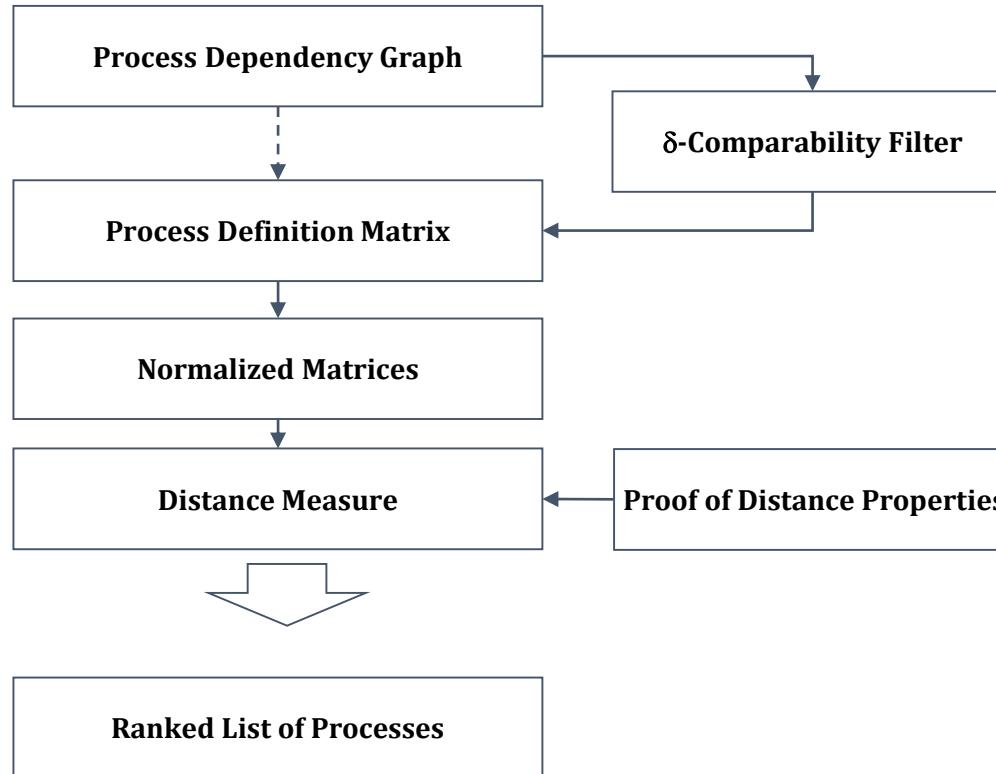


- $DN = \{A, B, C, D, E\}$
- $DE = \{A \rightarrow B, B \rightarrow C, B \rightarrow D, C \rightarrow E, D \rightarrow E\}$

3.5 Process Similarity Measure

Distance-based Process Similarity Measures

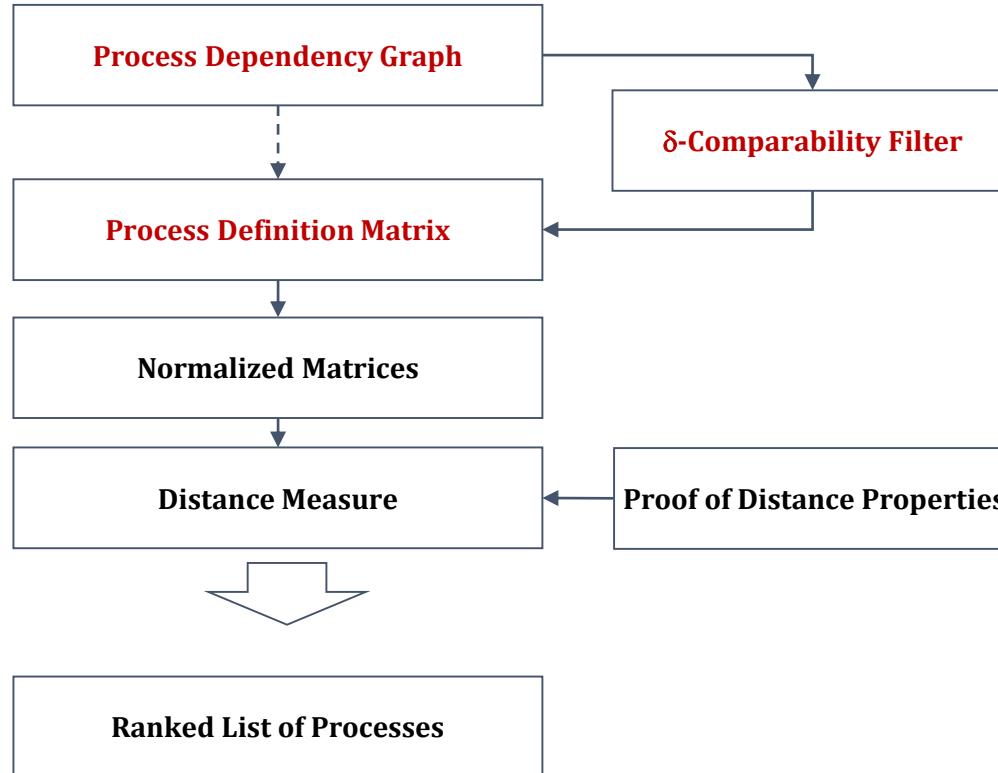
- Input and Output



3.5 Process Similarity Measure

Distance-based Process Similarity Measures

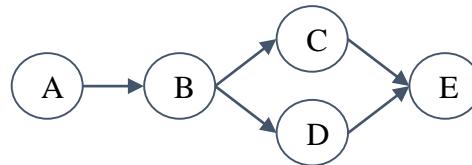
- Input and Output



3.5 Process Similarity Measure

Distance-based Process Similarity Measures

- Process Definition Matrix - Numerical Representation



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

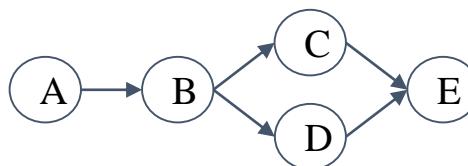
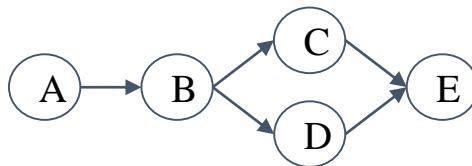
M	TO					
		A	B	C	D	E
F	0	1	0	0	0	0
R	0	0	1	1	0	0
O	0	0	0	0	1	0
M	0	0	0	0	1	0
	0	0	0	0	0	0

3.5 Process Similarity Measure

Distance-based Process Similarity Measures

- **Identical Dependency Graph**

- 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



3.5 Process Similarity Measure

Distance-based Process Similarity Measures

- Comparison Matrices

- 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

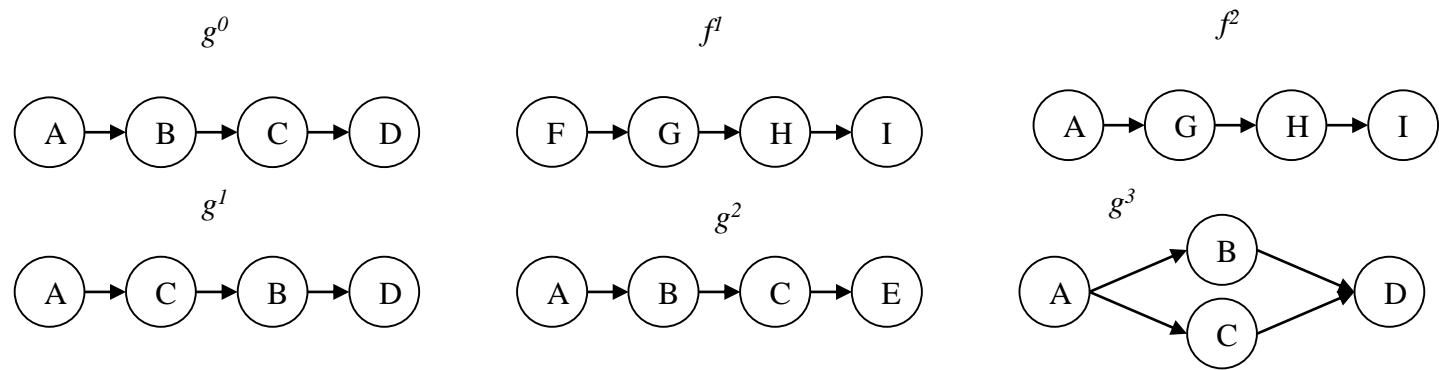
$$\frac{|DN_1 \cap DN_2|}{|DN_1 \cup DN_2|} \geq \delta \text{ holds, where } 0 < \delta \leq 1 \quad \blacksquare$$

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

3.5 Process Similarity Measure

Distance-based Process Similarity Measures

- Comparison Matrices
- Examples of δ -Comparability $\delta=0.5$



- g^0 and f^l 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

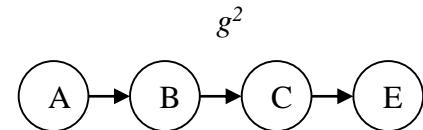
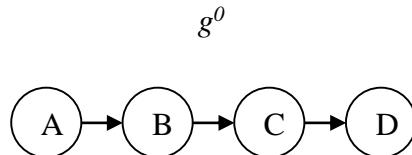
$$\frac{|DN_1 \cap DN_2|}{|DN_1 \cup DN_2|} = \frac{3}{5} \geq 0.5$$

3.5 Process Similarity Measure

Distance-based Process Similarity Measures

- 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.



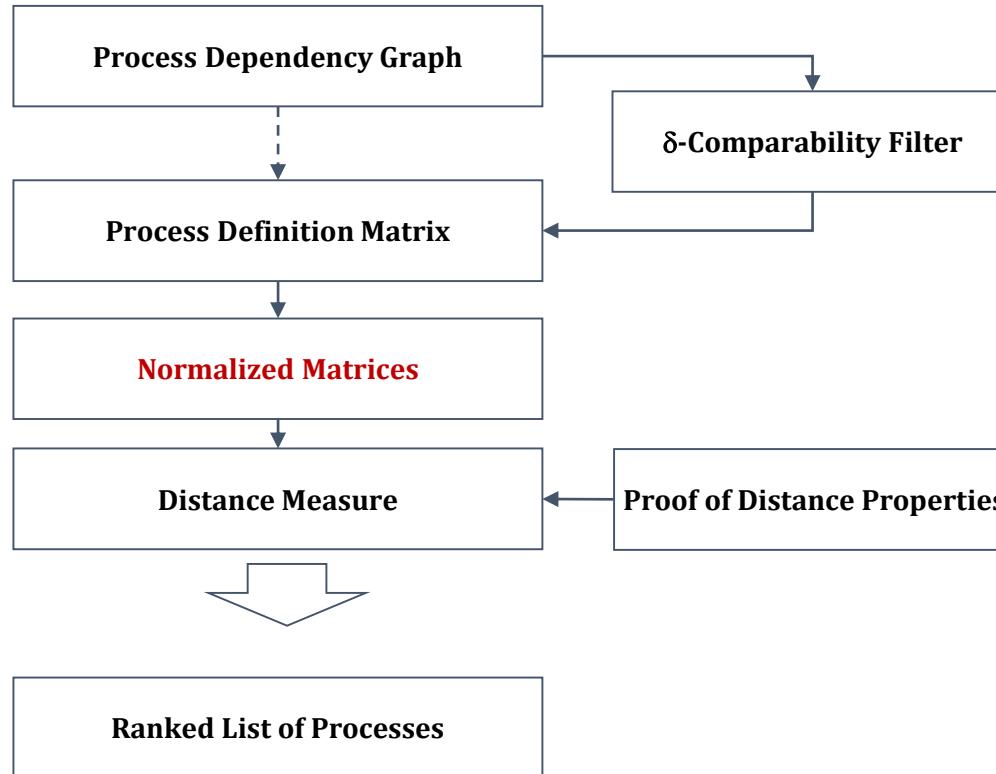
NM_1	TO				
F R O M		A	B	C	D
	A	0	1	0	0
	B	0	0	1	0
	C	0	0	0	1
	D	0	0	0	0

NM_2	TO				
F R O M		A	B	C	E
	A	0	1	0	0
	B	0	0	1	0
	C	0	0	0	1
	E	0	0	0	0

3.5 Process Similarity Measure

Distance-based Process Similarity Measures

- Input and Output



3.5 Process Similarity Measure

Distance-based Process Similarity Measures

- **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

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

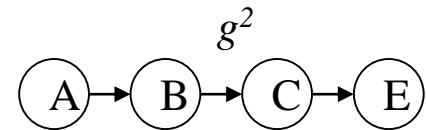
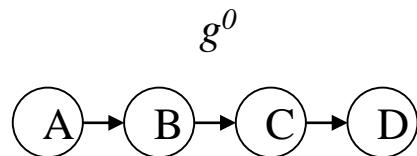
- ii) Let $DN_1 \cup DN_2 = \{A_1, A_2, \dots, A_m\}$
- iii) Let $NM_1(i,j)$ and $NM_2(i,j)$ denote the value of the i^{th} row and the j^{th} column in NM_1 , NM_2

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

3.5 Process Similarity Measure

Distance-based Process Similarity Measures

- Normalized Matrix, NM



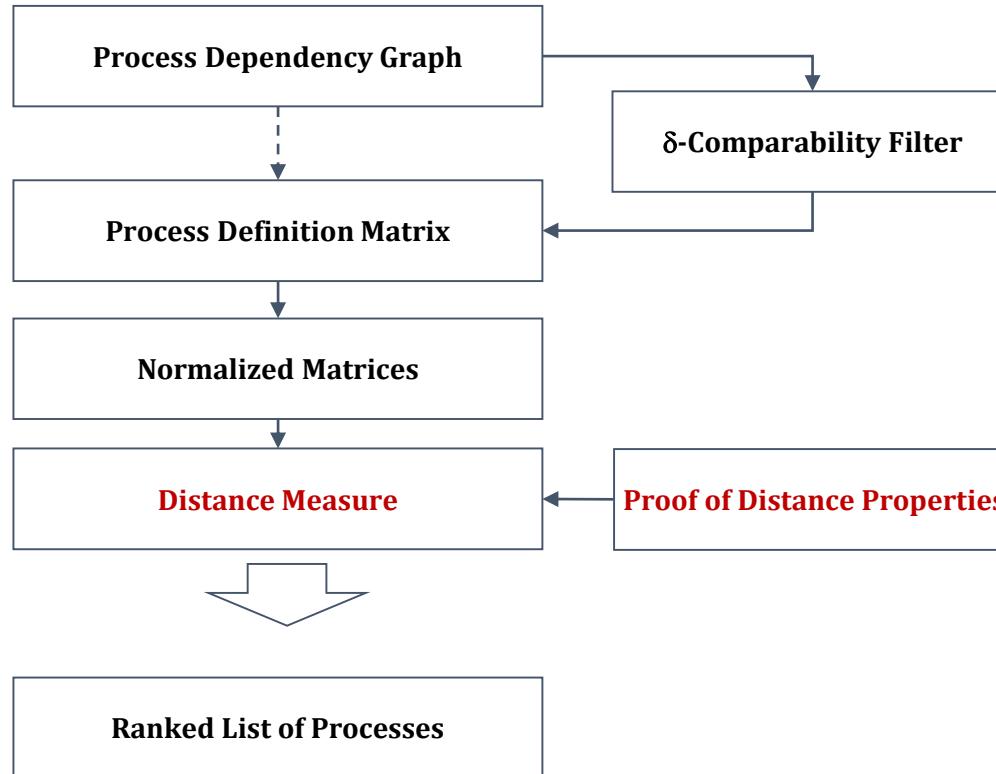
NM_1		TO				
F R O M		A	B	C	D	E
	A	0	1	0	0	0
	B	0	0	1	0	0
	C	0	0	0	1	0
	D	0	0	0	0	0
	E	0	0	0	0	0

NM_2		TO				
F R O M		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

3.5 Process Similarity Measure

Distance-based Process Similarity Measures

- Input and Output



3.5 Process Similarity Measure

Distance-based Process Similarity Measures

- Normalized Matrix, NM
- 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 $NM_1 - NM_2$

3.5 Process Similarity Measure

Distance-based Process Similarity Measures

- Normalized Matrix, NM

$$(NM_1 - NM_2)(NM_1 - NM_2)^T = \begin{bmatrix} & A & B & C & D & E \\ A & 0 & 0 & 0 & 0 & 0 \\ B & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & 1 & -1 \\ D & 0 & 0 & 0 & 0 & 0 \\ E & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} & A & B & C & D & E \\ A & 0 & 0 & 0 & 0 & 0 \\ B & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & 0 & 0 \\ D & 0 & 0 & 1 & 0 & 0 \\ E & 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$
$$= \begin{bmatrix} & A & B & C & D & E \\ A & 0 & 0 & 0 & 0 & 0 \\ B & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 1^2 + (-1)^2 & 0 & 0 \\ D & 0 & 0 & 0 & 0 & 0 \\ E & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \rightarrow 2$$

3.5 Process Similarity Measure

Distance-based Process Similarity Measures

- Normalized Matrix, NM

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

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

3.5 Process Similarity Measure

Distance-based Process Similarity Measures

- Trace, Inner Product of Matrix
- 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.,

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

where A_{ij} represents the (i, j) th element of A ■

- Inner Product of Matrix
 - For an m -by- n matrix A with complex (or real) entries, we have
 - $$\langle A, A^T \rangle = tr(A \times A^T) = \sum_{i=1}^n \sum_{j=1}^n a_{ij}^2 \geq 0$$
 - with the trace of A equals to 0 only if $A=0$. ■

3.5 Process Similarity Measure

Distance-based Process Similarity Measures

- Dependency Difference Metrix
- 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:

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

where $\text{tr}[\cdot]$ denotes the trace of a matrix, i.e., the sum of the diagonal elements. ■

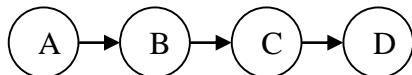
- That is, new difference metric d is inner product of NM_1 - NM_2

3.5 Process Similarity Measure

Distance-based Process Similarity Measures

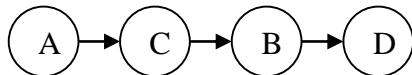
- Dependency distance measurements
- Examples of dependency distance measurements

g^0

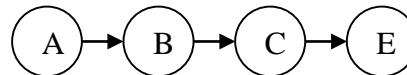


$$d(g^0, g^1) = 6, d(g^0, g^2) = 2, d(g^0, g^3) = 3$$

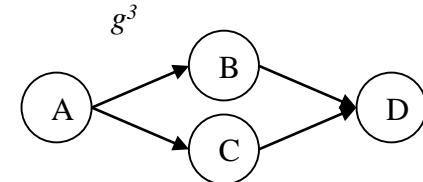
g^1



g^2



g^3

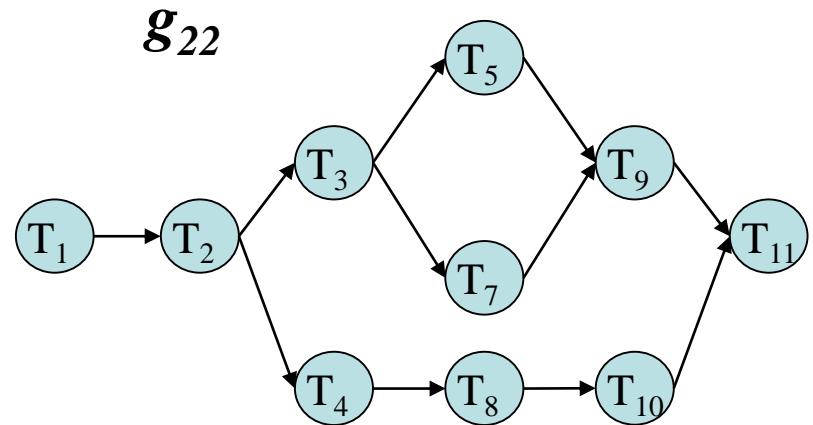
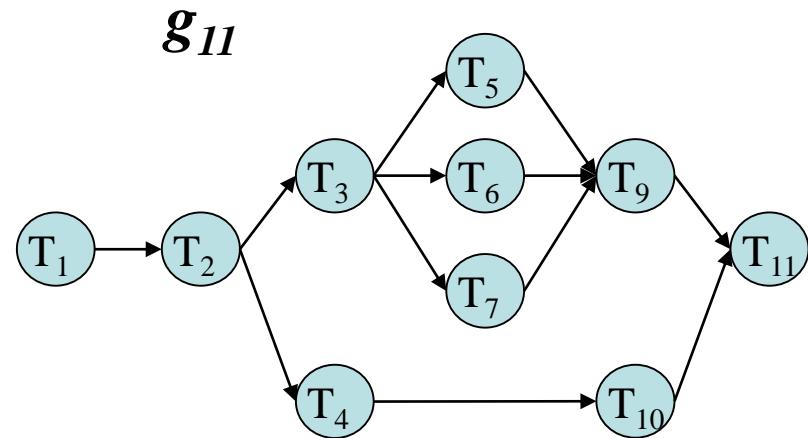


- The distance of dependency between g^0 and g^1 is 6, the distance of g^0 and g^2 is 2, and the distance of g^0 and g^3 is 3.
- This means that g^0 and g^2 are the most similar, which is intuitively correct because the first three activities are in the same sequence but only the last activity is different.
- 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.

3.5 Process Similarity Measure

Distance-based Process Similarity Measures – Extended Example

- Compare two graphs

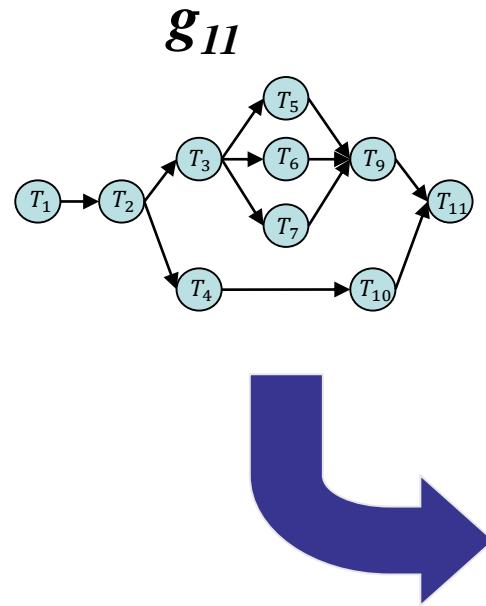


$$d(g_{11}, g_{22}) = ?$$

3.5 Process Similarity Measure

Distance-based Process Similarity Measures – Extended Example

- Process Definition Matrix of M_{11}

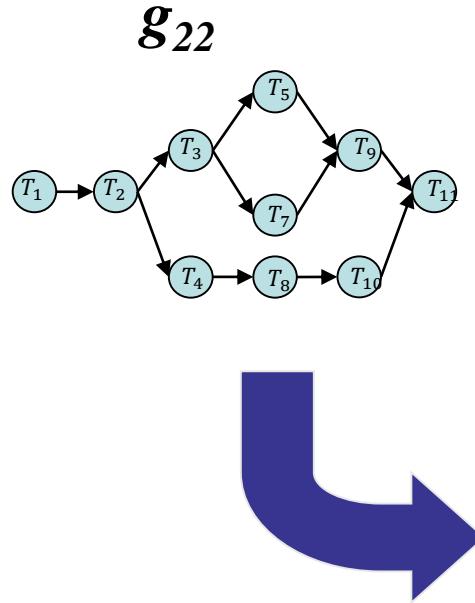


	TO										
	T1	T2	T3	T4	T5	T6	T7	T9	T10	T11	
T1	0	1	0	0	0	0	0	0	0	0	
T2	0	0	1	1	0	0	0	0	0	0	
T3	0	0	0	0	1	1	1	0	0	0	
T4	0	0	0	0	0	0	0	0	1	0	
T5	0	0	0	0	0	0	0	1	0	0	
T6	0	0	0	0	0	0	0	1	0	0	
T7	0	0	0	0	0	0	0	1	0	0	
T9	0	0	0	0	0	0	0	0	0	1	
T10	0	0	0	0	0	0	0	0	0	1	
T11	0	0	0	0	0	0	0	0	0	0	

3.5 Process Similarity Measure

Distance-based Process Similarity Measures – Extended Example

- Process Definition Matrix of M_{22}

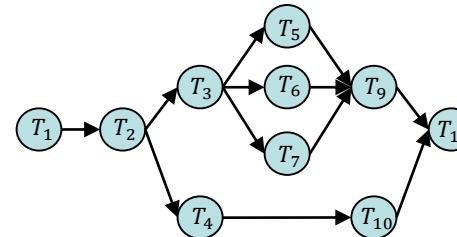
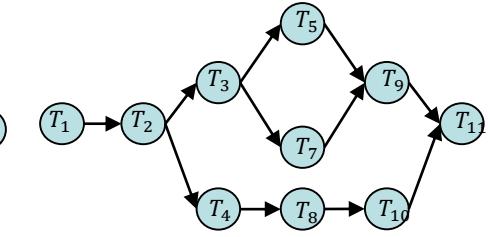


	TO										
	T1	T2	T3	T4	T5	T7	T8	T9	T10	T11	
T1	0	1	0	0	0	0	0	0	0	0	
T2	0	0	1	1	0	0	0	0	0	0	
T3	0	0	0	0	1	1	0	0	0	0	
T4	0	0	0	0	0	0	1	0	0	0	
T5	0	0	0	0	0	0	0	1	0	0	
T7	0	0	0	0	0	0	0	1	0	0	
T8	0	0	0	0	0	0	0	0	1	0	
T9	0	0	0	0	0	0	0	0	0	1	
T10	0	0	0	0	0	0	0	0	0	1	
T11	0	0	0	0	0	0	0	0	0	0	

3.5 Process Similarity Measure

Distance-based Process Similarity Measures – Extended Example

- Normalized Matrix of NM_{11}

 g_{11}  g_{22} 

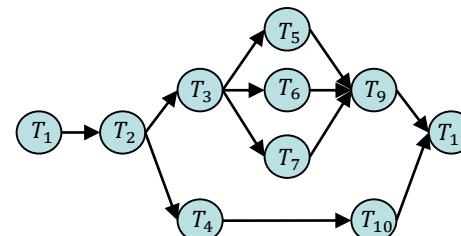
	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
T1	0	1	0	0	0	0	0	0	0	0	0
T2	0	0	1	1	0	0	0	0	0	0	0
T3	0	0	0	0	1	1	1	0	0	0	0
T4	0	0	0	0	0	0	0	0	0	1	0
T5	0	0	0	0	0	0	0	0	1	0	0
T6	0	0	0	0	0	0	0	0	1	0	0
T7	0	0	0	0	0	0	0	0	1	0	0
T8	0	0	0	0	0	0	0	0	0	0	0
T9	0	0	0	0	0	0	0	0	0	0	1
T10	0	0	0	0	0	0	0	0	0	0	1
T11	0	0	0	0	0	0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

3.5 Process Similarity Measure

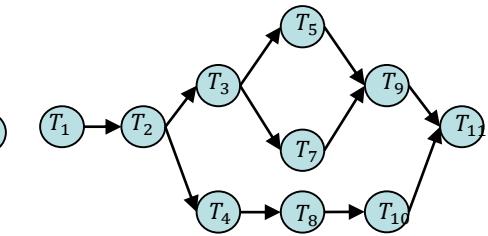
Distance-based Process Similarity Measures – Extended Example

- Normalized Matrix of NM_{22}

g_{11}



g_{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

3.5 Process Similarity Measure

Distance-based Process Similarity Measures – 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

3.5 Process Similarity Measure

Distance-based Process Similarity Measures – 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

3.5 Process Similarity Measure

Distance-based Process Similarity Measures – Extended Example

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

	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	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

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

3.5 Process Similarity Measure

Distance-based Process Similarity Measures

- Application of Distance Matrix

K-Means

KNN

Outlier Process Detection

3.6 Process Analysis with Clustering

Distance-based Process Similarity Measures

- 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

3.6 Process Analysis with Clustering

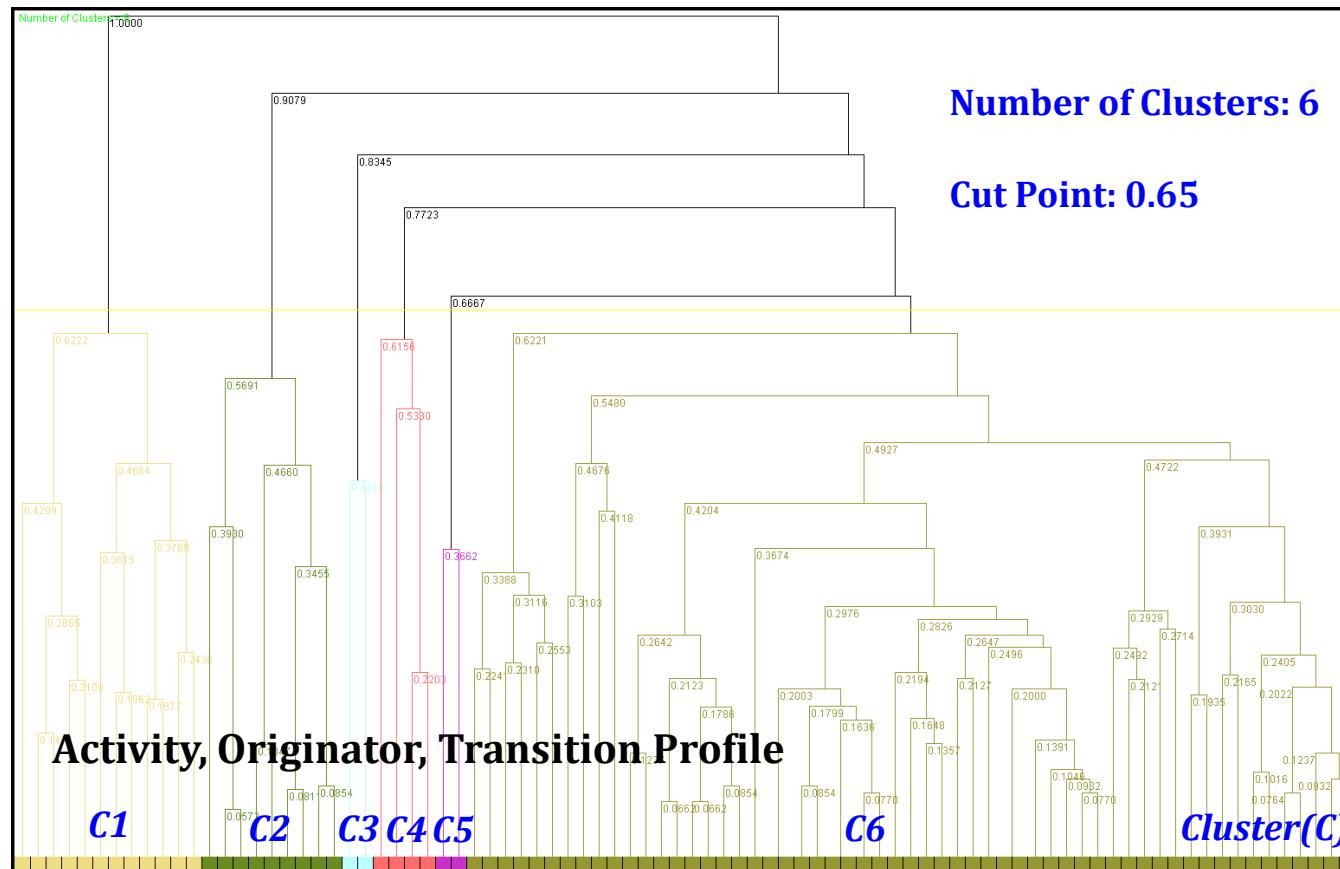
Distance-based Process Similarity Measures – Extended Example

3.6 Process Analysis with Clustering

Analysis of Block Assembly Process

- Clustering

Agglomerative Hierarchical Clustering (Complete linkage) – IRMS data

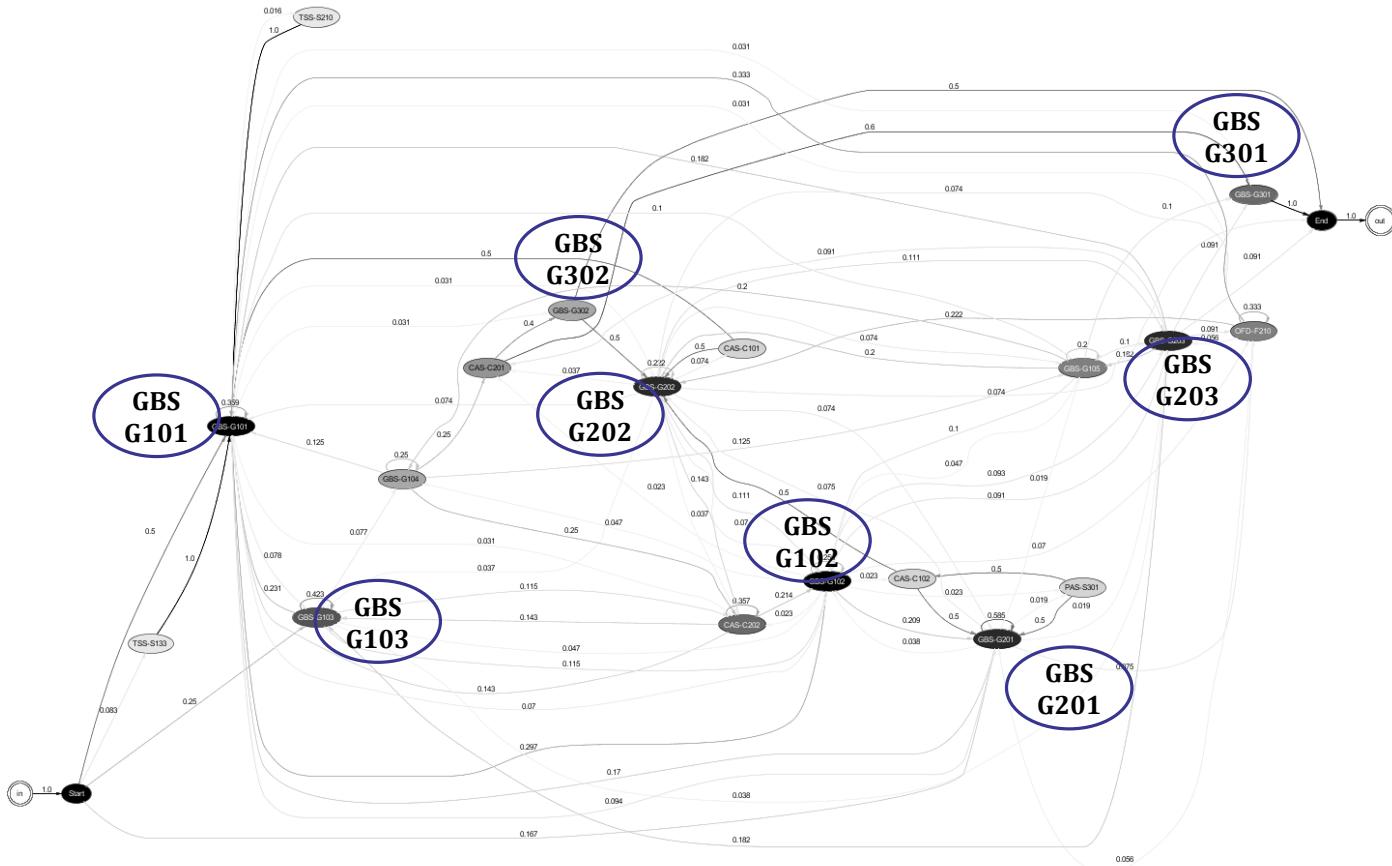


3.6 Process Analysis with Clustering

Analysis of Block Assembly Process

- Cluster 01

Block Assembly in GBS

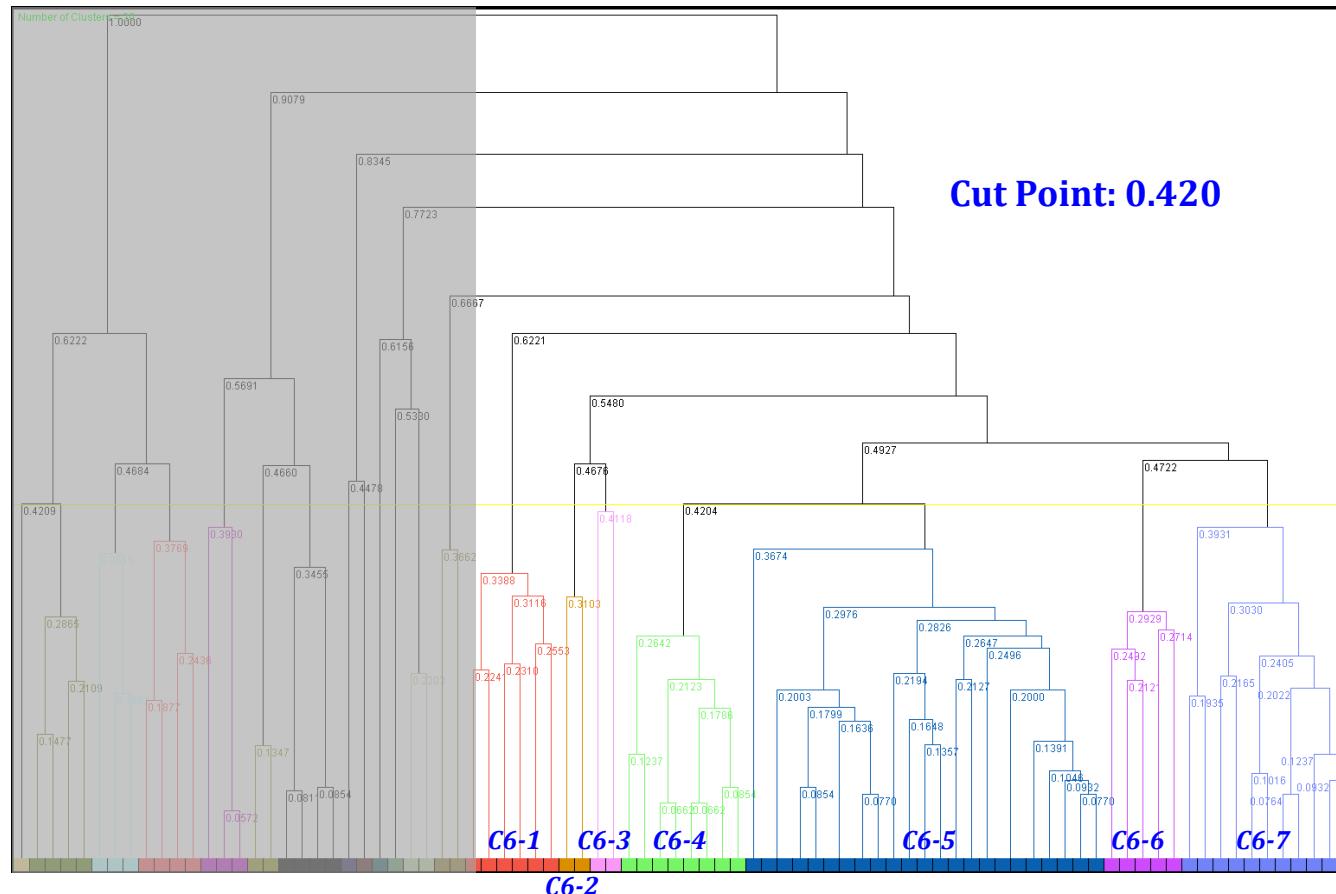


3.6 Process Analysis with Clustering

Analysis of Block Assembly Process

- Clustering

Agglomerative Hierarchical Clustering (Complete linkage) -

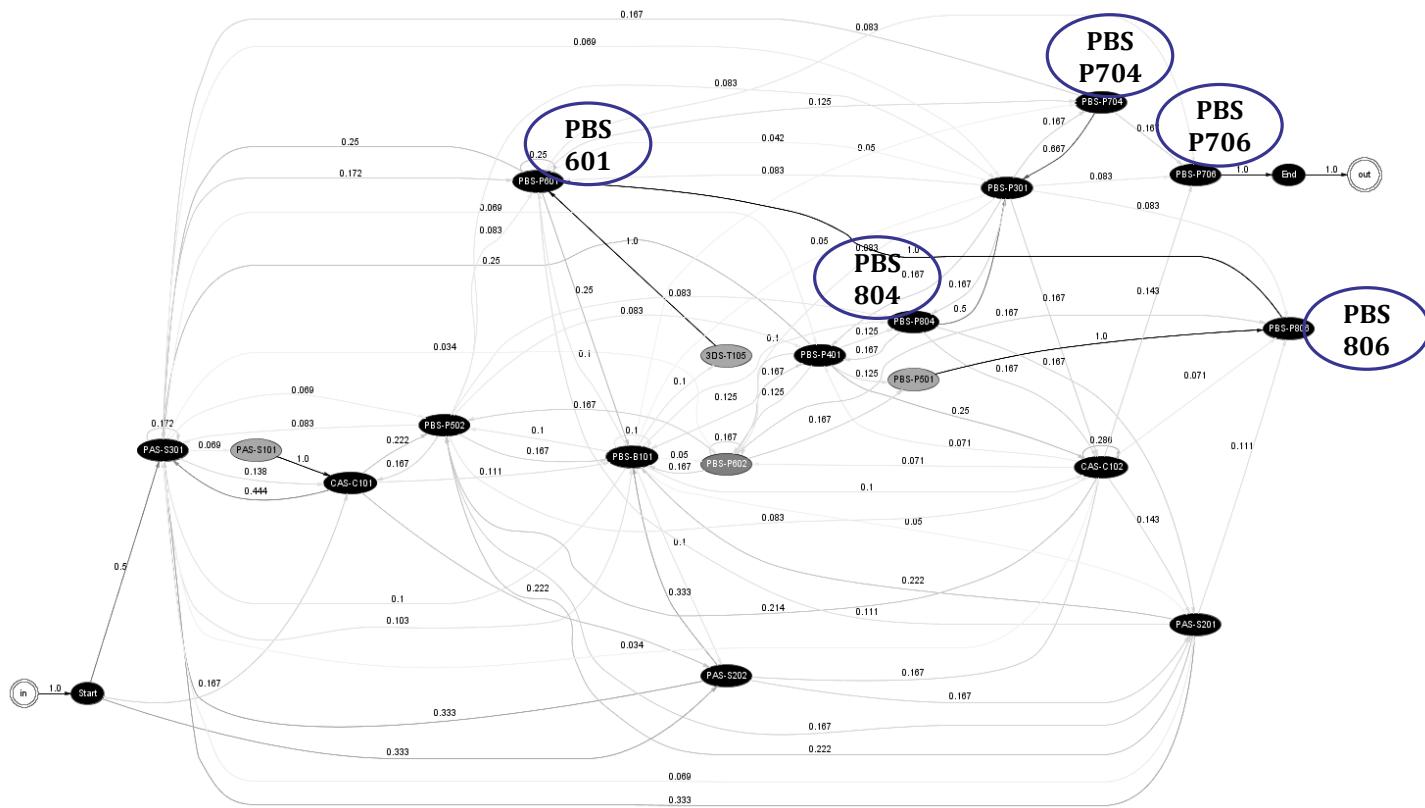


3.6 Process Analysis with Clustering

Analysis of Block Assembly Process

- Cluster 6 - 1

Block Assembly in PBS



3.6 Process Analysis with Clustering

Analysis of Block Assembly Process

- Research Result

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 조립 5공장(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

3.6 Process Analysis with Clustering

Analysis of Block Assembly Process

- Research Result

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(종조 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 곡종조를 공급해서 NPS에서 대조를 완료하는 BLK	DM9-113

3.6 Process Analysis with Clustering

Analysis of Block Assembly Process

- Research Result

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

배량 케이스 기준 군집

- 배량 케이스에 의한 군집: 계획에 없는 단위업무들이 실적에 많이 나타남
- IRMS에 의한 군집: 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

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

IRMS 기준 군집

3.7 Abnormal Process Detection

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’

- **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).

3.7 Abnormal Process Detection

Distance based API Detection : DAPID

- Activity Relation Matrix

CaseID	Event trace
c^1	$\langle a, b, c, d, d \rangle$
c^2	$\langle a, b, c, b, c, d \rangle$
c^3	$\langle a, c, b, c, d, d, d \rangle$
c^4	$\langle a, b, c, b, c, b, c, d, d \rangle$
c^5	$\langle a, b, c, d, d \rangle$
c^6	$\langle a, b, c, b, c, d \rangle$
c^7	$\langle c, b, c, d, d, d \rangle$
c^8	$\langle a, b, c, b, b, c, d, d \rangle$
c^9	$\langle a, b, c, d, d \rangle$
c^{10}	$\langle a, b, c, b, c, d \rangle$
c^{11}	$\langle a, c, b, c, d, d, d \rangle$
c^{12}	$\langle a, b, c, b, c, b, c, d, d \rangle$
c^{13}	$\langle a, b, c, d, d \rangle$
c^{14}	$\langle a, b, c, b, c, d \rangle$
c^{15}	$\langle c, b, c, d, d, d \rangle$
c^{16}	$\langle a, b, c, b, c, b, c, d, d \rangle$
c^{17}	$\langle a, b, c, d, d \rangle$
c^{18}	$\langle a, b, b, c, d \rangle$
c^{19}	$\langle a, c, b, c, d, d, d \rangle$
c^{20}	$\langle a, b, c, b, c, b, c, d \rangle$

Definition :

e : Event

a_i : Activity

c^l : Caseid

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

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

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

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

3.7 Abnormal Process Detection

Distance based API Detection : DAPID

- Algorithm Explanation

Definition:

\mathcal{C} : Set of cases

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

T : Total number of cases

r : Distance threshold ($r > 0$)

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

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

Algorithm: Detection of API using arc matrix

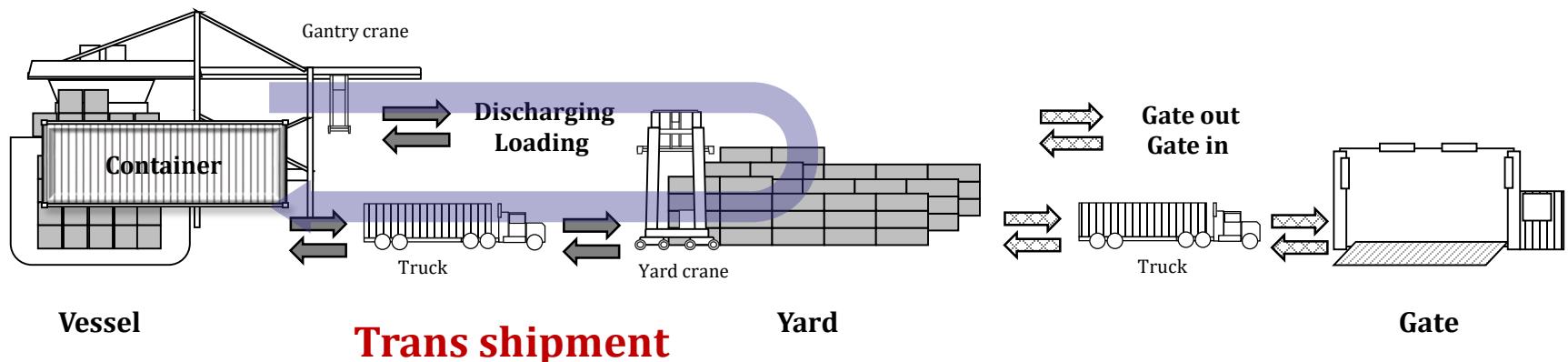
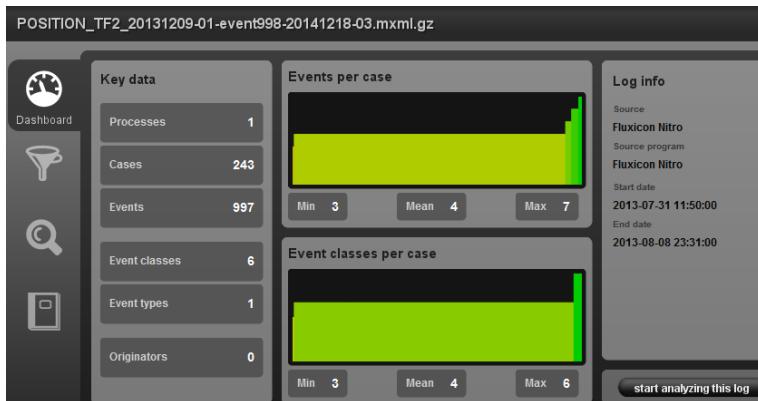
- Input data:
A set of Cases $\mathcal{C} = \{c^1, \dots, c^T\}$
Threshold r ($r > 0$) and π ($0 < \pi \leq 1$)
- Output: Distance-based (r, π) anomaly in \mathcal{C}
- Method:
 - 1: **for** int i=0; i< T; i++ (Base instance)
 - 2: **for** int j=0; j< T; j++ (Target instance)
 - 3: **if** $d_{ij} < r$
 - 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;

3.7 Abnormal Process Detection

Distance based API Detection : DAPID

- Case Study

Event of Container Terminal Operation System

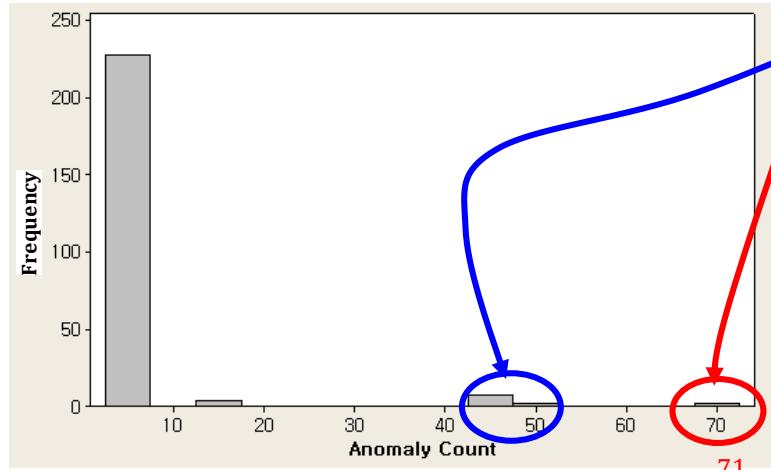


3.7 Abnormal Process Detection

Distance based API Detection : DAPID

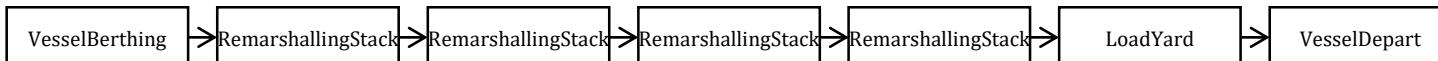
- Case Study

Max Anomaly Sum of Instance

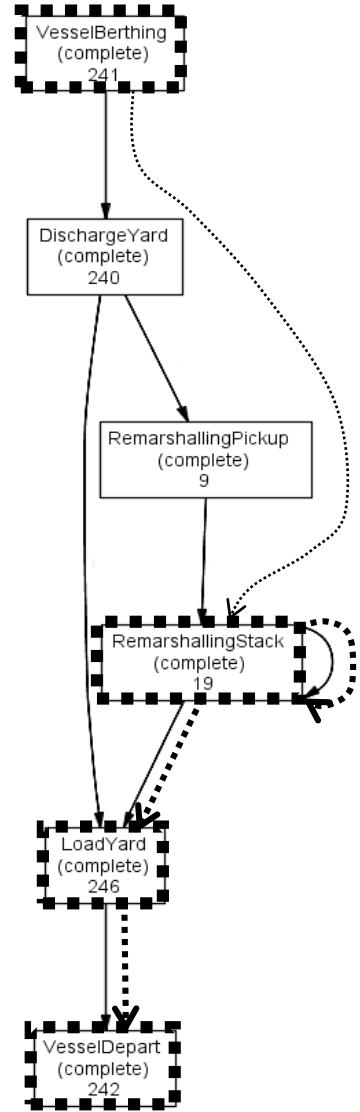
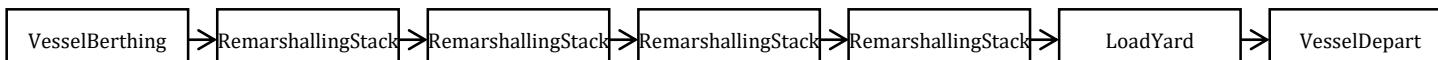


Caseid	Anomaly Sum
Container 197	71
Container 198	71
Container 060	51
Container 167	51
Container 015	44
Container 013	43
Container 014	43
Container 017	43
Container 018	43
Container 019	43
Container 020	43
⋮	
Average	9.095
Standard Deviation	9.213
Maximum	71
Minimum	7

Container 197



Container 198



3.7 Abnormal Process Detection

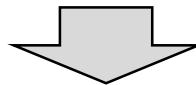
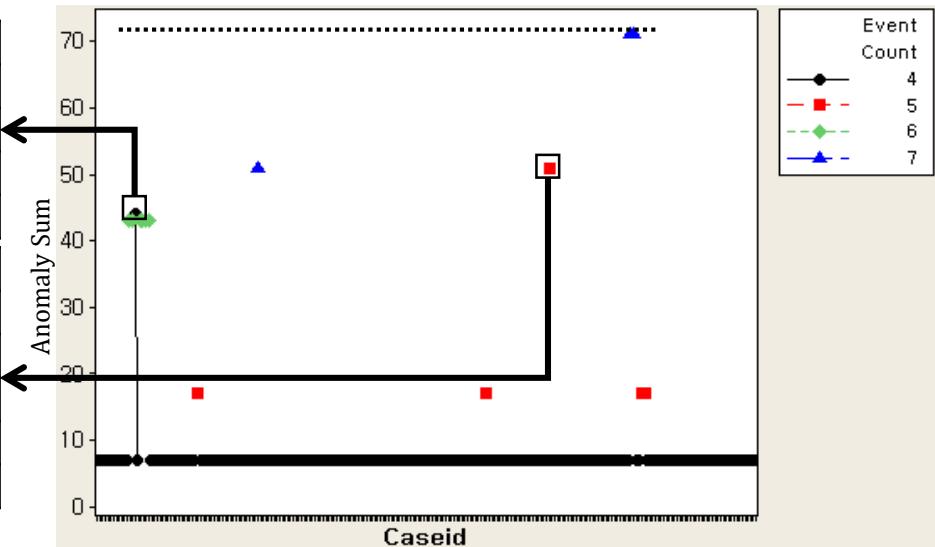
Distance based API Detection : DAPID

- Case Study

Needs for discovering “Container 015” and “Container 167”

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

3.7 Abnormal Process Detection

Local API Detection : LAPID

- Algorithm Explanation

Input data: $M = \{M^1, \dots, M^l\}$: 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

$blrд = 0$: local reachability density of base arc matrix

$tlrd = 0$: local reachability density of target arc matrix

$sum_lrд = 0$: 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** $d^{ij} < d_k^i (i \neq j)$

4: $C \leftarrow M^j;$

5: $n = |C|;$

6: **for** int $j=0; j < n; j++$

7: **if** $d_k^j > d^{ij} (i \neq j)$

8: $rd += d_k^j;$

9: **else if** $d_k^j < d^{ij} (i \neq j)$

10: $rd += d^{ij};$

11: $blrд = n/rd;$

12: $C = \{\};$

13: **for** int $j=0; j < n; j++$

14: $rd = 0;$

15: **for** int $p=0; p < l; p++$

16: **if** $d^{jp} < d_k^j (j \neq p)$

17: $C \leftarrow M^j;$

18: $m = |C|;$

19: **for** int $p=0; p < m; p++$

20: **if** $d_k^p > d^{jp} (j \neq p)$

21: $rd += d_k^p;$

22: **else if** $d_k^p < d^{jp} (j \neq p)$

23: $rd += d^{jp};$

24: $blrд = m/rd;$

25: $sum_lrд += tlrd/blrd;$

26: $lof = sum_lrд/n;$

27: **for** int $j=0; j < 10; j++$

28: **if** $lof > array_lof[j]$

29: $array_lof[j] \leftarrow lof;$

3.7 Abnormal Process Detection

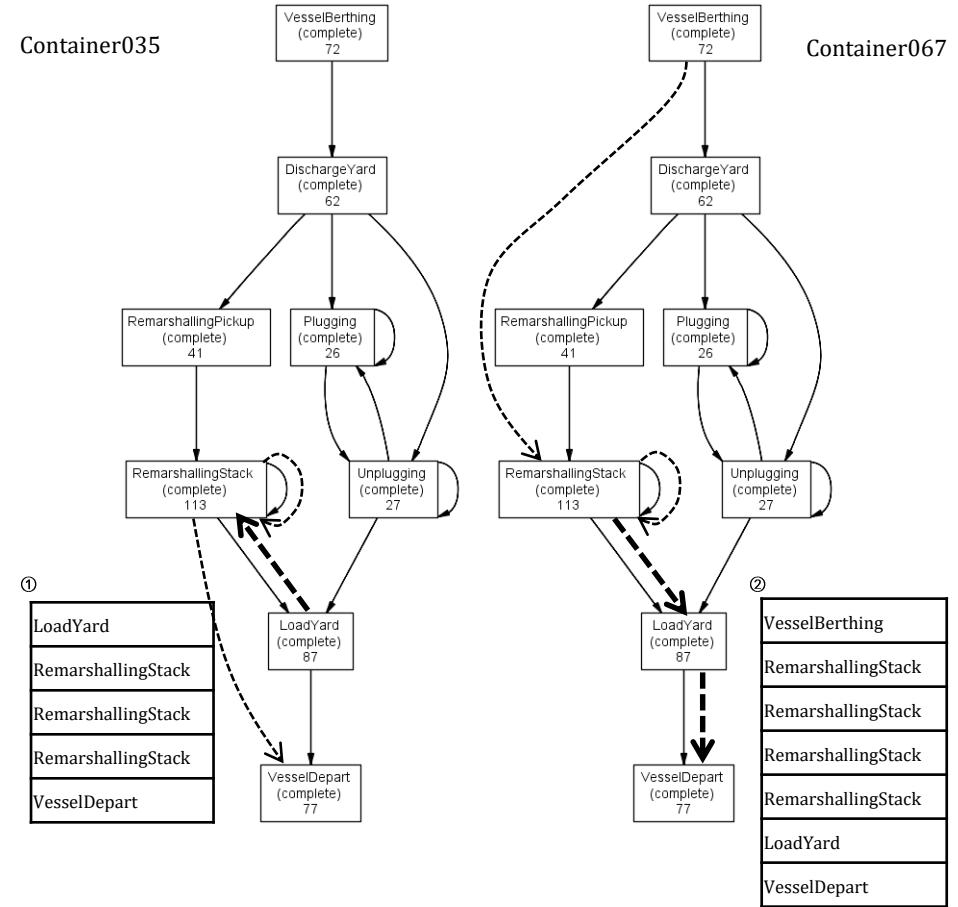
Local API Detection : LAPID

- Case Study

LAPID - Top 20

	DAPID	Results	LAPID	k=30
⑤	Container058	80	Container058	4.1484
⑤	Container041	80	Container041	3.6188
①	Container062	77	Container059	3.0281
③	Container035	76	Container062	3.0109
③	Container037	76	Container054	2.9301
③	Container054	76	Container011	2.5824
③	Container055	72	Container012	2.5824
③	Container056	72	Container015	2.5824
③	Container059	72	Container023	2.5824
④	Container063	72	Container065	2.5824
④	Container064	72	Container067	2.5824
④	Container022	71	Container070	2.5824
④	Container033	70	Container071	2.5824
④	Container077	70	Container078	2.5824
④	Container072	69	Container040	2.4889
④	Container011	66	Container033	2.3192
④	Container012	66	Container077	2.3192
④	Container015	66	Container055	2.2922
④	Container023	66	Container056	2.2922
④	Container065	66	Container063	2.2922

- ① ② New entry
- ③ Slope down
- ④ Slope up

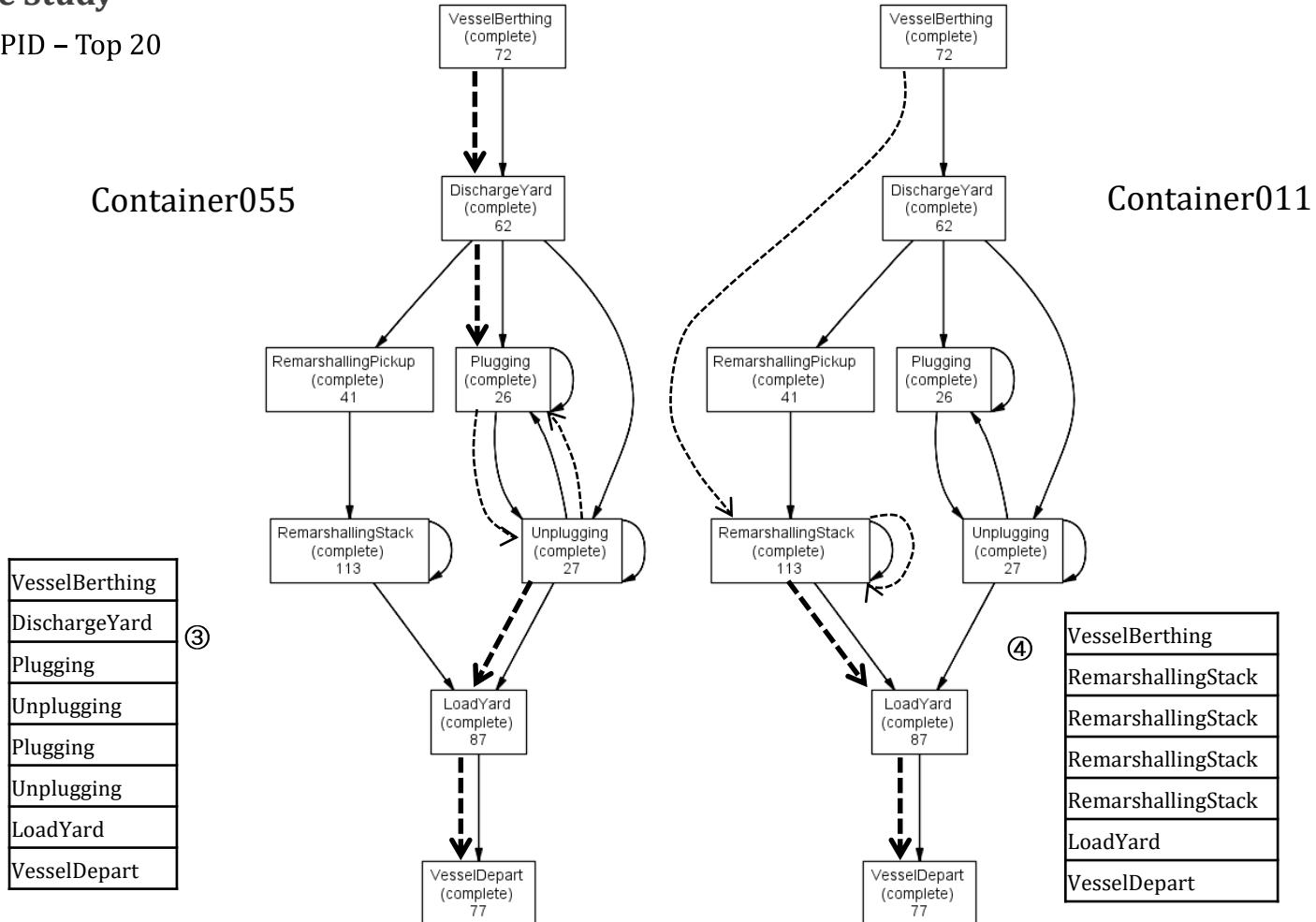


3.7 Abnormal Process Detection

Local API Detection : LAPID

- Case Study

LAPID – Top 20

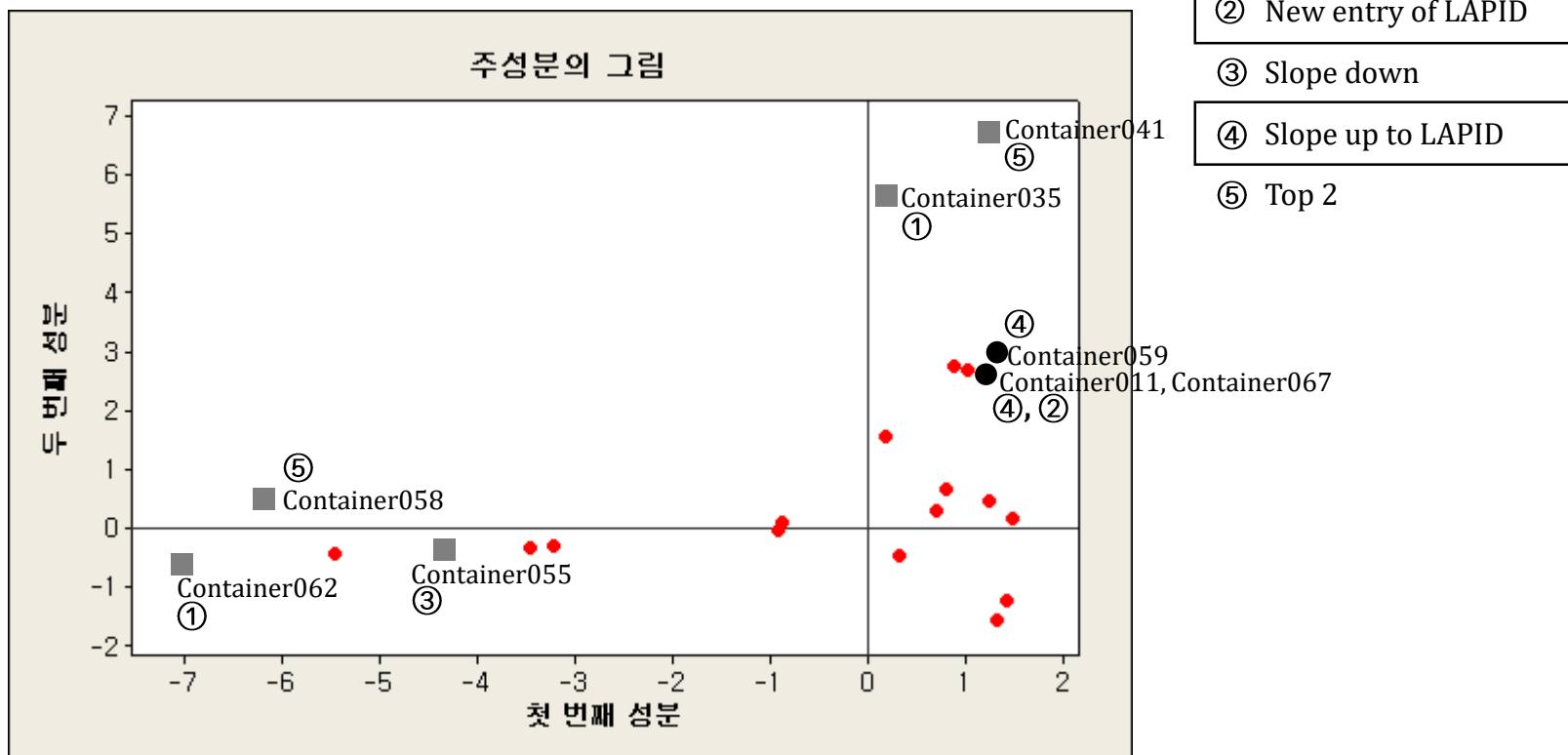


3.7 Abnormal Process Detection

Comparing DAPID & LAPID

- Case Study

Principal Component Analysis



3.7 Abnormal Process Detection

LAPID using Manufacturing Data

- Case Study

8,285 events & 236 CaseID

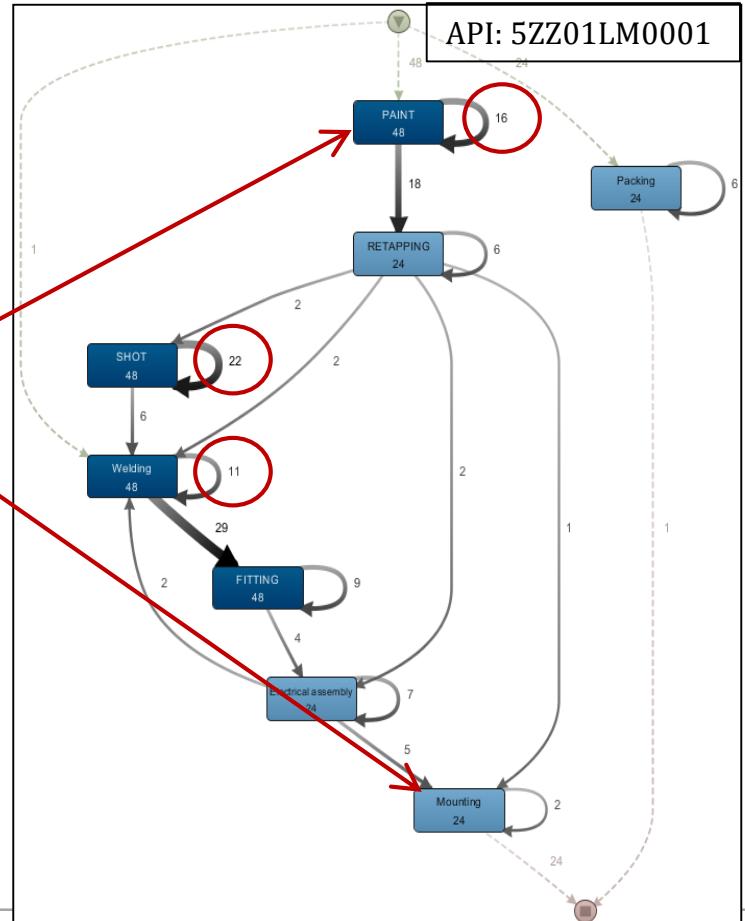
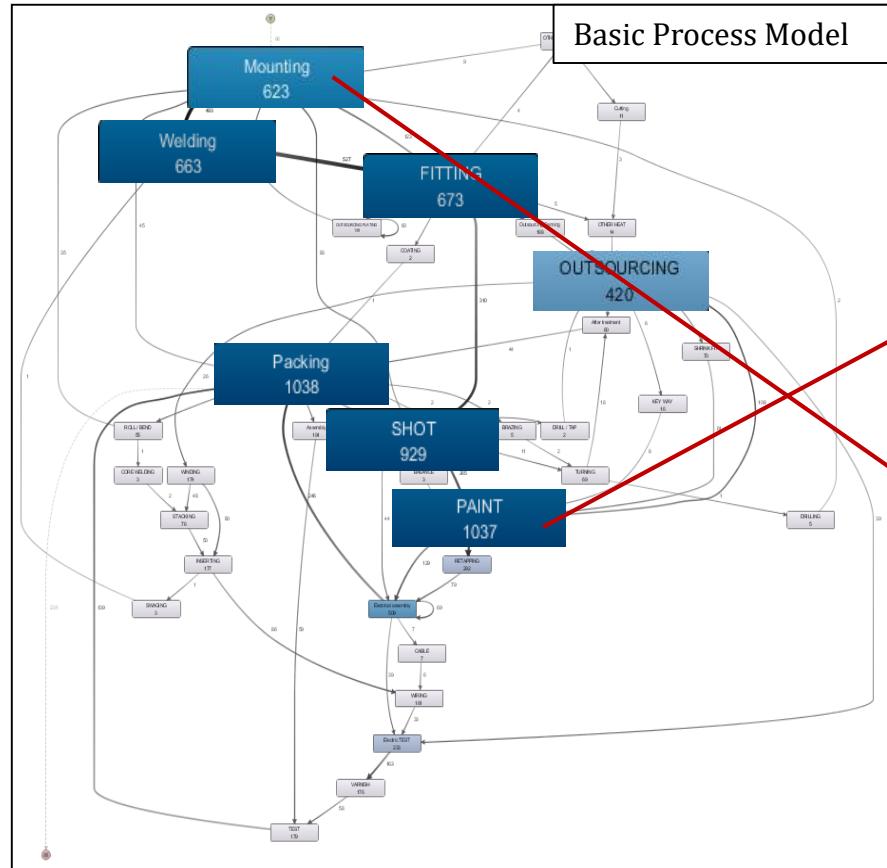
Caseid	(k=10)	rank	(k=15)	rank	(k=20)	rank
5ZZ01LM0001	5.150704838	1	5.559636419	1	5.585752894	1
43246RPM054	3.015462114	2	2.437775872	8	2.106783259	19
44257RAL633	2.938925109	3	2.464184825	7	2.207233025	11
40952RAS001	2.934231297	4	2.572686477	4	2.37872769	5
32301RAS012	2.900885409	5	2.152198964	19	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



3.7 Abnormal Process Detection

LAPID using Manufacturing Data

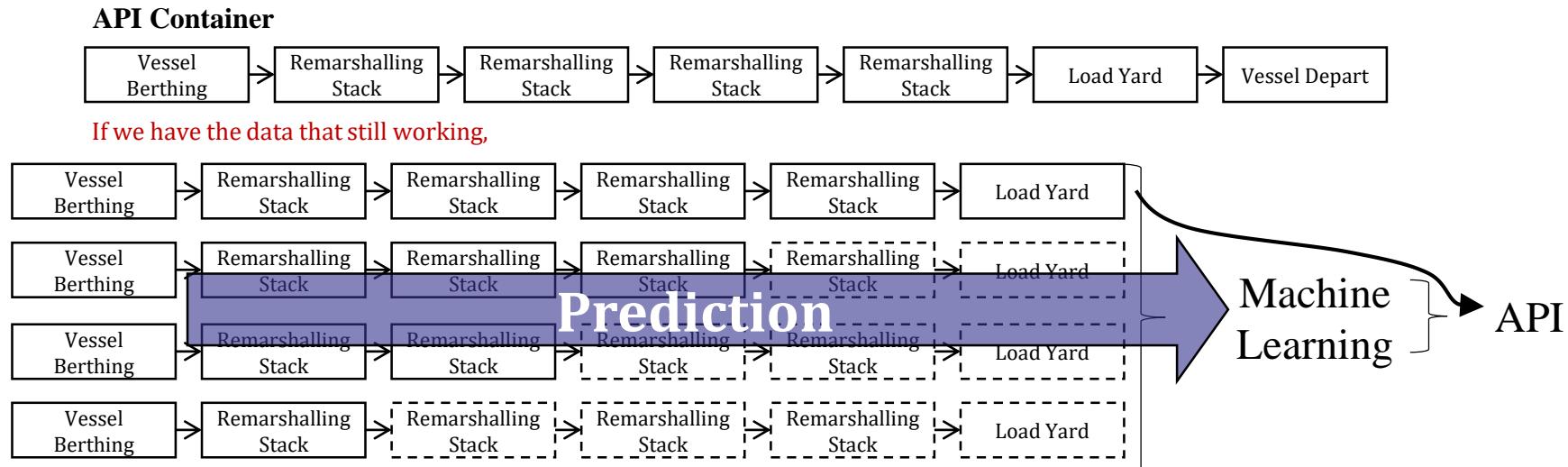
- Case Study



3.7 Abnormal Process Detection

Incomplete Anomaly Process Instance Predication : IAPIP

- Case Study



- 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

3.7 Abnormal Process Detection

Comparing Top 20 LAPID & the Estimated LAPID

- Case Study

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

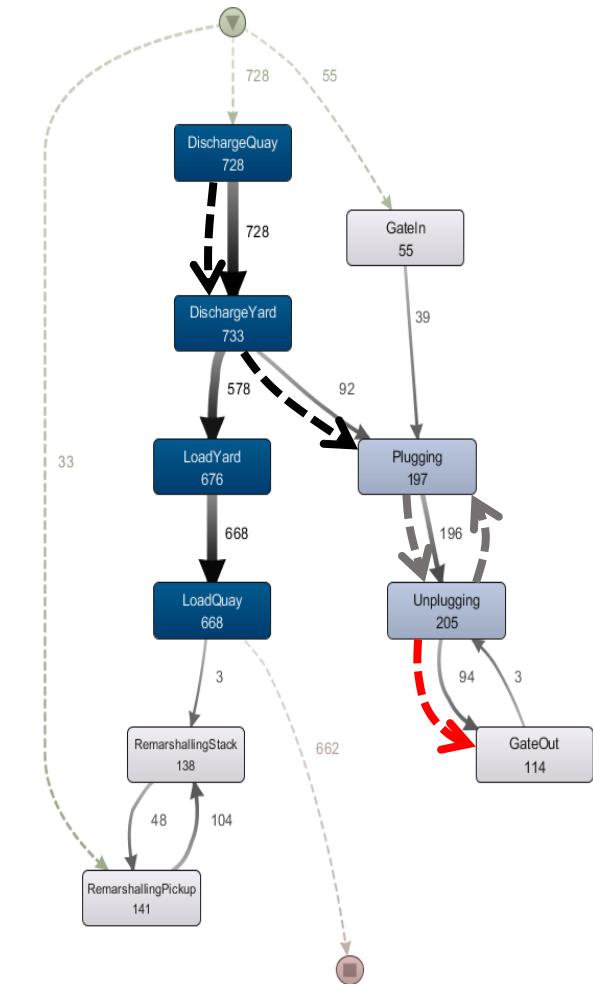
Container0905

Discharge Quay	Discharge Quay	Discharge Quay
Discharge Yard	Discharge Yard	Discharge Yard
Plugging	Plugging	Plugging
Unplugging	Unplugging	Unplugging
Plugging	Plugging	Plugging
Unplugging	Unplugging	Unplugging
Plugging	Plugging	Plugging
Unplugging	Unplugging	Unplugging
Gate Out		

60%

80%

100%



3.7 Abnormal Process Detection

Incomplete Anomaly Process Instance Predication : IAPIP

- Case Study

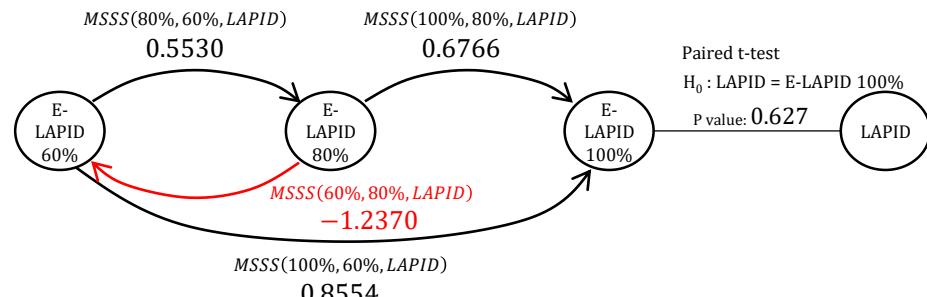
Top 10 of actual LAPID with Estimated LAPID
& Descriptive Statistics of test data

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

Correlation & Mean Squared Error between
actual LAPID & Estimated LAPID

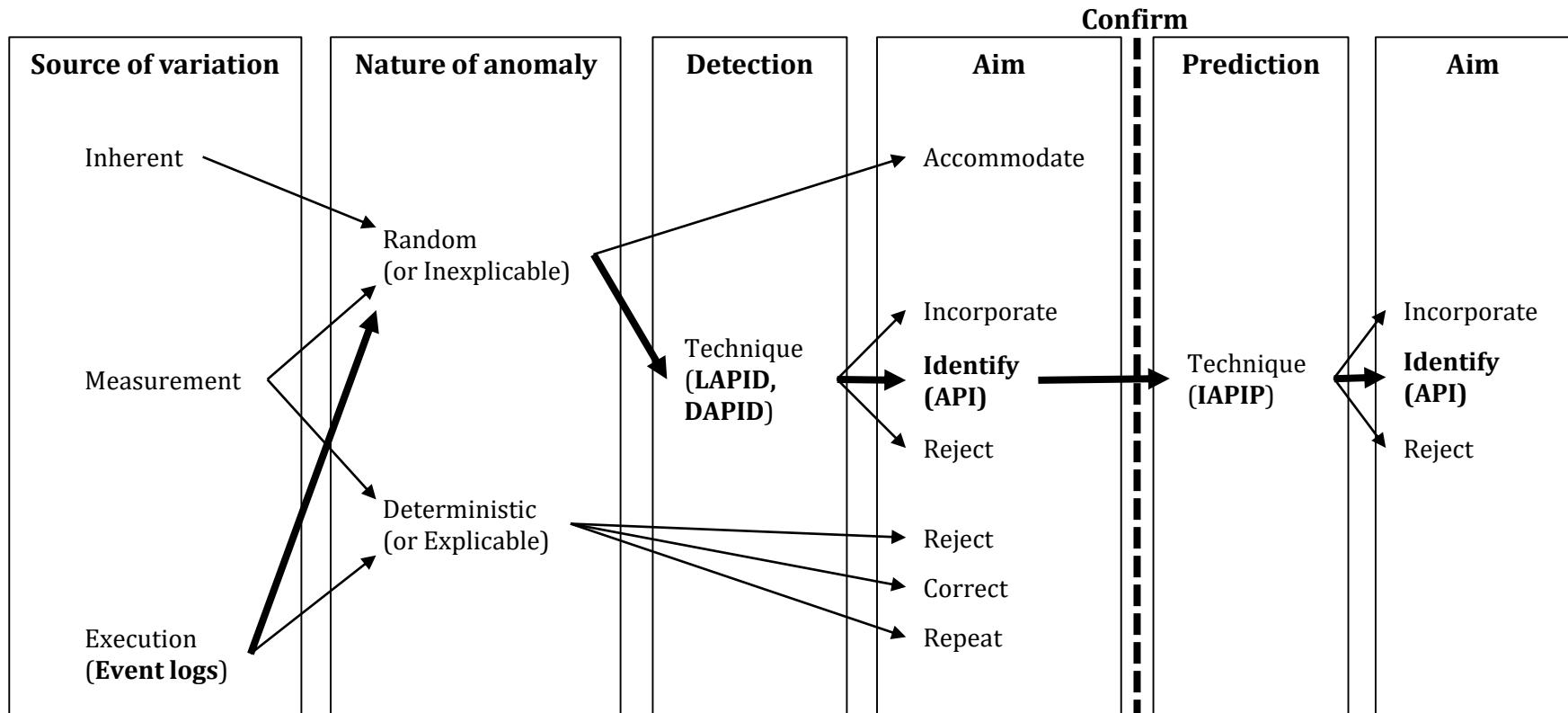
	Estimated LAPID (100% events)	Estimated LAPID (80% events)	Estimated LAPID (60% events)			
Actual LAPID	Correlation coefficient	0.949	>	0.868	>	0.633
	Mean Square Error	0.00270	<	0.00836	<	0.01869

- MSSS (Mean Square Skill Score)



3.7 Abnormal Process Detection

Overview of Detection and Predication of API



이론 강의 종료

고생하셨습니다